UNCLASSIFIED

AD NUMBER

AD488455

NEW LIMITATION CHANGE

TO

Approved for public release, distribution unlimited

FROM

Distribution authorized to U.S. Gov't. agencies and their contractors; Administrative/Operational Use; Jul 1966. Other requests shall be referred to Rome Air Development Center, Griffiss AFB, NY.

AUTHORITY

RADC ltr, 7 Dec 1971

THIS PAGE IS UNCLASSIFIED

RADC-TR-65-424
FINAL REPORT

DESIGN CONSIDERATIONS FOR AN ON-LINE MANAGEMENT SYSTEM

J. M. Allderige and P. J. Knoke

TECHNICAL REPORT NO. RADC-TR-65-424

July 1966

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of RADC (EMLI), GAFB, N.Y. 13440.

Rome Air Development Center
Research and Technology Division
Air Force Systems Command
Griffiss Air Force Base, New York

DESIGN CONSIDERATIONS FOR AN ON–LINE MANAGEMENT SYSTEM

J. M. Allderige and P. J. Knoke

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of RADC (EMLI), GAFB, N.Y. 13440.

AFLC, GAFB, N.Y., 31 Aug 66-113

FOREWORD

This report was prepared by Syracuse University Research Corp., Syracuse University, Syracuse, New York; under Contract Number AF30(602)-3505 and Project Number 4594. The RADC project engineer is William G. McClellan, EMIRD.

This is a final report; the originator's report number is SURC TR #20.

Release of subject report to the general public is prohibited by the Strategic Trade Control Program, Mutual Defense Assistance Control List (revised 6 Jan 1965) published by the Department of State.

This report has been reviewed and is approved.

Approved: WILLIAM G. MCLELLAN
Project Engineer

Approved: ROBERT W. QUINN, JR., Colonel, USAF
Chief, Intel and Info Processing Div.

FOR THE COMMANDER: IRVING J. GABELMAN
Chief, Advanced Studies Group

ABSTRACT

Under a contract to Rome Air Development Center (AF 30(602)-3505) awarded in September 1964, the Syracuse University Research Corporation has undertaken the design of a Managerial On-Line Data System (MOLDS).

This report, written from an operations research point of view, draws upon the numerous subtasks accomplished during the first year's effort under this contract, and attempts to put the entire MOLDS design problem in proper perspective.

Additionally, certain design specifications and implementation procedures are recommended. Discussions range from the detail level to the near-philosophic, but in many instances unnecessary detail (otherwise available to RADC in a series of interim reports) has been deliberately suppressed. Because the R and D management environment is the context for MOLDS design, specific attention is paid to that environment. However, it is pointed out that system design concepts and computer usage techniques developed for MOLDS should in general prove applicable to command and control systems and intelligence systems.

The major emphasis throughout is on the need for specific operating experiences to gain operational insights vital to effective implementation of MOLDS.

TABLE OF CONTENTS

TABLE OF CONTENTS - Continued

SECTION I. INTRODUCTION

1.1 BACKGROUND—LEADING UP TO MOLDS

In 1946, almost 20 years ago, the first modern electronic computer was built. The specific organizational idea had been on hand for more than a decade; the concept of automatic number manipulation for several centuries. The past 20 years or so have been bursting with technological innovations and also interesting conceptual innovations from Operations Research and the like on the management scene. Computers have grown apace. Hardware has progressed from tubes to transistors to current microelectronics, to trillion bit storage and nanosecond speed. Software has gone from cumbersome wired programs for simple arithmetic to simply phrased language programs for large complex algorithms.

With all the dazzling speed, mammoth storage and parallel sophisticated problem "solutions", there is an abiding sense of "underachievement" (a popular educational term these days for adolescents). The computer is still adolescent at 19 and there is no doubt some explanation in that for the unease. But, there has also been, over the years, premature and exaggerated heralding of what can be done and what such achievements mean. Fact never has quite matched pronouncement, which accounts for part of this dissatisfaction.

Yet, there is some truth to the unease, there is some underachievement; certainly computer capability now exists to do things which are not done. On the other hand, things must be and are done, particularly in the management field, which the computer cannot handle, at least now. Microsecond addition, 100,000 bit storage and linear programming (from operations research) are at an elementary (kindergarten) level relative to adult managerial activity. The embarrassment of would-be users, unable to use apparently adequate computerware, may well be unjustified - for in some matters, the hardware, and particularly the available software, are not adequate. This is really a tribute to the remarkable virtuosity and competence of management; a reasonably concentrated conceptual and technological assault has only scratched the surface. But, these scratches are impressive, and motivate a continuation of the assault which is building heavy breakthrough pressures.

Under Contract AF 30(602)-3505, the Syracuse University Research Corporation (SURC) has the task of investigating techniques for facilitating managerial use of computers. As a portion of this task, SURC is designing and will implement a Managerial On-Line Data System (MOLDS) at the Rome Air Development Center (RADC). Users of the system will be RADC Research and Development (R and D) Managers. Work on this contract to date has involved seven principal tasks. These are:

A. Determination of MOLDS specifications.

B. Development of a SURC prototype MOLDS (SMOLDS).

C. Study and experimental implementation of appropriate computer languages.

D. Familiarization with RADC hardware earmarked for MOLDS implementation.

E. Study of probable MOLDS impact on management.

F. MOLDS implementation.

G. Study of relevant areas of computer technology.

Specific results on these tasks are covered in a series of internal SURC reports, and will not be repeated here. Certainly, of all these tasks the first one is initially the most important, and it has received a major part of the first year's effort. Numerous SURC-RADC meetings have been held on the subject of specifications, because the great importance of the user's participation in the design of his system has been recognized.

The SURC techniques study does not begin from scratch, but rather has its basis in the successful on-line scientific system developed by Drs. Culler and Fried for classical mathematical analysis, also under RADC contract. Translation of techniques from a mathematical analysis environment to a management environment is not immediate, because the management environment is both different from and not as well structured as the mathematical analysis environment. Many of the techniques developed for a managerial environment should be directly transferable to command and control or intelligence environments, however.

This report represents an effort to put the entire MOLDS effort in perspective and share conjectures based on the experience of the project staff. Accordingly, much detail has purposely been suppressed; in particular, techniques are discussed in only a general way. The point of view of the report is that of the operations researcher rather than that of the (sometimes myopic) computer specialist. It means to suggest directions for action and study which will bring about an ongoing and ever-improving managerial-computer experience.

1.2 THE PROMISE AND PREMISE OF MOLDS

MOLDS has a straightforward promise: the freeing of an information user, such as management, from informational and conjectural bonds. It aims at enabling the user to get at all the information he wants with minimum delay and difficulty. It aims at releasing his imagination for ranging wide over complex future possibilities with minimum delay and difficulty. Note the word "aim" - this is a continuing process with new steps in "freedom" becoming possible as technology and knowledge progress.

This goal is not new; it has been one aim in Operations Research, Industrial Engineering, Management Science, Computer Technology, and the like, for many years. It is a new departure from the customary development work in these areas, however, in that it seeks to establish a computer-based "package" system applicable with varying extensiveness in a range of, primarily, R and D situations. There are concurrent efforts similarly designed in other situations, such as manufacturing and urban management. The task is large and will so continue, and we will benefit from the experiences and thoughts of others.

MOLDS has a premise fundamental to the continuing realization of the above promise: that the best from the state-of-the-art in several basic areas can be invoked for early designs of MOLDS and that design changes can be made to match progress in the state of these arts. This may seem obvious but it needs to be emphasized that a good deal of virtuosity is involved. We are dealing with computer and related electronic technology, with programming, with language, with the organization of systems and their representations, with information flow, with intelligence systems, with the process of managing. This is a comprehensive array of topics; it needs a comparable array of professional talents and knowledge.

Given this premise, the form of MOLDS depends on the developments in computers and display devices, in programming and language and in our comprehension of the target systems (such as R and D). These areas put constraints on some idealized forms; their states keep us from gathering and storing all the information we want to, from evaluating for intelligence as extensively as desired, from manipulating as large a number of factors as we might like to, and from communicating to the equipment and others the exact ideas and conjectures which we have. So as these states develop permitting more information and faster dealings with it, constraints are lifted. Also working in this direction is our increasing knowledge about what we need to know and manipulate - need as contrasted with "want" above. As we learn more about the operating characteristics of systems and the nature of the user-manager, we may well reduce requirements for storage quantity and computing speed in certain respects, or use them more effectively.

MOLDS is motivated almost in spite of the comprehensiveness of talent apparently required and the wide range of knowledge which must be tracked. MOLDS has a very practical motivation and in that there may be some basic theory. In getting something going, no matter how crude, the fundamental evolutionary process has started. Even though complete computer sophistication is not yet available, even though knowledge about languages or programming strategy is hardly consummate, there is a "live one" - a computer-based operating process of data inquiry and manipulation. This does not refute the premise, rather does it dramatize it, pointing out more clearly insufficiencies and their impact, a lack and how important it is to make that up. This motivation is always desirable; modifications should not wait for perfection of elements but should go ahead "cobbled up" if necessary. For in this aggregating of imperfect elements there is a large aspect of idealization. Elements, such as language or a particular computer mode, have most meaning in the total system - they may be less imperfect then we suspect. This is particularly true of the user-manager, another "element" in the above sense. The best way to learn of his needs and develop his capability to use the system to its utmost is to offer experience. This is the evolutionary process characteristic of a truly effective system with self-contained growth capability. MOLDS really suggests a modified way of life for the manager-user; it will take many experiences for him to establish the interrelations of his role and himself in this altered environment of instant total information and complex conjecture. It will take many such situations to create a workable "package" and the process of developing modifications to that package. What we can now implement is a skeleton, which has to be brought out of the cupboard before its dry bones can live, even a little. On this somewhat lofty note we can close these remarks and move on to specifics regarding this report and the work it represents.

1.3 REPORT COVERAGE

There are eight sections:

1. Introduction.
2. State-of-the-Art Projections.
3. Information Storage and Retrieval.
4. Information Processing.
5. Systems Projecting.
6. The User-Manager.
7. Summary and Conclusions.
8. Bibliography.

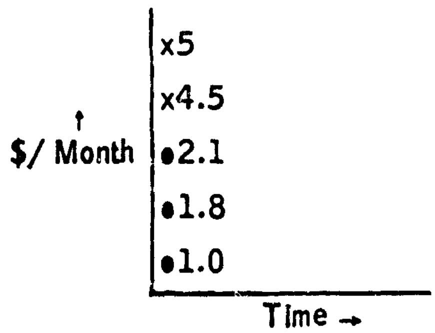
The introduction sets the stage emphasizing that the character of MOLDS depends on the state of several basic arts, such as computer technology. Section 2 roughly projects these states several years hence. Where we are now can ideally be expressed across a sophistication-cost spectrum and the looks-ahead can be similarly expressed. This idea is introduced and progress towards it noted. Given some idea of the states of these basic arts, the report, in Section 3 starts to look at the different subsystems of MOLDS. In discussing and projecting the Information Storage and Retrieval subsystem, we consider the computer as a massive storage unit with progressively lower cost per stored unit and progressively larger storage capacity. The retrieval process is thought of as answering the "what?" question. Information Processing (Section 4) represents an elementary form of doing something with the information retrieved - here, simple calculations (averages) and organizations of the data (histograms). Such manipulations start to help answer the "what if?" question, which is directly attacked in Section 5, Systems Projecting. Here the manipulations are more comprehensive and the user inserts conjectural data such as forecasted sales along with that of the stored data base. The aim is to answer such questions as, "What happens to the manpower load if sales (contracts) are up 23%?" - the "what if?" question. Extensive representations, called models, of the organizational system (or parts of it) are necessary to carry out "simulations" of future system behavior - simulations which are, in turn, required to answer this type of question. Section 5 notes the current capability for systems projecting and traces the impact on this vital managerial activity of future developments in the basic arts set forth in Section 2.

Up to this point, the discussion is concerned with the instrument - the state-of-the-art in technology, language, and so forth. The state of the user-manager is also important, obviously. Section 6 looks at the user and his target system. We are concerned with the management role in an R and D situation - what a manager needs to do, the character of his target system, how these considerations may change in time. Against this are the aspects of the user-manager himself - individual characteristics, learning capabilities, and the nature of his assimilation into a more technological environment. These two basic considerations - the system and the user-manager - permit some conjectures on implementation of MOLDS in its various possible forms at different times. Section 7 gives a summary and conclusions, with emphasis on projected work tasks. Section 8 notes some references, both internal (SURC Memoranda for the Record, etc.) and external (appropriate books and articles). These references are organized by section number and are not related any more closely to the report text.

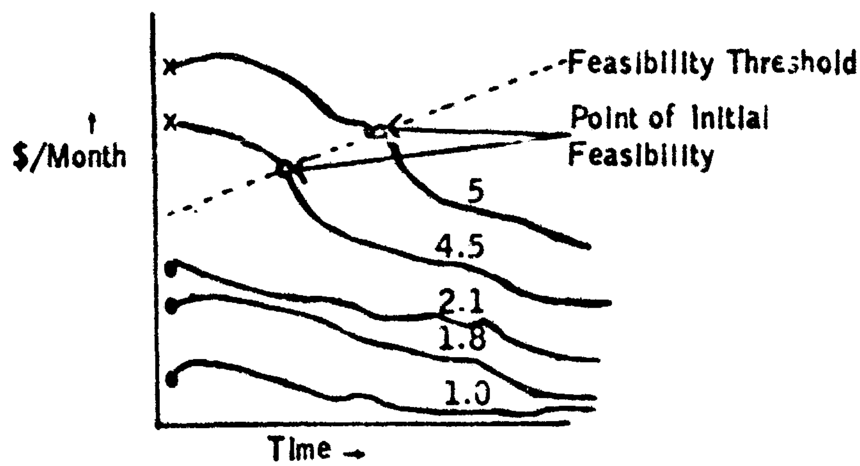
SECTION 2. STATE-OF-THE-ART PROJECTIONS

2.1 THE SOPHISTICATION-COST "FRONT"

In looking at different areas of computer technology, interest is in what you can get for what you have to pay. Neither side of this coin is explicitly defined but some approximations can be made. For instance, one gets storage size and speed, among other characteristics, from a computer system. These combine to give some sense of equipment accomplishment when different configurations are compared. It is possible to conceive of a spectrum of capabilities which can be related to cost. For instance, the use of a rating scale going from 1 to 5 for the capability or sophistication measure and considering cost as the conventional monthly rental amount, results in the following plot:



We then seek to track several points on the front through time. Note that two shown here are at a very high cost reflecting the fact that we do not now have the time or know-how to bring such configurations into being. We would expect these to come into range sometime in the future as different breakthroughs in technology occur. A possible projected picture is:



Different configurations may be affected in different ways by the developments yet to come, which accounts for the time lines not being parallel, while different system capabilities have a different feasibility threshold, which accounts for the slope of this line.

This portrayal is hardly complete. It is often inappropriate to combine features of a configuration for a single value measure of accomplishment.* Or, more correctly, weightings of characteristics may not be the same for different users - comparisons of configurations may differ from situation to situation. Moreover, cost is not as explicit as suggested; there may be fixed and variable costs involved. Nevertheless, the general sense of the projection-comparison remains in spite of these difficulties. The largest-fastest system is considered "better" and worth more than a small-slow system. The trick is to be more specific on dollar amounts and operational characteristics. The following paragraphs expand on the topics of computer hardware and software forecasts, and touch briefly on linguistics developments.

2.2 PROGNOSTICATIONS FOR RELEVANT TECHNOLOGY AND THEORY

2.2.1 General Remarks

Lack of accuracy is one normal characteristic of predictions of events depending on interactions of complex phenomena; another is the enormous effort required to obtain any quantitative prediction accuracy whatever. It is, in fact, a goal of MOLDS to alleviate the second difficulty. However, since MOLDS is not yet ready to assist in such work, and since there appears to be no lack of literature by computer futures prophets, no independent complete forecast of what's ahead in computer technology is attempted here (one small area has been closely investigated by SURC, however**). What follows is an attempt to capture the flavor of some probable future developments relevant to MOLDS.

*Combining characteristics to yield a single figure has proved useful in other fields, however. Electronic amplifying devices are sometimes identified with a "gain-bandwidth product", which is considered a figure of merit (in this instance, the higher the better). It is possible that an analogous figure could usefully characterize a computer system or subsystem, although we explicity define no such figure here.

**SURC TM No. 29, "Microelectronics and Computer Hardware".

2.2.2 Forecasts for Computer Hardware Technology

In a recent Harvard Business Review article,* John Diebold gives a thorough forecast of information technology. Loosely speaking, the best of future computers will be bigger, faster and more reliable than today's best, and the cost per typical problem solved will be less. Figures for certain specific areas, roughly based on the Diebold report, are:

	Today	Today Plus 10 Years	Gain
Add Time (nsec)	800	50	16 times faster
Reliability (MTBF, hours)	300	9000	30 times more reliable
File Size (millions of characters)	230	1,000,000+	4348+ times bigger
File Cost (cents per character)	100	0.4	250 times cheaper

Other probable significant hardware developments are too numerous to mention here, and the reader is referred to the recent computer literature for their coverage. Suffice it to say that future computer systems will offer more and greater capabilities at a cost within the reach of many. Even today, however, it may be said that the lack of suitable hardware is not the main limiting factor in computer applications. In many cases adequate equipment exists, but a lack of knowledge of techniques prevents its effective use. The new generation of display consoles, with their cathode ray tubes and light guns, is a case in point.

2.2.3 Forecasts for Computer Software Technology

The existence of a good software package is often a prerequisite to effective computer utilization. Computer languages, an important part of such a package, serve to increase a programmer's productivity (measured, say, in machine language instructions per day, or problems coded per day). Estimates of programmer productivity increases due to language development have been made, and a typical estimate

*"What's Ahead in Information Technology", by John Diebold, Harvard Business Review, September-October 1965.

appropriate to the field of scientific programming appears below.

Rate in Machine Language	Rate in Assembly Language	Rate in FORTRAN
X	10X	50X

Such estimates are useful, but they can be misleading. For example, there is evidence of the existence of what can be called a "convenience threshold" for a programming language, which implies that a certain class of would-be computer users will not use a particular language at all if the difficulty of learning and using it exceeds some critical value. Thus, many engineers and scientists may learn and use FORTRAN who would not have troubled to learn and use assembly language for a particular machine. Strictly speaking, the programming productivity gain for such a class of programmers is therefore infinite. Continuing in this vein, perhaps many managers who will not trouble to learn and use FORTRAN would use a suitable manager-oriented language if it existed. While one problem is certainly to implement such a language, a more immediate problem is to define such a language in the first place.

In the computer languages field, progress has been made in the mechanics of formalizing the definition of a language, and in systematically implementing a language which has been appropriately defined (say, in Backus Normal Form), but determination of the desired content of a language remains a cut-and-try procedure. The evolution of FORTRAN is an example of such a procedure, and shows that such a procedure works. Unfortunately, the procedure is expensive, because as each new FORTRAN version is defined a new compiler must be written for it (usually a time-consuming job). If we assume that the problem environment will continue to be too complex to enable the bypassing of the evolutionary mode of language development, then it follows that the need is for techniques to speed up both the evolutionary process of language definition and decrease the cost of the subsequent language implementation. Language techniques developed for MOLDS should apply to both these areas, because the language for MOLDS has the following goals:

A. The language developed for MOLDS will be universal (in terms of management); it will consist of a set of basic words in terms of which a manager is able to couch any of his problems.

B. The language for MOLDS will have growth capability, which means that new words can be easily coined, in terms of any previously defined or basic words, by the user-manager.

C. The language developed for MOLDS will be easy to learn and use.

A modified list-processing language called SLAP, being developed by SURC with these goals in mind, has shown good results. In SLAP, however, benefits to the programmer-user are obtained at the cost of computer run time and memory requirements to such an extent that for big, complex programs, SLAP may overtax the capabilities of even the biggest, fastest computers. This is an example of the inevitable trade-off between the finite capabilities of the computer user and the finite capabilities of the computer itself. To the extent that programs written in SLAP (or a similar language) are not unreasonably inefficient, SLAP can provide the vehicle for both the definition and implementation of the MOLDS language;* if the inefficiency becomes unbearable, then the principal contribution of SLAP will be in the language definition area. Language growth via this route is analogous to the development of a natural language wherein the users themselves develop the vocabulary and grammar while lexicographers and grammarians struggle to keep their respective documents up to date. In like manner, the observer** of MOLDS language development can serve to systematically collate and condense user-generated language output. It is interesting to note that insofar as a manager's language is a reflection of his environment, this technique of language development amounts to a method for "casing" the manager's environment.

Language implementation costs have decreased and will continue to decrease mainly because of increased implementation experience and improved implementation methods. The latter are related to improved understanding of what a language is; this will be discussed briefly below.

*See SURC MR No. 198, "Implementation of MOLDS in the SLAP Programming Language", by C. H. Burton.

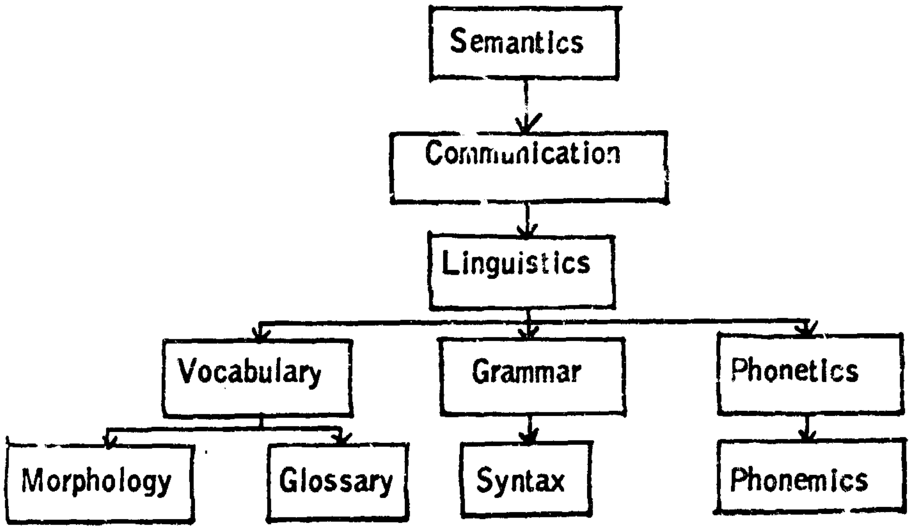
**Called the "System Director" by F. A. Dion, in RADC TDR-64-193.

2.2.4 Developments in Linguistics Theory

Serious study of linguistics has been underway for centuries, and progress has not been negligible. Nevertheless, in the area of "natural" languages there still exist disagreements on even such basic issues as what a language is and what a sentence is. The man who predicts resolution of all the major issues of general linguistic theory in, say, the next decade is therefore a true optimist. However, recent computer developments have stimulated interest in formal languages, and in this subarea of linguistics progress has been striking. Linkages can be (and have been) made between the theory of formal languages and the theory of automata, and, in turn, between the theory of automata and real computer systems. One result of these developments is improved understanding of computer languages and systems, and the outlook is for more of the same; improved design of computer languages and systems should be another result.

Nevertheless, he who predicts that computers can soon be instructed or interrogated in a reasonably complete version of a natural language (say, English) is still an optimist. Recent efforts in mechanical translation of natural languages highlight some of the problems, which are mainly in the area of semantics (see Figure 1 for a possibly helpful display of some of these linguistics terms). This is not to say that impressive results with pseudo-natural language input have not been obtained in experimental systems of limited scope (e.g., BASEBALL). In other languages like DEACON, semantic rules as well as syntactic are employed in the interpretation of sentences. However, there is a big difference between communicating with a computer in terms of a limited subset of English, as is the case in BASEBALL, and communicating in terms of a subset of English like that used in normal conversations with a business associate. Science fiction still leads the state-of-the-computer-art in this respect, and closing the gap will take time. Improved understanding of linguistics will help.

Progress in phonemics brings closer the day of speech input to computers. Machines which can recognize a small number of spoken words, to some extent independent of the speaker, now exist; but the mapping of spoken sentences into equivalent symbolic sentences is not yet feasible and here again some difficulties can be traced to semantics. Although some operating systems today use computer-composed speech <u>output</u> this does not imply that the problem of speech <u>input</u> is



DEFINITIONS

Semantics - The study of meanings.

Communication - A process by which meanings are exchanged between participants through a common system of symbols.

Linguistics - The science of language, including phonology, morphology, syntax and semantics.

Vocabulary - All the words of a language.

Glossary - A collection of terms limited to a special area or usage.

Morphology - A study and description of word formation in a language including inflection, derivation and compounding.

Grammar - The study of the classes of words, their inflections, and their functions and relations in a sentence.

Syntax - The way words are put together to form phrases, clauses, or sentences.

Phonetics - The study of systematic classification of sounds made in spoken utterance.

Phonemics - A branch of linguistic analysis that consists of the study of phonemes.

Phoneme - A member of the set of the smallest units of speech that serve to distinguish one utterance from another in a language or dialect.

FIGURE 1 . PARTIAL LINGUISTICS TREE

close to solution. Here again, it seems safe to predict no major breakthrough in the near future.

Another area where help from linguistics seems to be needed is that of computer graphics. Computer hardware which permits graphical input-output is now with us, but its effective utilization has yet to be generally achieved. Lack of suitable precise vocabulary is probably one of the difficulties. An interesting particular case of a graphical input unit is the RAND tablet, which could make handwritten computer input practical. Getting the computer to recognize sloppy handwritten characters, entered via this device, has proved difficult, though some success has been achieved. We will not go into pattern recognition techniques here; suffice it to say that if it were possible to give a precise verbal description of graphical forms like a "sloppy handwritten A", part of this particular pattern recognition difficulty would disappear.

In summary, many problems in computer usage can be fairly described as language problems, and hope for solution of many language problems seems warranted in view of contemporary vigorous study of (particularly mathematical) linguistics. Generally, the complexity of such problems makes continued steady progress seem more likely than any immediate theoretical breakthroughs. In the near future of 10 years or so, we can see computers which are somewhat faster and more reliable, quite a good bit less costly for the same job - or more competent for the same dollar - with tremendously larger storage. Display capability is opening another communication path permitting pictorial and diagrammatic exchanges between the user and the computer. Effective voice exchanges, however, are not in sight. Programming-language features are all in the direction of simplified user contact with the computer. Programming, as now known, will continue to move away from the user to the manufacturer and evidence itself in hardware-software packages which will get progressively bigger as more computer-applications worlds are conquered.

SECTION 3.

THE INFORMATION STORAGE AND RETRIEVAL SUBSYSTEM

3.1 THE "WHAT?" QUESTION

The information storage and retrieval subsystem enables the user to ask, and hopefully get an answer to, the "What?" question. A typical question from the R and D managerial context might be, "What are Jones' contracts?". Here, the system would produce at least a title listing of the contracts on which Jones was working. Now, this suggests a file of contracts with at least the names of people working on them included along with the title of the contract. The same question might really indicate that a user wanted to look at the document itself. Here, the system would "dump" the contents of that selected section of the file reproducing all of the particulars on the contracts. This example points up the ambiguity of conventional language and the need, as noted earlier, for explicit statements and methods for making explicit statements.

Another typical inquiry might be, "What time was spent last week by Jones on Contract 35-32?". This is a straightforward question for which the system should produce a single answer. But, it introduces another idea, that of file updating. For, in this case, there is an inquiry regarding a recent bit of information added to some original set of information in the file. Specifically, there is required a weekly posting of time amounts on the part of each engineer against the contract. So, elementary storage and retrieval is exactly what it says it is -- one collects a lot of information, puts it some place, and wants some or all of it back later.

This part of the MOLDS system, called RESS, is at one extreme end of the retrieval spectrum. It gives to the user the information as it is stored. This information can be screened, but RESS does not include manipulations of any other sort. There may well be alterations of source data enroute to storage, but this is a matter of storage design, not of retrieval, although it may anticipate certain retrieval needs. Any data manipulations are considered for convenience to be part of information processing or systems projecting discussed later.

3.2 THE DATA BASE

3.2.1 Document Input

Conventional information storage is still on documents and this will probably be the case for several years to come. So there is the problem of transferring this information from the documents to discs or tapes for computer storage. There are two broad aspects to this problem; conversion and storage. Conversion centers on two tasks - file establishment and file updating. The establishment of a data base often looms as a monumental task. One is faced with the need to transfer a large number of records into a computer-readable form. There is a choice - one can do all records for the total establishment at one time taking as current cost whatever man-months are necessary. The alternative is to evolve the data base; here you either start from zero and accumulate it over several years, or establish so-called key parts of the data base and let the balance accumulate over some longer time period. Since, in many instances, the existing document data base for any reasonably sized organization is an extremely large one requiring as much as six to nine months just to shift over, there is a strong argument for the alternative accumulating approach. Many retrieval systems bog down at the outset because of the overwhelming nature of file or data base establishment; perhaps it is better to plan a three or five year goal for complete establishment. Working out comparable progressive operational strategy is not that difficult.

Now this may undergo some meaningful changes in the next several years. The usual current method of transfer is by keypunching but we are moving into a time of optical print readers. Surely the speedy ultimate in transferring existing data base information to a computer would be to have the documents automatically read rather than laboriously transcribed. There are two difficult problems with optical readers besides the obvious technical problems - poorly executed source documents and the need for "intelligent" information reduction. Even with documents that are well structured there are many instances of source recording being done in a casual way. This results in a document difficult enough for a human to read, let alone automatic equipment. The information reduction problem is most critical when documents have prose portions. At present, these portions either must be paraphrased, or else key words must be established. Both these problems are enroute to solution, the latter perhaps more than the former*. But, they both suggest that, since optical reading will soon be an effective fact, current document strategy should be mapped out in the light of this eventuality. Forms redesign, monitored execution of source data recording, and even standardized language are all appropriate measures to take right now. Such measures insure that two or three or four years hence when the data base is subject to transfer, the documents will be in acceptable form for an effective conversion job. Needless to say, such measures are thoroughly helpful to any information system whether or not it ever gets on a computer. This process, moreover, is compatible with the accumulation alternative noted above. No matter what develops, quite a bit can be gained by insisting that new documents, at least, be in good format and appropriately executed.

3.2.2 Non-Document Source Recording

While most organizations, particularly R and D, have existing data in document form, recording technology is getting to a point where this might not be the case over many more years. Some non-managerial motivations have already resulted in card, magnetic tape, or disc storage of engineering and scientific information. In such instances, the data is already in computer usable form. Such data, however, quite likely went through an initial document phase and then a transcription phase from document to this other form. But, this is still a forerunner of the more effective developments in point-of-event recording.

In manufacturing operations, records of production, time records, and so forth can quite readily be noted at the operating position through remote recorders and sensors. The technology exists for such recording in an R and D situation also. With remote keyboard consoles having direct access to the computer, punched card badges, remote keyboards, and typewriters with either punched paper tape or magnetic tape output, practically all the source information can go directly into computer storage on the first noting of the event. Since such capability currently does exist, the sooner organizations can reform their current data keeping along these lines (where applicable), the more easily their major file establishment can be conducted in future.

*Human problems often tend to remain unsolved longer than technical problems.

Note that this source recording capability seems to be moving apace with optical scanning. Both of these strongly suggest complete overhauls of data and information origination procedures. We will get to the point where it is much less costly to conform to certain standards of writing multiple-choice procedures of information selection, of keyboard disciplines, or of punched card selection, than to go the current path of somewhat casual information origination and keypunching transcription. There is quite a bit of potential versatility developing in information system design - on one hand information can go directly to the computer which, in turn, can produce certain asked-for documents or displays for those parties needing a piece of paper or detail. Or, on the other hand, the paper can be produced for whatever circulation is required and then automatically put into the computer. This, of course, raises the whole issue about local or personal information storage (why the document for circulation ?), desk files, notebooks, and the like, in light of immediate retrieval capability complete with remote display. The data base question is really being asked for the first time.

3.2.3 Storage Strategy

One key to having good retrieval of information is either knowing where it is or knowing what it is called (and the two are not necessarily basically different). Computer storage presents alternatives which may suggest redesigning some of the current storage concepts. Particularly as we move into an era of tremendously increased storage capability, there are several new possibilities.

One can store and identify information exactly as it is on the documents. This is a straight mimic of what is currently done. Since many document files tend to be at some point in a haphazard evolution, another approach is to reorganize the documents and/or the information on them in some way. The user is accustomed to the old way, however, and can only, perhaps, recall the information through the old name. As a consequence, a dictionary or glossary is required so the user can convert from his old concept to the new. Then, there is the move to collapse the information eliminating as many of the redundancies as possible. A document, in fact, often happens to be a convenient (at the time) response to a certain collection of information, disregarding other existing information. As noted above, in a period of instant retrieval and display, redundancies may be eliminated in information origination.

Any such collapsing of existing document files, however, requires that the system produce a glossary or dictionary. Of course, the ultimate and soon to be is both non-redundant information storage and non-redundant information collection. Such information would then be extracted using a thesaurus concept - the user could lapse into informal and casual language while still retaining some hope of getting at least the information he wanted (most likely more, since, when in doubt, the computer would serve up anything close as a possibility).

Although the user thinks largely in conventional terms with regard to information and his own unique terms, there is enough experience to realize that re-education or re-learning is not impossible. Any new storage system carries with it a tremendous training responsibility which can be effective. But, the impression regarding storage and storage strategy is that things are in a state of considerable change and any system installed should gird for such change. While, for a long time we probably will not get to the point where the computer will respond sensibly (if this is even possible) to the inquiry, "Give me what What's-his-name is doing.", we will shortly be able to accept a query such as: "What are the jobs that Joe Bones or Johnson is working on?". When Jones is really the man in question, the computer will respond with several possible names resulting in a dialogue not untypical of a manager with a secretary, at least on this kind of matter.

3.2.4 Some Notes on Storage Size

Some early examinations of the organization size and data base requirement have been made at SURC and RADC. This needs a lot more work to have sharp operational meaning, but there is an approximate estimate that from 3,000 to 5,000 characters of storage are required per person in an R and D organization. A 100 man organization, according to this rule, would require from 300,000 to 500,000 characters of storage. While this may seem relatively small, or rather that our current capability more than handles it, it must be noted that other considerations prevail. For instance, this refers just to the data base and not to the fairly heavy storage requirements of program instructions. It is stated for an on-line real-time mode but there may be other portions of the Research and Development activity seeking an on-line real-time mode. While the world of the Research and Development efforts may be somewhat different from the world of the manager, there can be overlaps. Certainly, in contract negotiation and proposal preparation quite a bit of substantive information is required which would tap this other world, the non-managerial data base. More operational work needs to be done to get a good description of the real equipment requirements for a total system operation.

This does introduce a size-speed-coverage problem associated with storage. If all of the data base cannot be represented, then there are two alternatives. One is to give it key representation and still permit an on-line real-time operation, perhaps not up to the 100% quality, however. The other alternative is to split the storage hoping there will not be need for interrogation of two or three or four tapes (discs) within one inquiry and trying to design to avoid this. But, even if this is avoided, there still is a time delay to shift tapes or discs after it has been established that the one on line does not have the desired segment of the data base. What the penalty is for partial off-line operation is hard to say. It does need study, however - not only from the point of view of limited storage but also from the point of view of quality. The noise problem has been raised before and it should be noted in passing here that the discipline of off-line operation in some things may add to the quality of the inquiry and the understanding of the problem.

3.3 INFORMATION RETRIEVAL

3.3.1 The General Mechanics

Here we are talking about identifying the information which we want and getting it. Several different mechanics can be identified, the most direct of which might be called "inquiry". This is an immediate, unambiguous statement of what the user wants. Ambiguity, or lack of it, not only exists in the mouth of the speaker but in the ears of the listener. A non-ambiguous exchange requires some joint agreement on what words mean what. The computer is not yet part of human society, and though it has done remarkable things, it can still only understand a pidgin English. So there is a constraint on the user to use this pidgin English and he must be trained in its explicit meaning. This still represents quite a step forward, for such inquiries are a kind of programming and such stilted English languages represent user-oriented languages in the true sense of the word.

Here is an example from SMOLDS (SURC's prototype for MOLDS) of the language required to ask the question, "How many contracts are for more than $50,000.00?" This assumes a file of RADC Form 77s is part of the SMOLDS data base. The system is on-line and the user is at the console typewriter. The following takes place:

Computer: Please type your program.

User: Known 1 77/AMOUNT/G/50000.00/

Now the above is not complete - there is an instruction regarding totaling of such contracts but we will touch on that later. This does convey enough of the kind of language that is required. Specifically, the user will start out with the word "known" any place on the line. He must then space any number of spaces and put an identifier ("1" here), and then space again any amount before typing the terminal part of the message. The terminal part opens with the form identification number ("77"), a slash line, the information block identifier ("Amount"), slash line, the relationship ("G" standing for "greater than"), slash line, the value ("50000.00" representing fifty thousand dollars), slash line. This is not an atypical language and there is no intent here to demean it. But it does require a certain conformity and a certain wording on the part of the user. One might question, for instance, the mixture of free spacing and fixed format. It might actually be easier for the user to be taught that everything is fixed and that the message would start out as "known/1/77/", etc. There are involved here considerations of user-learning capability, and of language design to meet that capability and minimize the user's learning time.

The need for studying languages in the above light may be avoided by the use of a man-computer dialogue which extracts conformity from the man. It is possible in one of the languages of MIT's Project MAC, for instance, to preface an inquiry with a vocabulary list permitting the user to select the appropriate terms from the "menu". While it may take a few exchanges of conversation for the user finally to frame his inquiry, it does not presuppose any learning on his part in advance of the communication. This kind of capability is one step away from completely free speech - that is, from almost free speech, for there are always some constraints. Nonetheless, there is a continuing move towards computer acceptance of fairly free-form inputs, constantly approaching normal written communication.

But, it must be noted that special forms for input and inquiry do serve the purpose of better problem formulation. While it may be technically possible to engage in some relatively free inquiry, it may be still operationally advisable to keep such inquiry in a somewhat stilted and formal form.

3.3.2 Some Operational Considerations

There has been some reference up to this point of the do-it-yourself kind of system. One important capability of MOLDS - one important specification - is that of enabling the user to build his own convenience vocabulary. It is easy to see that after considerable use certain standard inquiries may evolve - that is, certain inquiries may be repeated with sufficient frequency to permit them to be established as "standard". "Standard" suggests that they are always to be used under a certain set of circumstances, so it might be more appropriate to call them labeled or identified inquiries. For instance, we might call the above inquiry "No. 1". Then, in the future, any time we elected to inquire about contracts over $50,000 we would merely introduce the words "No. 1". Or, we might construct slightly simpler language and call the above "C 1". Here, the letter "C" might stand for contract and the "No. 1" the first inquiry regarding contracts. Whatever the form, there is the idea of accumulating evidence regarding frequency of use and taking advantage of that frequent use in some automatic way. The evolutionary aspect of this system has been mentioned before and here it is in evidence in the specific context of the retrieval subsystem.

It is not inconceivable, moreover, to anticipate the kind of inquiries which would be made frequently and present the user with an opening set of such standard inquiries. These may disappear in part or in total as time moves on but they may well serve a purpose in that process.

In somewhat the other direction, it may be quite possible that a type of inquiry cannot be made within the framework of the existing form of the language. For instance, the concept or instruction of "greater than" coded by the letter "G" may not be expressable in the user's language. The need for such a relationship expression may well arise and the user is logically interested in adding this to his system. One way is to instruct a programmer to go into the program and make such an addition. If the host language for the system is, say, FORTRAN, then there may be some question as to whether the user-manager will be in a position to effect the addition. If, however, the program is in the form of some minimal set language, such as SLAP noted in Section 2, then conceptual additions may be within the capability of the user if he elects to handle it that way. A minimal set language seeks to establish as low a number of language elements as possible, offers a parallel set of operating or manipulating instructions, and then commences an evolutionary process. It is a kind of boot-strap operation but, as such, it permits the user to get by with a very limited opening vocabulary together with a few exact, yet unfamiliar, rules of grammar and syntax. It may put a considerable strain in speed of execution on the computer, but this is still an appropriate direction in which to move and it does permit ease of change. In the various forms such languages have taken, for example SLAP, it does present a somewhat bizarre exercise and it remains to be seen in operation how a user will respond to it. In developing a convenience set of operators for inquiry, such a language is well suited. It is right in keeping with the evolutionary concepts noted and, when coupled with good tracking and monitoring techniques, should present an interesting approach operationally.

There was some note above on the size-speed-coverage problem relative to storage. This is really a storage-retrieval problem and it should be noted again that the on-line, off-line determination is still to be made and is an important area for study.

3.4 SUMMARY, NOTES AND TASKS

Information storage and retrieval requires extensive discourse with the computer. There is the initial job of getting the man-computer cultures speaking to each other - akin to making contact between alien races. The establishment of the data base in computer form is the first task - it can be done all at once or accumulated over some period of time. An existing data base in punched card, magnetic tape, or disc form presents no problem and is ready for conversion in its entirety. Conversion of <u>documents</u> can be a real problem but developments in optical scanning may shorten the all-at-once approach considerably. We can look forward to direct information input with remote keyboards as well as pictorial input - graphs and the like - from some of the capability now being developed. There is considerable versatility emerging in information input and display. It suggests some basic changes in concepts very significant for information storage and retrieval. Storage strategy considerations are still of interest to make maximum use of this progressively enlarged capability.

Retrieval languages continue to move apace, all in the direction of more user-oriented languages. This tends to relax some discipline in problem or inquiry formulation but widens the base of potential users considerably. Operational characteristics of management on-line retrieval systems are not well known but it is envisioned that any one situation will evolve its own unique set of inquiries. To this end - to this do-it-yourself end - certain minimal set languages are proposed which seem to be more compatible with this evolutionary process.

Information storage and retrieval is on route to a far more effective day as technological improvements in size and speed and display continue. The great need is in understanding use and gaining operational experience with this new dimension in managerial living. Certain tasks for MOLDS suggest themselves as some of these developments move in on the system. Document discipline is a continuing study need. This includes the nature of form designing, the problems in accurate execution of data origination, procedures, and the impact of forms thinking on the data originators. Regarding documents, but in another study direction, is the task of staying abreast of optical scanning developments. Special attention should be given to remaining current and working on, where possible, more effective user-oriented languages. This includes the range from minimal set to task-oriented languages.

From an operational point of view, the on-line-off-line problem needs considerably more work. Moreover, of supreme importance, is the task of monitoring the use of the retrieval portion of the system, of looking at it as an evolutionary operation and gaining insights so as to accelerate this evolution. Information storage and retrieval, the sensing of information, putting it someplace, and getting it back again - this process is a life process and will go on and on. We are developing the capability to jump from a Neanderthal to a Cro-Magnon state. If we can use the same competence in evolutionary development of the use of technology, which we have demonstrated in the development of the technology itself, then the cultural shift will, in fact, have taken place. Resigning ourselves to haphazard interpretation of the experiences and response to impulse will only result in a vastly underused potential which may, in itself, prove to be debilitating. The evolutionary operational concept is one which will be emphasized again and again - it is the cornerstone of MOLDS philosophy.

SECTION 4. INFORMATION PROCESSING

4.1 SOME DEFINITIONS

Raw data as it is stored, either from documents or from remote serving stations, is usually not in appropriate condition for study. There is a natural inclination on the part of the user to manipulate it into better form to understand the problem of the moment. These manipulations range over a rather wide continuum - from the extremely simple to the tremendously complex. We have elected to call simple manipulations "information processing" and to call the more complex manipulations "systems projecting". This distinction reflects what is felt to be a user comprehension of these manipulation methods.

The simple manipulations considered in information processing fall into two classes: Comparative and Arithmetic. Comparative manipulations deal with the array and classification of raw data. Special group listings, histograms, selective classification reporting, ranking, and so forth, are all comparative manipulations. On the other hand, a total, an average, a percent, a range, a quartile, and so forth, represent arithmetic manipulations. These manipulations, in both classes, are quite well understood by most users who have probably carried them out manually at one time or another. There are possibly other kinds of manipulations not noted here; a list of specific operators is discussed below.* There is, of course, no rigorous operational definition of information processing, although the required number of machine language commands for description of a manipulation is a possible measuring scale (assuming a command set typical of general purpose computers).

In terms of the current IBM 1620 configuration at SURC, for instance, a profile of different processors or operators has been drawn based on the number of FORTRAN statements, assembly language statements, and core storage digits required. This is shown in Table 1 . While only approximate and dependent on the version of the operator, the table does show the order-of- magnitude difference in operators - "count " and "order" are considered as information processing while the others are part of systems projecting.

By way of an example to complete the information processing definition, consider a logical follow-up regarding Jones' contracts. In 3.1 on page 3-1, recall that there were two information retrieval questions, "What are Jones' contracts?" and

* See Section 4.4

TABLE 1.

PROCESSOR PROFILE

Processor	No. FORTRAN Statements***	No. Assembly Language Statements***	No. Digits Core Storage*
Queuing Model	500	2000	25K
Funds Position Simulator	500	2000	50K
Stuff (Data Analysis, Regression, etc.)	NA	7000**	220K
SMOLDS Count	15	100	1K
SMOLDS Order	75	600	16K

*No. digits core includes table storage.

**Estimating that slightly less than one-third of the total digits of core are devoted to instructions, at 10 core/instruction.

***Includes processing portions only - input/output instructions not included.

"What time was spent last week by Jones on contract 35-32?". Here are two information processing questions, "What is the average weekly time, per contract, over the last thirteen weeks? What is Jones' average weekly contract load?". The system would produce two answers, for example:

Average time per contract per week (hours): 7.1

Average weekly contract load (hours): 28

These both represent simple manipulations on data base information. More manipulation may be wanted, but if it is complex we consider it under systems projecting discussed in Section 5.

4.2 THE "CONVENIENCE SET"

The introduction of manipulative capabilities also introduces a fairly important concept, the "convenience set" concept. In the discussion up to this point, considerable emphasis has been placed on the user's ability to fashion his inquiries using basic language elements or a "minimal set" as noted in Section 3. It was also noted that some inquiries may be repeated and procedures can therefore be stored to be called out in a simpler fashion than originally. While not so named, such stored and self-developed inquiry procedures are a convenience and the collection of them comprise a convenience set of inquiry procedures.

There are some procedures which are so logically usable and generally understood it seems pointless to let the user literally rediscover and reform them. Indeed, they should be either a part of his building blocks or basic language. The manipulations noted above as part of information processing are of this nature. Since they would quite likely emerge as a self-developed convenience set, they are introduced into the MOLDS system at the outset, in anticipation of their extensive usefulness. There is, then, a standard convenience set of procedures as well as the self-developed convenience set. Such a set is convenient in several senses. The first has already been noted; it is more convenient to use one word, say "average", then to set forth many, many times the specific elementary instructions for obtaining an average. These are conveniences in the initial version of MOLDS for any particular situation - the user has a starting point regarding his processing operators. And, finally, the user has a convenient set of building blocks which, when used with some minimal set, may be used for developing a larger collection of operators, a collection which may turn out to be one of a more complex convenience set.

4.3 COMPARING "CONVENIENCE" AND "MINIMAL" SETS

Here is an example of current SMOLDS procedure, part of which has been displayed before. To recall, the user wants to find out how many contracts, recorded on Form 77 (per RADC), are for amounts greater than $50,000. The computer and the user, at the on-line console, have the following exchange (line numbering is done automatically by the computer):

Computer:	Please type your program.	
User:	Line 1	KNOWN 1 77/AMOUNT/G/50000.00/
	Line 2	END
Computer:	Please type your processing specifications.	
User:	COUNT/1/	
Computer:	7 Forms were retrieved for this label.	

A minimal set language requires a little more. Here, for instance, is a part of the same problem in SLAP assuming one was starting from scratch (retrieval routines for SLAP have not yet been worked out):

COUNT:LAMBDA,X,(COND,(NULL,X),T,Φ,1+(COUNT,(CDR,X)))

The above serves to define the operator "COUNT".

Thereafter, the operator may be used as

COUNT,LIST

Where LIST is the name of the list to be counted.

In actual practice, such a language as SLAP would also anticipate usage and include "count" or "total" as part of convenience set. The point here is to note the details, slightly, of fashioning a simple command, details which may prove easier to master than those of a language such as FORTRAN. The advantage of convenience set is seen, as well; the combination of minimal and convenience sets is basic to the success of any system.

4.4 THE CURRENT INFORMATION PROCESSING OPERATORS

The following is a list of convenience set operators, grouped under information processing, which can be accommodated with SMOLDS or are being planned. (In all cases the operand (or operands) are lists.)

ABSOLUTE VALUE	MINIMUM
ADD	MULTIPLY
AVERAGE	ORDER
COUNT	PROFILE
DIFFERENCE	REVERSE ORDER
DIVIDE	SUBTRACT
FIRST ELEMENT	TOTAL
MAXIMUM	VARIANCE
MEDIAN	

The activities indicated on this list, while not exhaustive, are certainly reasonably inclusive. They all conform to the criterion mentioned above - they are familiar to most possible users and probably have been carried out manually at some time or other.

There is provision to add to or delete from this set as well as combine for the larger sets noted above. This group of operators plus a minimal set of basic relationships comprise, at any point in development, the system's total language for current inquiry, manipulation, and new vocabulary formation.

4.5 OPERATIONAL DEVELOPMENT

Operational development moves toward a more effective set of information processing operators. If we take the point of view that a convenience set of simple manipulations will probably be unique for a given situation, a particular R and D manager eventually might find himself using only a portion of the original operators and possibly wishing he had two or three others. The evolutionary aspect of MOLDS has been noted before and here it is noted again; the user can probably meet his needs, at least to some extent, with respect to these other operators since there is the built-in capability of developing his own procedures. He can also eliminate procedures from the initial package if they prove not to be useful.

In giving the user an initial convenience set there is the responsibility of the system to give him the capability of evolving to his own particular set. One of the operational developments will be to give him better guides for observation of his performance with these sets of procedures. Without such guides it is possible for him to carry out essentially the same function with different collections of instructions and continue to miss a pattern of instructions which could be stored as a convenient procedure. Some clinical observation capability must be put into the system so that such patterns are detected and maximum efficiency achieved over some period of time. Observation and review on the part of another party might well develop more effective procedures than those naturally evolved by the user. This process of monitored evolutionary development is the same as discussed in Section 2 relative to language. Operational development, then, will give us more rapid and incisive ways of converging on some eventual convenience set unique to the particular situation.

4.6 TECHNICAL DEVELOPMENTS

There is no technological block to reasonably rapid execution of the simple manipulations ininformation processing. Technology is apace use, for the most part. There are several situations, however, where on-line, real-time results would currently have considerable impact - such as during contract negotiation or in the rapid fashioning of a proposal. Moreover, as use and understanding improve, there will probably be more effective routine (as opposed to special) work done because of on-line arrangements.

New technical developments of Section 2 will produce greater ease and speed but the effect of these changes cannot be very large on information processing. As far as the user is concerned, he can quite easily and rapidly get an average now, for instance. Quality or reliability improvement is not easy to evaluate. The growing capability of display and picture input may give information processing an important enrichment. It certainly is appealing to envision an instantaneous histogram - the psychological boost may be important.

The size projections - some 4,000 times larger in ten years or so - noted in Section 2 are of interest. The user's scope will vastly increase and real-time information processing will be possible over any currently conceivable size of R and D establishment. It would relieve the speed-size-coverage problem existing currently in very large organizations as noted in Section 3, 3.2.4. Current storage capability either forces a restricted data base with real-time capability or a split-facility data base with only partial real-time capability. The extent of time deferment would depend on the computer manning. The penalty for non-real-time is hard to assess - more on that is noted in Section 5, 5.2.2.

4.7 SUMMARY NOTES

It is not within our scope to project explicitly along the idealized lines introduced in Section 2. Certain trends stand out and certain tasks are suggested. Information processing, conceptually, is reasonably well established. Technical developments in speed, language, and size will need to be responded to but this is seen as routine with large benefits only in size. Display and pictorial communications developments may be of some significance, vis-a-vis information processing, and deserve more than routine attention.

Operational tasks are vital; it is in this sphere that breakthroughs and progress are to be made. Tracking experience in different environments will be a continuing responsibility. Only by formalizing such experiences can we develop effective convenience sets and progressively more powerful methods of deriving ever-new sets. These really comprise an evolving language aimed at better user-computer communications for simple data and information manipulation.

SECTION 5. SYSTEMS PROJECTING

5.1 CONSIDERING CONJECTURAL DATA

Manipulating data base segments is not sufficient for a manager. Indeed, it is only a beginning. For the "What...if?" question can only be answered slightly when one is limited to the historical data base. The user-manager understandably has possible data he wants to insert to complete his thoughts on a matter. In the Jones question of Sections 3 and 4, the user initially wanted a listing of all Jones' contracts (retrieval), page 3-1. Then he wanted a listing of time on a contract last week (retrieval, modified by prior classification). A logical follow-up was the average time per week per contract using the last thirteen weeks (information or data base processing), page 4-1. An equally logical inquiry might well be, "What will his contract load be if he handles two more contracts at five hours a week?". Here is the "What...if?" question with conjectural information, "two contracts" and "...at five hours a week".

Since Jones' contract load was earlier noted on page 4-3 as 28 hours, simple addition gives the new load as 38 hours. This assumes, however, that his contract capacity is, say, 40 hours and that any current new-contract work is deferable. This conjecture, as such, is straightforward and MOLDS can accept such conjectural information and work with it simply.

However, such a crisp state is not real. A balancing problem is more likely. In its simplest form, Jones may have a contract capacity of only 33 hours and something has to give. Ideally, there may be a set of rules guiding the allocation of time when it is a problem. Consider a priority system:

Contract No.	Priority	Ideal hrs/wk
1	2	6
2	1	5
3	3	9
4	4	8
		28

This is the current situation with no problem since the 28 ideal hours are less than the 33 hours capacity. Adding two new contracts though, creates this:

	Contract No.	Priority	Ideal Hours
	1	3	6
	2	1	5
	3	4	9
	4	6	8
New	5	5	5
New	6	2	5
			38

The two new contracts revise the priority structure and make the total higher than capacity. A possible set of rules for cutback might restrict the top two to 10% cutback each with the remaining cutback pro-rated among other contracts. The arithmetic here is:

38 - 33 = 5 hours to cut back

Total hours on top two priority contracts (No. 2 and 6):

10% x 10 = 1

Remaining hours to cut back = 5 - 1 = 4

Hours on remaining contracts (Nos. 1.3,4,5) = 28

Pro-rated cutbacks:

Number	Ratio	x 4 = hours	Final Allocation
1	6/28	.9	6 - .9 = 5.1
3	9/28	1.3	9 - 1.3 = 7.7
4	8/28	1.1	8 - 1.1 = 6.9
5	5/28	.7	5 - .7 = 4.3

These rules can be entered using the above logic as a guide or structured more formally as: *

$$\text{for } \sum_{i=1}^{M} h_{ij} < C_j \qquad h'_{ij} = h_{ij}$$

$$\text{for } \sum_{i=1}^{M} h_{ij} > C_j \qquad h'_{ij} = .9h_{ij} \text{ for } i = 1,2$$

$$h'_{ij} = \frac{h_{ij}}{\sum_{i=3}^{M} h_{ij}} \left[\sum_{i=1}^{M} [h_{ij} - C_j - (.1) \sum_{i=1}^{2} h_{ij}] \right]$$

* For simplicity of representation, a renumbering of contracts has been assumed here.

where h_{ij} = ideal hours per week on contract i by engineer j

i = 1,2, ... M, M = total number of contracts

h'_{ij} = allocated hours for engineer j on contract i

C_j = capacity in contract hours for engineer j

Whichever way, MOLDS must store such rules, accept the conjectural data and produce the new allocated hours.

This is idealized, however. Rules are often more vague. The manager may want to try out some loads and note the effect on other engineers or on completion time. The nature of his conjecture, then, along with the conjectural data, is an input. This nature may be standard, in that the form of the manipulation is stored, (convenience set) or it may be specially fashioned by the user. These two matters - standard manipulations (using existing models) and user-originated manipulations have been discussed above and are continued in some detail below.

Conjecture is one of the nobler managerial pursuits. MOLDS aim is to make this fast and, therefore, more extensive. The computer, here, is in its element as a high speed manipulator of complex and massive sets of numbers. In this role, it permits the user-manager to pre-live a wide variety of his systems experience. Systems projecting is another name for conjecturing - projecting the system's behaviour under differing sets of environmental facts and system reaction.

5.2 SIMULATION AND PROJECTION

5.2.1 Some Names and Definitions

This process of manipulation on a conjectural basis has given rise to several terms and names which might well be clarified to some extent now. Related to this are the terms Model, Simulation, Projection, "Canned" Routines, and Convenience Set Languages. Model refers to the depiction of a system usually in mathematical or computer language form. It is a static concept; it awaits data and commands so that it can act as though it were the real system. Simulation is the act of manipulating a model so that there is a "simulation" of the real live system. It is customary to reserve this manipulator process for a particular kind of computer model often referred to as "simulator". Projection refers to the process of simulation or mimicking under some forecast input. It gives rise to another conflict in our terms - forecast and prediction. A model can be thought of as a prediction, a prediction of the way elements of the system will act and interact. A forecast is conjectural and relates to environmental data often in probability form. While these are not standardized definitions, they are useful and approximately the conceptions of those in practice. Projection, then, remains as the manipulation of a model with forecast or conjectural data. A "canned" routine is a manipulation process which has been standardized. One might speak of it as a model - it might also be called a decision rule. It is a process of data manipulation which is carried out automatically in a certain set of circumstances. All of the instructions for its execution are on file so there is no need to re-write or re-program or re-construct. The simplest version of this is the instruction for calculating an average - rather than write out exactly what is to be done each time, one merely uses the term "average" to convey the instruction. Models plus information processing routines can be thought of as comprising a set of convenience or problem oriented languages, as opposed to the set of general purpose languages which may be awkward and inefficient from a particular user's standpoint.

5.2.2 Models and MOLDS

No matter what the name or how fashioned - standard or do-it-yourself - the collection of models available to the user-manager will be ever-increasing. These models must be properly coupled in operation to MOLDS. This, in itself, raises some problems since different models have different input characteristics and requirements; some thought must be given to the possibility of standardizing their couplings to avoid confusion. Given any development or growth in models, there is the ever-growing problem of what to use, how to use it, and when to use it. It may well be, also, that the models can be used in part and not in whole. While these models give added power to the conjectural processes, they also require greater knowledge for their effective use. It is not hard to imagine an accumulation over several years of models, some of which may fall into disuse. Periodic review and cleaning out of the "stable" seems unavoidable.

- The Funds Position Simulator -

To gain some idea of the problems which must be met, it is appropriate to discuss some of the available model types in the R and D situation. Here modeling has not been as extensive as in manufacturing and distribution where there is a relatively large body of modeling knowledge available. Some models with which we have specific familiarity and which are the object of current work can display characteristics of this model coupling process. The funds position simulator is a model which calculates the future position of available funds. It accepts conjectural information on period-by-period funds releases as well as contract expenditures. It then sums these up and carries out other routine manipulations to produce a balance for the future periods in question. The simulator does this very rapidly, permitting the user-manager to try out fifteen or twenty arrangements of expenditure and receipt in the course of an hour or so. He would use this at budget times, while at other times special events might suggest a change in the budget pattern. As powerful as this technique is to rapidly scan future possibilities, its use still requires a certain routine effort. For example, a system with eight funds sources and 55 contracts, looking ahead over six months requires for simulator input 48 units of funds release information (eight sources times six months) and 330 units of contract expenditure information (55 contracts times six months). For simplified input, the simulator can accept rates (say, ten percent of $50,000.00 each month on source two), but it still requires some thought and some detailed understanding on the part of the user-manager. The system should permit him to present his conjectures and modifications of them in the simplest possible way, but even after all possible simplifications have been made he is still faced with the need for inputting these amounts of data. Although this may be done in blocks he is nonetheless dealing with the basic elements.

- Waiting Line Models -

Several waiting lines system models are available. A waiting line system, by way of explanation, is characterized by serve and serving elements such as a product (served) and an operation (serving). The served units typically demand service sporadically, and this results in clusterings for service with associated clusterings in the servicing operation. The serving units usually do not elect to staff for the peak clusters and a waiting line of served units develops during such clustering periods. The manager is interested in exploring the results of some different staffing or the impact of some different demands. Waiting line models make certain assumptions regarding the character of the demand and the service and predict the waiting line characteristics of a proposed configuration.

The standard waiting line models deal with both the infinite source and the finite source systems. The infinite source typically has no cut-off, theoretically, to the number of arrivals. The case of telephone calls coming into a switchboard is a typical infinite source example. The machine servicing system is a typical finite source example - the machines "arrive" for service, but there is a limited number of potential arrivals which is the number of machines in the system. Assuming certain distribution characteristics of arrival rate (Poisson) and servicing time (negative exponential), closed form solutions exist for key characteristics such as the average length of the line, the average wait time, and all probability statements regarding the number waiting or for how long. The user must identify inputs: the average servicing time, the average arrival rate, and the number of servers. That is for the infinite source model - the finite source model, in addition, requires the number of served units in the system. Here, the inputs are relatively simple but there is the problem of identifying which model is appropriate and there is the larger problem of interpreting results. The underlying assumptions of these models are never exactly met and while there are convenient and practical interpretive techniques they still must be known by the user.

In the event the system is not a simple served-server arrangement, but instead has some additional complexities, then there is a special purpose queuing model available.* This is currently geared to the finite queuing situation typified by machine servicing where there are two skill levels required. This has certain general purpose applications and represents a two-station queue network system. The user has more flexibility with this model in that he can deal with two types of serving and he can manipulate the distributions of all units of both elements of the system. This additional power, however, requires additional user knowledge and again points up the educational requirements associated with effective use of MOLDS and its models.

*Programmed for the SURC IBM 1620.

- Analysis of Variables -

Another available model deals with the analysis of variables. A large universal program exists for both regression analysis and factor analysis.* Regression analysis permits the prediction of a variable, say Y, from a set of explaining variables, say X_1, X_2, etc. The general linear form is as follows:

$$Y = A_0 + A_1 X_1 + A_2 X_2 + \ldots\ldots + A_n X_n$$

The program, of course, has the capability of developing non-linear forms as well. In factor analysis the interest lies with the explaining variables. The process reduces the dimensionality - to express, say, 35 Xs in terms of, say, 6 Xs called factors. These factors may or may not have operational meaning but they usually result in a more tractable problem. The procedures go on from there with various other tricks in a multi-variable system all aimed at exploring relationships and structuring the system.

- Models to be Explored -

The above models and standard modeling capability have all been specifically experienced in the MOLDS context or in direct association with it by project staff. There are quite a few other models of probable applicability in the R and D situation with which the staff has had experience outside the MOLDS context. Among these are matrix algebra models, linear programming, the assignment algorithm, calculus of variations techniques, and PERT. Of specific interest and worth comment in this collection is PERT. The relatively simple concept that PERT represents - the concept of a critical path through a time network representing portions of tasks - is not a difficult one to grasp. Its remarkable non-use lies more in execution and it is an ideal computer application model. Designed for the highly complex system, it now needs to be made operational for the semi-complex and reasonably simple R and D system.

Of consuming interest to observers on the management science scene is the potential of cathode ray display equipment and its correlative pictorial and diagrammatic communication capability. There are many management problems,

*Called TSAR, programmed for the Syracuse University IBM 7074.

notably the scheduling problem, which are of a permutations nature. Digitalizing these has proven to be cumbersome even with the fastest and largest computers. The manager would like to see a variety of alternatives and judge the best from many points of view. It is not a simple matter to establish the criteria for the several values involved in some of the simpler scheduling situations. As a consequence, the schedule board in many organizations exists and outdoes the computer. A man's capability of scanning alternatives in his mind's eye is considerably more rapid, still, than what the computer can do on a digital basis. So we can see the cathode ray display as a means of manipulating a schedule board with greater rapidity than it now can be manipulated. Capability of display and communication through pictures joins the best in the human and in the computer for certain kinds of problems, and it should produce some startlingly effective results.

- Technical Coupling -

The discussion on models so far has aimed at indicating the nature of such models and some idea of the operational coupling. By the latter we mean the kinds of inputs that are required and the kind of knowledge which the user must have for effective utilization. There is a problem of technical coupling as well. In other words, these models which are not necessarily self-generated and which exist in an outside body of knowledge must be incorporated in the MOLDS program. If there were just a few of them there would be no particular question - they would be reprogrammed in the hardware of the particular MOLDS installation. However, there are more than just a few and the body of knowledge is large. Moreover, it is not going to stop growing, and there will always be this problem of absorbing (as well as eliminating) new models and procedures. The technical coupling problem, therefore, is raised with an eye to the future more than just the immediate difficulty.

There are several alternatives. Re-programming was mentioned above - this would put the model in the specific mode and on a real-time basis. On some of the more complex offerings, however, this might turn out to be a fairly costly effort. Moreover, it may exceed storage capacity although this is something to be concerned with less and less as time moves on. If it stays on the facility for which it was initially programmed it is then in an off-line position unless there is a link between the facilities. In the event that it is infrequently used but important when needed this linkage with the larger unit is a strategy to be considered. On the other hand, the programming problem is one which will have to be licked if this matter is to occur quite frequently. A case can be made for need of on-line availability. Models and procedures are more likely to be used during a creative period and there is strong sentiment to favor immediate feedback from an on-line real-time system to keep the creative spirit alive. This admittedly does not happen now, and life seems to go on; but we don't really know how much we are missing, in one sense. Technical coupling is a vital consideration from many points of view.

5.3 CONSTRUCTING MODELS

5.3.1 The Minimal Set Approach

We return, once more, to the do-it-yourself considerations. As noted several times above, the minimal set languages, such as SLAP, have a built-in capability for constructing fairly complex chains of operators which we are calling models. The unknown regarding such an approach is entirely operational. This has two parts to it - on one hand there is the problem of effectively and efficiently piecing together the appropriate minimal and previously developed operators. It is not known yet how the minimal set approach will proliferate operating elements. At any point in time, the user may well have a fairly large collection of different size building blocks with various ways to piece them together to produce the particular operator of immediate interest. Just how efficiently he accomplishes the current modeling depends on his awareness of the elements that are at his command. Of course, it may turn out that efficiency is not critical - that the process is done far better inefficiently than any other way at all.

Another operational concern relates to a user's system concept. Availability of language does not guarantee articulation. The user may have a certain feel about his system yet not be able to express it. Operational studies will bring out whether or not this is meaningful. Such operational studies - studies of the user in the process of working with the modeling process - will also examine the "re-discovery" tendency. Considering the wide range of modeling already done on the operational management scene, one cannot help but wonder how much of this knowledge will be reformed by those not completely aware. This reforming, admittedly, may turn out to be an excellent educational vehicle - perhaps the only possible one.

5.3.2 Simulation Languages

Moving one step or so away from the user on a do-it-yourself basis, we have a category of languages still requiring some programming instruction for use. While not as demanding, if that is the word, as FORTRAN, they might well be more than the casual user would elect to learn. They are effective in special purpose situations, however, and offer a powerful off-line delegated modeling capability. It is fast programming, so the loop is short even though off-line.

There are three such languages with which the project staff have been working. There are many such languages in existence and more being developed. They all have hardware and special purpose orientation and accordingly varying degrees of simplicity. One of the earlier languages, SIMSCRIPT, developed by H. Markowitz of General Electric and RAND has been reviewed but not worked with to any extent. It is a very general purpose systems simulation language, not exhibiting some of the conveniences in later developments. A language which has been specifically worked with and adapted to the SURC 1620 for consideration in the MOLDS program is SOL. It stands ready for use and will be part of the continuing development tasks with MOLDS. It is similar to a general purpose simulator developed at Burroughs and California Institute of Technology. A special purpose simulation language for queuing networks has been developed by IBM (the General Purpose Systems Simulator III) and is considerably more compact than SOL. It is, however, directed to a much smaller systems situation. The goal must be to find the niche for these languages and learn more about evaluating them as their successors emerge.

5.4 SUMMARY NOTES AND TASKS

5.4.1 Conjecture and Its Needs

In this section we have emphasized the importance of the conjectural process in managing. This is the process that permits the manager to pre-live his experiences so that while there may be distasteful events there will only be a few surprises. The successful manager, perhaps, is the one who can plan his disasters. Of course, the corollary thought is that such pre-living permits extremely effective exploitation of opportunities.

System models are needed for conjecture. Since the conjectural process is highly creative, it is desirable to have models, as well as data base information, right on hand. A real-time on-line arrangement would seem to be a new enriched way of managerial life.

There are many models in existence relating to managing activity in general and having relevance to the research and development situation in particular. Some of these have been explored by the project staff while others, such as PERT, are slated for early study. Display capability as a means of modeling combinatorial problems, is extremely interesting. The list can go on and will go on - the manager and observer of a MOLDS system are faced with the problem of selection. Not only that, there is a problem of operational coupling - what are the inputs and what does the user have to know? Technical coupling is another problem - we must know how to add outside capability to the system software for an on-line arrangement.

Besides the wide range of existing models and knowledge, the user has a very real capability of building his own or tailor-making certain procedures or models for his situation. While this demands greater system comprehension in a formal way on his part, there do exist ways whereby this understanding can now be expressed and, by exercise, enriched. The user can invoke minimal set languages, such as SURC's own SLAP, or delegate such use. Moreover, there is a range of simulation languages which are not as complex as FORTRAN but do require a programmer. Reasonably rapid programming is possible so the user is not off-line for long periods of time after establishing in his mind a need for some new construction. While the do-it-yourself process is more lengthy and difficult to conceive in model building, the capability is technically there as it is in information retrieval and information processing.

5.4.2 Task List

All of this suggests that certain things be done at this point to make this conjecturing dream a reality. Technical coupling is one such task - we must learn how to add knowledge to the system with sufficient ease to make it attractive to consider outside knowledge. While this is directed toward very specific bodies of knowledge and techniques such as models, it may extend to a larger area of languages and concept. Perhaps this task is really one of establishing a programming strategy for change. There will be continuous innovations and the whole program must be able to move with these innovations. At the operational end, there is the obvious task of examining more models, the accumulated knowledge of those in management science, and whatever other disclosures there are relevant to the R and D situation. From this task could come guides on the selection and appropriateness of certain convenience sets at the modeling level. A final and vital task - as usual - is the evolutionary and monitoring task. Only by watching users in the action of conjecturing can we gain more insight as to the various features of this part of the program. As with information processing, we would want to present a particular operational situation with a good initial convenience set, converge to the unique convenience set of selected and developed models, and respond appropriately to future opportunities of development and expansion. We must learn not only what a manager is quite likely to want, but also what are good evolutionary procedures to accelerate this approach to some expected level of effectiveness.

SECTION 6. THE USER-MANAGER

6.1 TOWARD AN R AND D MANAGEMENT MODEL

6.1.1 What a Manager Does

Up to this point we have been discussing the possible. Now it is appropriate to turn about and talk on what is probable. This is dictated by the user: what he does; what he has to do; his personality. In this section we want to touch on these matters and develop a program of implementation. On one hand, we have the implements, the wide-range capabilities that have been set forth in the discussion up to this point. On the other hand we have the user with his characteristics and needs. The job now is to get these two together in the most effective manner.

The subject of managerial activity is widely written on. Any paraphrasing or truncation can no doubt bring criticism. Yet, in the spirit of practicality, we want to set forth another skeleton, this time of managerial activity. In the broadest sense we can identify two basic managerial considerations; control (executive), and improvement or design. Control is defined as dealing with the system as it is, while improvement is seen as upgrading the elements of the system. Making out a schedule would be a control activity; studying the advisability of a new machine in a testing lab is an improvement activity. Laying out a five year plan is a control activity; discussing an engineer's performance with the aim of upgrading it is an improvement activity. While these categories do not represent some ultimate structuring of managerial activity, they do prove useful in the MOLDS context, for the nature of the use of MOLDS differs depending on the activity category.

6.1.2 The Control Activity

Control may be described as "accepting performance measures, comparing the measures to standards, and taking appropriate action, if any". In other words, control is sensing, evaluating, and responding. Now, while this might be said of all of life, the trick in designing a good control activity is in knowing how to respond in a way that puts the system where you want it and, most important, in establishing rather clearly where you want it. The latter has to do with the values or the performance standards that are established. All systems have this control operation, but with different degrees of effectiveness. Modern technology and knowledge seeks to make this ongoing process better – to move the system back onto a well-defined track with a minimum of effort if it is inadvertently pushed off the track by environmental circumstances. Better yet, it seeks to anticipate environmental circumstances so the system never leaves the well-defined track.

- Value Concepts -

If the real problem is in the specifics and not in the generalities, it is appropriate to look at this matter in a specific context. Here the specific context is a research and development organization, a kind of organization which primarily has a collection of tasks to be accomplished. There is also general activity in the procurement of these tasks as well, and sometimes in such routine operations as testing. But the bulk of the effort is related to these tasks. Three main value concepts can be identified: Cost, Response Time and Quality. These are likely to be considered independently since the trade-offs are not well established. For example, on a specific task a top-grade engineer will cost more, do the job more rapidly, and do the job better than a lesser one. Is the reduced lapse time of project and the better quality worth the incremental cost? If the contract set forth specific measures of quality and related these to payment and made similar statements with regard to time of completion and payment, then trade-offs would be established and the matter could be said to be a single value matter. While time is sometimes included in contracts, and quality is alluded to (regarding the performance of some resulting mechanism), these three considerations are by and large not explicitly related.* This means that any task must be evaluated along at least these three lines. It is a multiple value situation with no simple good-bad axis.

- Value Measures, Cost -

Not only are there several value considerations - but also there are several possible <u>measures</u> that suggest the location of a value on some high-low axis. We want to look at those possible measures and consider, after that, the nature of the

---

*There exist some examples of contracts wherein the cost-time-quality relationships are made explicit. The Martin Company's Titan III program is an example of how such a contract works, and is discussed in amusing fashion in "Move Up in Space - Win Points" (Business Week, 25 September 1965, pp 92-103).

action that might be taken to put things to rights in the event that value standards are not being met. Starting with the more tractable cost consideration, it is fairly standard practice to track the time and hence the money, spent by the people assigned to the tasks. It is also fairly standard practice to track other expenditures on a particular task or contract looking at supporting documents such as purchase requisitions, shipping tickets, and so forth. Direct cost is fairly well measured in any R and D situation. While there are sometimes problems in the accuracy of time reporting, in the allocation of expenditures on jointly purchased equipment or supplies, and on the allocation of indirect labor and overhead, these are problems which can be resolved and are not insurmountable or requiring of brand new approaches. Sensing cost, then, is a well-established and formalized process - whether or not it is effective as a practice we will discuss below in looking at evaluation.

- Value Measures, Response Time -

Response time, or progress towards a stated deadline, is not so straightforwardly sensed. One simple measure, of course, is elapsed time from the start of the task. This, however, does not give a sense of proportional completion which is usually the measure of interest. Ideally, the task at the outset will be broken into subtasks with time deadlines and proportional time statements for each subtask. This is not frequently done nor is it a customary practice to reproject the task proportions as the work progresses. This is not to say that reasonable evaluations are not made - on evaluation, there is comment below - it is saying, rather, that there are not now formal procedures well established for the measuring of response time status. One modern breakthrough on this is PERT, but this is remarkably unused.

- Value Measures, Quality -

Quality considerations are even more elusive. Part of this difficulty lies in the definition of the term "quality". On one hand, it may be referred to the performance characteristics of a particular piece of hardware developed by the task effort. On the other hand, it may refer to the "goodness" of the theoretical or conjectural work reported on in the course of the task. In the former case, when dealing with a bit of hardware, some of the traditional quality control concepts of manufacturing can be invoked. There can be the testing of components and the testing of subunits of the final system, and so forth. While such situations lend themselves to formalized information, it is not general practice to have in-process reports and summaries with regard to such quality matters. Now, in an informal way, there may well be adequate sensing of status of component quality but it tends to be on a personal and non-formal basis.

When the task is not producing hardware but instead some concept or system or theory, then quality measures are difficult to obtain. The quality is the closeness of match to specification; this is difficult enough to ascess at the end of the task, and quite involved to determine as the task is progressing. Here again, there are no customary formalities for quality determination - it is left up to the supervisor or other managing interests who have an opportunity to review and control the task. Quality measures, then, in the R and D situation are not customarily generated - they are quite possible to obtain when the output is hardware, but they are difficult to obtain and require considerable insight when the product is software or concept.

This casual rundown on the sensing apparatus requirements or characteristics in the R and D situation is but a beginning on the complete job. Given the need for specific measures - and this will be commented on in a moment - there still remains the hard and imaginative job of setting up an intelligence system where none has before existed. But it is possible; modern techniques in statistical sampling and handling of judgement measures coupled with the new dimensions of power, computer technology, give this measuring problem great hope for good solution.

- Evaluation -

The important thing, though, is to establish the need. We have up to this point talked about the value considerations and the associated measures that might possibly be helpful in such considerations. But, measurement for the sake of measurements is not necessarily good, even though new technology makes it possible where it was impossible before. Measurement carried to its ultimate can result in a noise system, not an intelligence system. With only casual attention to the need, it could be called an information system - but there is a difference between (possibly irrelevant) information and a control message. A control activity needs certain messages which can be obtained from a correct setup in the sensors coupled with the filtering action of the evaluation phase.

The measurement needs, then, can be said to be affected by the nature of evaluation processes. But evaluation, in turn, depends to some extent on the action - on what can be done about a good or bad situation. These parts of the control activity - testing, evaluation, and response - are all interrelated, and we need to talk about integrated design of a control activity. To work on this more specifically, consider the effect cost evaluation has on the cost measurement process. If it were decided to evaluate the total cost progress of the task against some standard or ideal progress, then reporting of time by individuals could well be done on a gross basis. That is, there would be a reporting of time against that task and no additional detail. On the other hand, if it were decided to track different subtasks within that task, then the reporting detail would be much greater. Each individual would be required to note the time against a particular portion of that task. At the other end of the scale, it may be decided to evaluate cost across a group of projects, in which case the time recording would be against this grouping. In this simple way the beginnings of a more lengthy argument are shown on how the evaluation affects the measurement process. Having the capability of measurement does not necessarily mean that that measurement need be invoked. Technically there is the capability for an individual to log minute-by-minute activity during the day. Intuitively this has not been considered necessary in R and D situations, while on the other hand it has been considered appropriate in some manufacturing situations. The structuring of the intelligence system - the sensing and the evaluation portion of the control activity - requires a good look at both measurement and evaluation so that unnecessary measurement is not undertaken and so that impossible evaluations are not requested.

- Some Action Considerations -

While we are talking specifically about the control evaluation ideas with respect to cost, quality, and response time, it is probably appropriate to note the relationship between action and evaluation. For just as non-appraisable measurements are inappropriate, so are non-action evaluations. In other words, it may be determined that the cost situation is "not good" but there may be no clear line of action suggested. This raises some basic questions as to whether the situation really is "not good" if there is no remedial action possible. More likely, the conceivable action is only appropriate under impetus of a more severe condition.

Consequently, the evaluation of "no good" was inappropriate, premature - the evaluation system was too sensitive relative to the action that was possible. So, from the point of view of a control, no particular purpose is served in over-sensitive systems. Other purposes may be served in such sensitive systems (the process may keep certain groups in better operating state or may keep certain people "happy"), but from a control system and control activity point of view, over-sensitive parts create noise and confusion and possibly inappropriate action.

Moving back to the specific R and D situation, the cost evaluation technique has already been suggested. Tracking costs progress against predetermined standards of technical progress is a fairly conventional method. Evaluation depends a lot on the establishment of these standards, and the standards, in turn, depend on the action which can be taken. In the last section is a reference on such a controlling technique applied to manufacturing situations. It suggests that an arrangement of boundary limits can be applied to, for instance, this idealized progress where the limits are indicative of actions which can be taken. If cost is getting high, one has for alternative actions the early termination of the task, the search for additional funds, or the reduction of intensity of that activity. Any evaluation process has associated with it these alternative actions that are suggested by the measures, so that the good measures and somewhat understandable actions in the cost phase tend to result in a reasonably tighter concept of control as far as cost is concerned. In the quality and response time areas, however, the evaluation process is considerably more difficult owing to measurement problems discussed earlier. Not only are we faced with uncertainty as to whether or not the system really has the indicated quality or elapsed time, but also complete uncertainties as to what in the world we are going to do about it if the system does not conform to these notions. The action may well take the form of a defensive preparation, like getting the buyer ready for something different from specifications with a different delivery date. It might mean the discharge of somebody involved in the task or the addition of a specialized talent to bring about the technical content set forth in the original task specification. It is not the convention, now, to specify these alternative actions in advance - there is a strong tendency to improvise. Nonetheless, the concept of the evaluation based on action possibilities is there and the computer simulation techniques plus certain modern logic concepts can help shore up this evaluation and action phase of the control activity.

Effective action plans depend on knowledge of the system, knowledge which can only be formalized through models. Model manipulation can explore the impact on value measures of proposed actions. This is answering the "What ... if?" question introduced earlier. It should be noted that models usually have some elements in them which reflect status quo particularly with respect to research methods. While not specifically a topic for MOLDS there may well be some spin-off of managerial experience with certain models and the computer to the scientists and engineers. Conning an existing system cleverly is one way; conning a better system may be easier.

6.1.3 Improvement or Designing

While managing often seems to be preoccupied with control or tactical problems (execution), it more rightfully ought to be in the area of improvement or designing or conjecture. In discussing the control activity at some length, many points relevant to this were unavoidably made, since the control activity itself is subject to designing. The long range considerations which manager takes specifically to alter the status quo and particularly to change some existing elements, require some framework of understanding. All of the discussions regarding models in Section 5 and the references in the earlier part of this section regarding action pertain here. The impact of a new piece of equipment or an engineer's improved performance through education is what a manager seeks to assess. And, as has been noted on more than one occasion to this point, the model is vital in making such assessment and in predicting an outcome.

There has been considerable discussion about formal models of operational aspects of the system. Conjecture is one part of a designing process in that an alternative is being proposed. The word designing comes from the Latin "designare" meaning "to select" from a collection of possible alternatives. While conjecturing is artistic and value-oriented, the execution of the model with that conjecture is technical. The assistance that a rational program of on-line computing capability can give is in extending the execution capability of such formal models.

There are many situations, however, where the modeling is non-formal and where the designing process does not move along as crisply as defined above. We would be quite hard put to specify the result of an engineer's night-time Master's Degree program. We do not know the relationship between study and knowledge accumulation and performance. It may well be that the matter does not require specific information; it may be that the approximate measures, even considering all possible error, indicate only one course of action. While we do not offer much in the way of formal models in such situations, it nonetheless seems important to note that non-formal models, perforce, exist. When the greater insights of the future can articulate these models, technology must be ready to execute them in all their possible extensiveness and complexity.

6.2 THE USER

6.2.1 As An Individual

The job requirements have been touched on - what of the user himself? We have some sense of what the system needs from him, against a background of what can possibly be done. Now we can consider possible restrictions to full use of technology and possible shortcomings of that technology suggested by the "state-of-the-user". One conventional method is to look at a "typical" user and chart a variety of personality characteristics; it is also a thought to idealize the user. This has not been done to any reportable extent so far in the study and, in fact, it may not be a necessary task. Besides tending to be unproductive,(such studies are always somewhat vague when it comes to action recommendations - idealized standards are obtained calling for personnel apparently a mix of Leonardo da Vinci and Flash Gordon) this direction is argued against by the very practicality of MOLDS. It is literally usable by everybody no matter how normal or bizarre; its evolutionary character permits it to be tailor-made for the particular user in his particular situation. Resulting system states may be interesting later on. The wide variety of user characteristics may be a later testament to its virtuosity. Or, more appropriately, such virtuosity has been and remains a specification of MOLDS.

6.2.2 Language and Concept Position

While the personality of the user may not be too relevant, some of his intellectual characteristics are. This is probably more a comment on the range of MOLDS possible users; for on the one hand a user adept in and interested in language will fashion inquiries and models with his own hand using the general purpose languages, while on the other hand one with no formal experience or interest in language will rely on the "convenience set" of languages, the standard inquiries and established models. This language characteristic is recognizable to most. One with an exact sense of meaning and a precision of speech coupled with a strong propensity for logic might well be one with language interest and, possibly, experience. Conveying a thought is satisfying to him and he will no doubt respond to the general purpose languages. Many in managing roles, however, are not given to such precision and may not ask the same substantive question twice the same way, using the same words. Their speech, no matter how effective, is not predictable and minute twists of logic are not of interest to them. Such a person will not respond to the general purpose languages and will not find in their generality the informational and conjectural power necessary for the user's conduct of the management or inquiry.

- The Systems Concept -

Neither of these types (and their description is admittedly not in the best behavioral terms), are, perforce, failures or successes. One other characteristic is dominant, that of systems concept. The precise logician may ask irrelevant questions and make myopic conjectures albeit quite accurately and with considerable technical skill. The imprecise "mover of people" may invoke the wrong "convenience" term and flamboyantly steer the whole effort off the track. Or, they may both be lucky; or, more hopefully, they may both have a deep sense of the concept of a system.

This system concept characteristic is also recognizable although the following may not be in the best language of psychology. A move-by-move chess player lacks such a sense of concept - the one who senses overall strategy and sees each move in the large plan has such a concept. There are degrees, of course. But, there must be awareness of the values involved, of the control process described earlier - measurement, evaluation, and response - and of evolutionary operations in general. The full power of MOLDS in routine hour-in and hour-out use can only be realized as this awareness exists. It seems unavoidable, too, that different practices in language will still result in similar patterns of control and improvement activity. With a strong system concept, each type of user will establish his own convenience set either self-made or self-selected from some existing and developing body of knowledge. He will collect his own routines for inquiry and conjecture. Given a similar target system these routines are bound to be somewhat similar also. Since MOLDS deals with operational matters primarily, target system similarity is understood to mean operational similarity - same size, same number of contracts, same field of endeavor, and so forth. Such a similarity of routines is not critical in the development of MOLDS, it is just an observation about what may well happen.

This systems concept does not always exist with a user. With old technology it may have been frustrated to the point where it withered, or else it just may never have developed in the user. MOLDS offers a vehicle for education as well as implementation, all in keeping with the evolutionary aspect of the program. There seems to be no doubt that better language facility and more insightful systems concepts will emerge in the manager just from the use of MOLDS. But the responsibility to educate the user is there, and the program of MOLDS installation requires that such be incorporated.

6.2.3 Technology's "Shortcomings"

While we have been talking on the user and possible variation of styles, it seems appropriate to look at the other side of that picture and comment on technology's shortcomings. There is no attempt here to categorize such shortcomings for they all seem to lie in the direction of communication with the computer. One problem type, noted in Section 5, is the combinatorial problem such as that raised by scheduling. The computer does not have a "mind's eye", so the best it can do is to extend visually its results to a person who has such capability. This is now being done through display devices, and probably represents a most important rational breakthrough. It may be the most that can be hoped for until the computer somehow or other achieves a sense of art. It will be some time before pictorial discourse becomes effective. That it is on its way to solution, however, is giving relief to one of the major technological shortcomings.

A shortcoming which is not entirely out of the technical range and vitally important is that of no memory redundancy checks. This is a very poor term; it seeks to identify the need for avoiding re-discovery of old solutions. A good manager usually has a good secretary to remind him that he has already dealt with this matter or to indicate precedent - and that term is probably the most appropriate one. The shortcoming might be called a lack of precedent identification. Law, to some extent, has some of the mechanism that is needed here. An individual's mind is remarkably good in rapid juxtaposition; it tends to be somewhat peculiar in its storage which gives rise to the re-discovery characteristic. A large memory with extremely good functional redundancy checks in close coupling with the juxtaposing capability of a human would be the ideal. The coupling is being worked on with written and spoken language but a great need exists for work on the redundancy portion of the problem. Again, this is poorly stated but even so perhaps still recognizable by those who have been faced with this difficulty and sensed this need.

6.2.4 User Response

The user has the capability of responding to MOLDS in an immediate and effective way. For some this is easy, for others it is hard. A characteristic of the effective user will be a curiosity about what he is doing, leading to continuing and helpful responses. These responses involve altering the form of a language, the actual development of a model, the altering of display techniques, and so forth. This response characteristic is an eagerness to give back to the process as well as use what the process has to offer. It is a feedback concept, reasonably familiar to all. But, it can be invoked with such power that it may take skill to use effectively. The slow-to-respond, or withdrawn, will be ineffective; the over-responsive, equally so. This trait of learning, of participation in one's own way of doing things, of curiosity - what you will - is vital to MOLDS progress and the progress of the user. Apathy and compulsive tinkering are to be avoided; healthy curiosity and response are the goals.

6.3 IMPLEMENTATION

6.3.1 The General Scheme

Implementation of MOLDS in any situation is seen fundamentally as an evolutionary operation. This thought has been noted again and again; here we will try to put some flesh on it. Implementation, though evolutionary, must start, so there is the need to establish starting points, appropriate packages of procedure to be used as points of entry or initiation. These packages will possibly be altered or even completely eliminated in the course of evolving some eventual system. The possible nature of such starting packages is set forth in The Initial Step, below. The Evolutionary Process discusses the tracking procedures, educational programs, and corrective actions. Section 6.3.4 notes a few specifics regarding SURC and RADC, both current and proposed.

To be re-emphasized is the fact that any situation, no matter how small or apparently trivial, is an appropriate experience at this point. MOLDS is technically feasible: its operational characteristics need to be developed now, along with, of course, additional technical development. Ten or so different installations in different organizations of any direction - R and D, military, commerce, manufacturing - are going to be required before there is a good feel for the implementation process; different activities, different organizations, different people are basic grist for this mill of understanding we are now operating.

6.3.2 The Initial Step

An initial package must include basic retrieval and basic intelligence-control programs. Elementary retrieval, in an R and D situation, would use a data base from existing documents covering cost and personnel allocation, and only the core of these at that. Where manpower cost is the major cost component, the weekly activity sheet is a basic updating document. Contract cost status, usually found on a document similar to a ledger card, is the initial document updated by the activity sheets. The core concept can be invoked by noting initially only those contracts comprising, say, 80% of the dollars, possibly around 30% of the contracts. The others can be picked up as time goes on and included into the system. Personnel deployment is often indicated on a contract loading and information sheet similar to RADC Form 77. This is a basic document for initial storage; again, it may be wise to concentrate initially only on contracts with the large proportion of dollar involvement.

Some basic processing of raw information is part of an initial package and also some elementary evaluation capability to yield a primitive intelligence system. Appropriate for inclusion are averaging (totaling), classifying, profiling, and histogram generation. Major contracts can have some progressive cumulative cost limits set, possibly arbitrarily at 10% above and below the line of balance, and one output routinely generated showing only departures from these limits. This is a very limited intelligence scheme but it does start the user moving on all fronts of MOLDS; it sets him up for the upcoming evolutionary process. It may be possible, in the latter case, to establish some response or decision rules in the event limits are violated (such as a reassignment order for personnel or low-priority tasks). However, initial installation should not be held up for that twist ; it will come later in the ordinary course of events. At the outset, moreover, since the limits may be arbitrary, full reports on contract status would run parallel to the exception report. Eventually, such full reports would drop out only if specially requested.

Such, then, is the character of an initial package. It is a full representation of MOLDS, albeit not a complete one. An analogy to this is the underspoked wheel, a primitive wheel for slow non-stressed rolling. It ultimately evolves to the fully-spoked wheel for high-speed, effective carrying. In its initial state, MOLDS permits the user to make inquiries about the state of current matters, it permits him to process data in a limited way, and it permits him to get an evaluated report possibly based on his own evaluation rules. Moreover, he can conduct a limited set of evaluations (comparisons), and make some limited conjectures. It is a full yet not complete program, but it takes that vital first step. Different situations, such as strategic military, may require a different core of documents in such an initial package; and larger organizations may need some additions to the documents, but the core idea is still recommended. It need not take much to get the user on-line, and yet his early on-lineness may return much in keeping application apace with technology.

6.6.3 The Evolutionary Process

Proponents of evolutionary operations on the industrial scene take the extreme view that a manufacturing system's information is more important than its product - almost that its product exists in one form or another through time to produce information. This is somewhat akin to the biological bit of philosophy that the hen exists for the egg. The egg and information, then, are what you are really after; the hen and the product help you get them.

6.3.4 SURC-RADC Experiences

There have been some beginnings with respect to this implementation and monitoring. These have taken place at RADC and SURC. At RADC the Funds Position Simulator is close to being operative on the 1604. While this may not be in an on-line position at the outset every attempt will be made to simulate on-line operation either over short time periods or using a time-lapse approach. At SURC more activities have been engaged in. There has been a learning program of modest extent with respect to SMOLDS. This has brought about some of the comments noted in Section 3 regarding the character of that language. It was also noted that out of context it is difficult to grasp completely what the learning characteristics are. Along with that too, managers have been tracked with respect to their day-in and day-out activities and the character of the decisions that they make. Some of this work has been reported in an earlier MR (No. 190), and suggests the frequency of invoking the on-line arrangement. The project manager activity is being explored with respect to data base requirements. At SURC we are bringing together the analyst and manager, preparatory to more extensive real-life lashup. A concerted effort to expand this implementation is to be made in the early part of the upcoming project year.

6.4 SUMMARY AND TASKS

6.4.1 The User and Information Technology

This section has been looking at the user against the background of the technology, and it has drawn some conclusions regarding an implementation program. The control activity was described to some extent and the need for formal consideration of values and the need for models were pointed out. The user is caught up continuously in execution which, for effectiveness, depends on some pre-planning or on some effective designing of control systems. And, as noted earlier, he is quite involved in conjecture which is related to the improvement of things. Here again, models are dominant - they offer a method for formal and rapid conjecture.

So, while the user's activity suggests more and better models of real-life, the user, himself, suggests methods for inducing curiosity. The key to any effective system is feedback; MOLDS is no exception and requires user participation in its evolution both generally and on location. It may be that certain people will not exhibit this curiosity and, we feel, not get the most out of an on-line real-time system.

Early implementation seems important to start the evolutionary process going. This is getting to be an old theme by now in the report, but it does not diminish in importance. Just as we need to know more about the various elements to the system, so do we need to know about the user and his personal interaction with it. Implementation requires some initial step with a package of sorts just to get the ball rolling. Then the evolutionary process begins and we, as before, seek effective means for tracking, for education, and for correction. We seek to get a good implementation scheme.

6.4.2 Task List

Certain tasks loom as extremely important. The control concept in the R and D situation and the intelligence subsystem need continuous looking at and operational development. The design of the R and D control system is an unavoidable responsibility. Another task centers on the user experiments. We need to know how to conduct these with the user in mind. We need to know more about inducing feedback, of getting the user to participate more extensively. And the final task, which ties in with all of the others, is in this area of tracking or monitoring or conducting the evolutionary process. The behavioral features of the problem introduced in this section suggest a wider disciplinary approach to this evolutionary process, but, more important than that is its early beginning.

SECTION 7. SUMMARY AND CONCLUSIONS

7.1 THE SIMPLE VIEW

MOLDS is one year old; it is hardly a time for nostalgia but for looking ahead. Work on the birth and development of MOLDS has been varied, almost to the point of appearing non-relevant. This is one specific reason for this report's looking ahead since the meaning of such work can only be considered in certain future lights.

But, even in this format of perspective, the trace has been a long one and there is no doubt a feeling that this is a complex matter. There are many details, to be sure, but in concept MOLDS, at one year, presents a basically simple and traditional problem: the new tool and what we are to do about it. The response is really equally simple and traditional: study the tool's construction, its use, the user, and the business of getting it used. That is what the project, with this report, is all about.

7.2 MOLDS, THE MANAGING "TOOL"

What is MOLDS made of? This was Section 2's theme and therein were discussed computers, display devices, programming and language. Computers are fast and expensive, moderately reliable and large in storage. Pictorial display has just been born; programming is needed routinely and requires special training; and language theory continues to baffle, but is apparently "good enough". In the next ten years we will see inexpensive, somewhat faster and more reliable computers, with vast storage capability. Routine programming by users will turn toward (still non-verbal) conversation made possible by professional, but removed, programmers and clever hardware. Pictorial display will permit two-way picture discourse; language theory will still baffle but nevertheless contribute to needed understanding for language design.

How is MOLDS to be used now and what do we see for later? Sections 4 and 5 dwelt on this matter. MOLDS stores moderate amounts of information; it enables one, with some ease, to get it out "as is"; it can carry out a limited number of manipulations (processing, models) on stored or conjectural information as well as on the manipulations themselves (the do-it-yourself modeling feature). Looking ahead, the new computer storage size potential will up the data base size tremendously and explode the range of possible manipulations. It will be easier by far to retrieve information and direct its manipulation. Moreover, and most important, the creative conjectural process for systems projecting will be progressively freed from bonds of slow feedback and small storage. Display technology, moreover, will extend the user's "mind's eye" capability for dealing with the large collection of combinatorial problems such as scheduling and personnel assignment.

What of the user and his activities? On the R and D managing scene, Section 6 went to some lengths on the control (execution) and designing (improvement) functions. These are both complex tasks needing study and insight to define extensive interrelationships and manipulate large collections of data. Our ever-larger society, moreover, practically insures greater magnitudes in these tasks - larger organizations, many more functions.

The intellectual character, rather than the personality of the user, was seen as indicative of a MOLDS installation's effective growth. The user's curiosity leads to reactions or feedback, basic to the evolutionary process.

Getting MOLDS used - this is the capstone problem. There must be early use in a real-life situation. For this an elementary package is needed, one with at least some of the data base, some retrieval capability, some processing operators, and some elementary models. With such a start, monitored and constantly expanded activity can be realized, and MOLDS will be used.

7.3 THE UPCOMING TASKS

In each section, tasks have been noted. Some of these overlap; the following list and brief discussion put the whole task scheme in order. All tasks are viewed from the point of view of the system, not the state-of-the-art. This first group are technical, capable of being conducted away from the scene of MOLDS in operation; the second group, then, are on the scene and considered operational studies.

A. Document Discipline - A study of forms design and the problems of execution in information origination.

B. Optical Readers - Maintain knowledge relative to the field; study the problem of getting operational characteristics of units so that the niche, ideally and practically, can be determined as developments occur.

C. Minimal Set Languages - Continue with SLAP and modifications; maintain knowledge in the field so as effectively to interpret and "niche" other languages.

D. Technical Coupling - Study ways of adding existing and "other language" models to MOLDS; develop a programming "strategy for change" to handle future alterations and expansions.

E. Display- Maintain knowledge in the field; explore specific function of combinatorial problem-solving through pictorial communication.

F. Organizational and System Models - Continue study of existing body of model knowledge and effect operational coupling to MOLDS; establish evaluative procedures for assessing future developments.

G. Research and Development Control Systems - Study the specific tactical control problem of the R and D organization; use the evolutionary experiment underway with SMOLDS; seek to establish short-range project control via computer-based procedures.

The operational tasks are really one large one: the conduct of an operational experiment with SMOLDS. SURC is the initial location for this effort with certain parts of RADC early possibilities. As noted in 6.3.1, other situations (up to 10) are needed. There are several concerns in this:

1. The development of a meaningful initial package;
2. Specific needs, as experience goes on, in the storage and retrieval process, in elementary information processing operators, and in systems projecting models;
3. The conduct of MOLDS implementations - how to measure and track performance, how to accept feedback, how to tailor-make MOLDS as we progress.
4. The impact of MOLDS - what it does operationally and behaviorally to the organization's performance.

These tasks and concerns represent a lot of work with many skills involved. They are not impossible nor are they, to our thought, trivial. No priorities have been noted but the operational study can be conducted on what is now available and is clearly the most vital effort. In any event, subsequent effort needs more planned specifications which will be forthcoming after agreement on the general ideas.

7.4 MOLDS - ONE YEAR OLD

Looking ahead, at this point, can be done quite well by looking back, at least into the report. For the main impression leading to the major point-of-emphasis has been registered in several ways again and again. The need now is for getting MOLDS used.

Not only can this accelerate specific ' cal benefits but, more importantly, it will set the stag for exercising installations of this evolutionary system. We now know what the elements are; we feel strongly that it can be put to work; we yet don't know, however, how to get it operative. Experience in use and, before that, installation is a fundamental need.

The very power of MOLDS lends urgency to this argument. Technology is creating an operational vacuum, and fast. If orderly exploration is not undertaken then costly haphazard patch-work undertakings will move in. This can be more than costly and worse than disasterous; it can be almost permanently debilitating. A growth effort which wallows in the muck of vaguenes and relentless misdirection creates more and deeper muck and proliferate direct ons. While a good clean disaster is one way out, a good clean assault on order is far better.

An aim is to give information systems the same good shakedown procedures enjoyed with, say, ships and chemical plants. There are many beginnings but they are only that. MOLDS offers a unique chance for the development of the implementation theory. Such theory can serve as a solid base for realizing good information systems in practice. Information systems are jumping into a vastly new era; computer technology lets us do things, literally miraculous things. We can only do these things if we get operational savvy which matches our technological savvy. MOLDS, bringing out the best technologically, must also bring out the best operationally. Nothing is more important than moving on with the business of getting it used and learning how to get it used.

SECTION 8. BIBLIOGRAPHY

8.1 SECTION 1 - INTRODUCTION

Mentioned in Section 1 and elsewhere in this report were the various informal SURC reports generated in connection with the first year's effort. The principal purpose of these reports has been to ensure good SURC-RADC communication; a secondary purpose has been to facilitate control over, and documentation of, a broad spectrum of tasks. Speed of issue has taken precedence over polish in most cases. The complete set is listed below.

- Memoranda for the Record -

A. MR No. 145 - "SURC Prototype for MOLDS (SMOLDS-1)"

B. MR No. 147 - "General Specifications for MOLDS"

C. MR No. 151 - "RADC Information Needs"

D. MR No. 155 - "RADC Form 77"

E. MR No. 156 - "Some Management Analogies"

F. MR No. 157 - "A Note on MOLDS-RESS"

G. MR No. 159 - "Newspaper Article Retrieval System"

H. MR No. 161 - "SURC Prototype MOLDS (SMOLDS-2)"

I. MR No. 163 - "An Output Matrix Proposal for SMOLDS"

J. MR No. 164 - "Newspaper Article Retrieval System (NARS)"

K. MR No. 166 - "Design of MOLDS-1"

L. MR No. 168 - "MOLDS Scope"

M. MR No. 171 - "Some Notes Regarding an Optical Print Reader"

N. MR No. 173 - "NARS Data Base"

O. MR No. 175 - "Proposed Study of the Requirements for Connecting Remote Display Consoles to a CDC-1604-B Computer"

P. MR No. 175a - "Additional Equipment Approaches Pertaining to MR No. 175"

Q. MR No. 177 - "The Impact of a Management On-Line Data System (MOLDS)"

R. MR No. 178 - "Minimal Sets of Operators"

S. MR No. 180 - "Comments on MR No. 166"

T. MR No. 181 - "A Recommendation for Simulation Language Implementation"

U. MR No. 184 - "Comments on ALERT"

V. MR No. 185 - "Notes on ITA"

W.	MR No. 187	- "Specifications for an Ideal Management Tool"
X.	MR No. 190	- "Toward an R and D Management Model"
	MR No. 191	- "Statistics for an R & D Manager
Y.	MR No. 195	- "SLAP Programming Language"
Z.	MR No. 196	- "Preparation of Documents for Processing by the Philco General Purpose Print Reading System"
AA.	MR No. 197	- "Overview of DOCUS and its Applicability to the MOLDS Project"
BB.	MR No. 198	- "Implementation of MOLDS in the SLAP Programming Language"
CC.	MR No. 202	- "SURC Prototype MOLDS (SMOLDS-5)"

- Trip Reports -

A.	16,17 June 1964	- Symposium on Computer Augmentation of Human Reasoning.
B.	17,18,19 November 1964	- Symposium on ME and Large Systems/Visit to Westinghouse.

- Monthly Letters -

A.	6 October	1964	- Monthly Progress Report No. 1
B.	12 November	1964	- Monthly Progress Report No. 2
C.	30 November	1964	- Monthly Progress Report No. 3
D.	7 January	1965	- Monthly Progress Report No. 4
E.	10 February	1965	- Monthly Progress Report No. 5
F.	4 March	1965	- Monthly Progress Report No. 6
G.	6 April	1965	- Monthly Progress Report No. 7
H.	7 May	1965	- Monthly Progress Report No. 8
I.	4 June	1965	- Monthly Progress Report No. 9
J.	8 July	1965	- Monthly Progress Report No. 10
K.	11 August	1965	- Monthly Progress Report No. 11
L.	15 September	1965	- Monthly Progress Report No. 12

- Conference Records -

A.	23 September	1964	
B.	30 September	1964	
C.	28 October	1964	
D.	4 November	1964	
E.	20 November	1964	
F.	9 December	1964	
G.	23 December	1964	- Visit Record No. 1
H.	6 January	1965	- Visit Record No. 3
I.	13 January	1965	- Visit Record No. 4
J.	27 January	1965	- Visit Record No. 6
K.	4,8 February	1965	- Visit Record No. 11
L.	11 February	1965	- Visit Record No. 12
M.	17 February	1965	- Visit Record No. 13
N.	3 March	1965	- Conference Record No. 16
O.	17 March	1965	- Conference Record No. 22
P.	24 March	1965	- Conference Record No. 23
	24 March	1965	- Conference Record No. 26
Q.	7 April	1965	- Conference Record No. 27
R.	23 April	1965	- Conference Record No. 29
S.	21 April	1965	- Conference Record No. 30
T.	12 May	1965	- Conference Record No. 32
U.	19 May	1965	- Conference Record No. 33
V.	26 May	1965	- Conference Record No. 37
W.	3 June	1965	- Conference Record No. 39
X.	3 June	1965	- Conference Record No. 40
Y.	9 June	1965	- Conference Record No. 41
Z.	11 June	1965	- Conference Record No. 42
AA.	14 June	1965	- Conference Record No. 43
BB.	14 June	1965	- Conference Record No. 44
CC.	10 June	1965	- Conference Record No. 45
DD.	23 June	1965	- Conference Record No. 46
EE.	24 June	1965	- Conference Record No. 47

FF.	24 June	1965	- Conference Record No. 48
GG.	25 June	1965	- Conference Record No. 50
HH.	7 July	1965	- Conference Record No. 51
II.	24 August	1965	- Conference Record No. 61
JJ.	29 September	1965	- Conference Record No. 64
KK.	1 October	1965	- Conference Record No. 67

- Technical Memoranda -

A. TM No. 29 - "Microelectronics and Computer Hardware"

B. TM No. 32 - "Error Control in Machine-Machine Communication"

8.2 SECTION 2 - STATE-OF-THE-ART PROJECTIONS

Computer Prophets predictably give voice on certain particular occasions, among which are:

A. The publication of a special technical issue devoted to some computer topic.

B. A particular anniversary of a technical organization.

C. End-of-the-year issues, generally.

D. Upon the writing of conclusions to a survey article.

Some of the references below are attributable to such events; others are more random.

A. Proceedings of the IRE, Vol. 50, pp 529-1448, May 1962.

This mammoth fiftieth anniversary issue of the Proceedings features both summaries of the past 50 years and predictions for the next 50 years over a very broad range of technology. In particular, computer futures receive coverage in several short papers.

B. Sakai, T. and Doshita, S., "The Automatic Speech Recognition System for Conversational Sound", IEEE Transactions on Electronic Computers, Vol EC-12, pp 835-846, December 1963.

C. IEEE Transactions on Electronic Computers, Vol. EC-13, August 1964.

In this special issue on Computer Languages, the editorial is of particular interest.

D. Simmons, R. F., "Answering English Questions by Computer: A Survey", Communications of the ACM, Vol. 8, pp 53-70, January 1965.

E. Lindgren, N., "Machine Recognition of Human Language - Part I", IEEE Spectrum, Vol. 2 pp 114-136, March 1965.

F. Diebold, J., "What's Ahead in Information Technology", Harvard Business Review, Vol. 43, pp 75-82, September-October 1965.

G. Chomsky, C. S., et al "The BASEBALL Program: An Automatic Question Answerer", AD 432038.

H. Thompson, F. B., et al, "DEACON Breadboard Summary", AD 612167.

I. Thompson, F. B., "The Semantic Interface in Man-Machine Communication", AD 612170.

J. Bar-Hillel, Y., "Language and Information", Addison-Wesley Publishing Company, Reading, Massachusetts, 1964.

K. Davis, M. R. and Ellis, T. O., "The RAND Tablet: A Man-Machine Graphical Communication Device", AD 444103.

L. Mathis, S. J., Jr., Wiley, R. E. and Spandorfer, L. M., "Microelectronics and Large Systems", Spartan Books, Incorporated, Washington, D.C., 1965.

8.3 SECTION 3 - THE INFORMATION STORAGE AND RETRIEVAL SUBSYSTEM

The volume of literature being generated these days on the subject of information retrieval is tremendous, and no letup is in sight. Some of the papers are quite shallow, and some highly technical (and therefore inappropriate to this report). Listed below are an index, a paper written in a mode suitable to this report, a paper of the critical analysis variety, and several general references of interest.

A. Fineman, A., and Revens, L., "Permuted (KWIC) Index to Computing Reviews (1960-1963)", Association for Computing Machinery, New York, New York, 1965.

This index, which itself represents quite an achievement as an operational, semi-automated information retrieval system, contains 196 references to reviews of information retrieval articles. These reviews have all appeared in "Computing Reviews" a journal published by the ACM, during the period 1960-1963.

B. Herner, S., "Deciding When to Establish Your Own Storage and Retrieval System", AD 432518.

An insightful paper written from a non-technical standpoint.

C. Allderige, J. M., "Toward a Theory of Design", Industrial Research, June-July, 1961.

D. Dearden, John, "Can Management Information Be Automated", Harvard Business Review, March-April 1964, p. 128.

E. Diebold, John, "ADP - The Still-Sleeping Giant", Harvard Business Review, September-October 1964, p. 60.

F. Diebold, John, "Pitfalls to Business Data Processing", pamphlet.

G. Oettinger, Anthony G., "A Bulls-Eye View of Management and Engineering Information System", ACM Proc. of 19th Nat. Con., August 1964.

H. Proceedings, The Third Annual Conference, Urban Planning Information Systems and Programs, American Society of Planning Officials, September 15, 17, 1965, Chicago, Illinois, Northwestern University.

Hayes, R. M., "Generalized versus Tailored Systems".
Kibbe, J. M., "User-Motivated Software".
Postley, J. A., "Urban Management Data Systems".

8.4 SECTION 4 - THE INFORMATION PROCESSING SUBSYSTEM

The following are some specific on-line information processing references, along with some general references in the field.

A. Culler, G. J., and Fried, B. D., "The TRW Two-Station On-Line Scientific Computer", Chapter 5, Computer Augmentation of Human Reasoning, edited by Sass, M. A., and Wilkinson, W. D., Spartan Books, Washington, D. C., 1964.

B. Dion, F. A., "A User-Oriented Information Processing System", Technical Documentary Report No. RADC-TDR-64-193, Information Processing Branch, RADC, Research and Technology Division Air Force Systems Command, Griffiss Air Force Base, New York, 1964.

C. Hald, A., Statistical Theory With Engineering Applications, John Wiley, 1956.

D. Harmon, H.H., Modern Factor Analysis, University of Chicago Press, Chicago, 1960.

E. Sass, M. A., and Wilkinson, A. D., editor, Computer Augmentation of Human Reasoning, Spartan Books, Washington, D. C., 1964.

F. Wagner, H. M., "Statistical Decision Theory as a Method for Information Processing", Journal of Industrial Engineering, January-February 1959.

8.5 SECTION 5 - SYSTEMS PROJECTING

A. Churchman, C. W., et al, Introduction to Operations Research, John Wiley, 1957.

B. Goode, H., Machol, S., Systems Engineering, McGraw-Hill Book Company, New York, 1957.

C. General Purpose Systems Simulator III Introduction, IBM, Technical Publications Department, White Plains, New York, 1965.

D. Knuth, D. E., and McNeley, J. L., "SOL - A Symbolic Language for General Purpose Systems Simulation" and "A Formal Definition of SOL", IEEE Transactions on Electronic Computers, Vol. EC-13, No. 4, August 1964.

E. Krasnow, H. S., Merikallio, R. A., "The Past, Present, and Future of General Simulation Languages", Management Science, Vol. 11, No. 2, November 1964.

F. Markowitz, H. M., Hausner, B., and Karr, H. W., SIMSCRIPT - A Simulation Programming Language, Prentice-Hall, 1963.

G. Morse, Philip M., Queues, Inventories, and Maintenance, John Wiley, 1958.

H. Roberts, E. B., The Dynamics of Research and Development, Harper and Ross, New York, 1964.

I. Current readings in the following periodicals:

Management Science
Operations Research Journal
Journal of Industrial Engineering

8.6 SECTION 6 - THE USER - MANAGER

A. Boulding, K. E., "General Systems Theory - The Skeleton of Science", Management Science, April 1956.

B. Box, George E. P., "Evolutionary Operation (EVOP): A Method for Increasing Productivity", Applied Statistics, Vol VI, No. 2, 1957.

C. Box, George E. P., and Hanter, J. S., "Condensed Calculations for Evolutionary Operation Programs", Technometrics, Vol. I, No. 1, February 1959.

D. Cerper, W. W., Leavitt, H. J., Shelly, M. W., II, New Perspectives in Organization Research, John Wiley, 1964.

E. Hitch, C. J., and McKean, R. N., The Economics of Defense in the Nuclear Age, Harvard University Press, Cambridge, Massachusetts, 1965.

F. Meister, D., Rabideare, G. F., Human Factors Evaluation in System Development, John Wiley, 1965.

G. Shewart, W. A., Statistical Method From the Viewpoint of Quality Control, Department of Agriculture, Washington, D. C., 1939.

H. Simon, H. A., Models of Man, John Wiley, 1959.

I. Simon, H. A., and March, J. G., Organizations, John Wiley, 1964.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Syracuse University Research Corp. Syracuse University Syracuse, New York	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE
Design Considerations for an On-Line Management System

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Final Report

5. AUTHOR(S) (Last name, first name, initial)
Allderige, J.M.
Knoke, P. J.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
July 1966	86	131

8a. CONTRACT OR GRANT NO. AF30(602)3505	9a. ORIGINATOR'S REPORT NUMBER(S) SURC TR#20
b. PROJECT NO. 4594	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	RADC TR-65-424

10. AVAILABILITY/LIMITATION NOTICES
This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of RADC (EMLI). GAFB,N.Y. 13440

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Rome Air Development Center (EMIRD) Griffiss AFB, N.Y. 13440

13. ABSTRACT

Under a contract to Rome Air Development Center (AF30(602)3505) warded in Sept 64, the Syracuse Univ Research Corp has undertaken the esign of a Managerial On-Line Data System (MOLDS).

This report, written from an operations research point of view, draws pon the numerous subtasks accomplished during the first year's effort nder this contract, and attempts to put the entire MOLDS design problem in proper perspective.

Additionally, certain design specifications and implementation rocedures are recommended. Discussions range from the detail level to he near-philosophic, but in many instances unnecessary detail (otherwise available to RADC in a series of interim reports) has been deliberately suppressed. Because the R and D management environment is the ontext for MOLDS design, specific attention is paid to that environment. owever, it is pointed out that system design concepts and computer usage echniques developed for MOLDS should in general prove applicable to ommand and control systems and intelligence systems.

The major emphasis throughout is on the need for specific operating xperiences to gain operational insights vital to effective implementation f MOLDS.

DD FORM 1 JAN 64 1473

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Management Computers Computer Programming						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through ________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through ________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through ________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.