R E P O R T   R E S U M E S

ED 010 558   56

RESEARCH AND TECHNOLOGY DIVISION REPORT FOR 1966.
BY- BAUM, C.
SYSTEM DEVELOPMENT CORP., SANTA MONICA, CALIF.
REPORT NUMBER NDEA-VIIA-1130-01   PUB DATE JAN 67
REPORT NUMBER BR-5-0738-01
REPORT NUMBER TM-530/010/00
EDRS PRICE MF-$0.36 HC-$9.36 234P.

DESCRIPTORS- *SYSTEMS APPROACH, *SYSTEMS ANALYSIS, *SYSTEMS DEVELOPMENT, *COMPUTER ORIENTED PROGRAMS, *PROGRAMING, INFORMATION PROCESSING RESEARCH PROJECTS, EDUCATIONAL RESEARCH, SANTA MONICA, CALIFORNIA

THE WORK OF THE RESEARCH AND TECHNOLOGY DIVISION OF SYSTEM DEVELOPMENT CORPORATION DURING 1966 IS REPORTED. THE PROGRESS OF VARIOUS STUDIES AND ACTIVITIES DISCUSSED IN THE REPORT WERE ADVANCED PROGRAMING, INFORMATION PROCESSING RESEARCH, PROGRAMING SYSTEMS, DATA BASE SYSTEMS. LANGUAGE PROCESSING AND RETRIEVAL, BEHAVIORAL GAMING AND SIMULATION RESEARCH EDUCATION AND TRAINING, MATHEMATICS AND OPERATIONS RESEARCH, COMPUTER CENTER DEPARTMENT, AND SPECIAL SERVICE OPERATIONS. IN ADDITION, THE REPORT CONTAINS DESCRIPTIONS OF DIVISION-SPONSORED BOOKS, DEMONSTRATION PROGRAMS, MEETINGS AND COLLOQUIUMS, AND PROFESSIONAL ACTIVITIES OF THE STAFF. (TC)

TM-530/010/00

TECHNICAL MEMORANDUM

(TM Series)

Research & Technology Division Report for 1966

by the

Research & Technology Division Staff

D. L. Drukey, Division Manager
B. O. Barancik, Assistant Division Manager
K. W. Yarnold, Director of Research
J. I. Schwartz, Director of Technology
G. H. Dobbs, Mgr., Computer Center Dept.

C. Baum, Editor

January 1967

SYSTEM DEVELOPMENT CORPORATION
2500 COLORADO AVE.
SANTA MONICA
CALIFORNIA
90406

ED010558

A-2348 4/65

ABSTRACT

This document describes the work of SDC's Research & Technology Division for 1966. The progress of the various studies and activities in the Division is described under the following major headings: Advanced Programming, Information Processing Research, Programming Systems, Data Base Systems, Language Processing & Retrieval, Behavioral Gaming and Simulation Research, Education & Training, Mathematics & Operations Research, Computer Center Department, and Special Service Operations. In addition, the back of the Report contains descriptions of Division-sponsored books, demonstrable programs, meetings and colloquia, and professional activities of the staff.

ACKNOWLEDGMENT

The work reported in this document was supported by the SDC Independent Research Program and by the following contracts and grants:

Agency	Contract or Grant Number
American Documentation Institute	65-1
Department of Commerce	
Bureau of Public Roads	CPR-11-4191
Department of Defense	
Advanced Research Projects Agency	SD-286 DA-49-083 OSA-3124 Order 773 (AF 19(628)-5166) F 1962867C0004
Department of Health, Education and Welfare	
U. S. Office of Education	7-14-9120-217 OE-6-85-076
Public Health Service	1 R01 LM00065-1
Fund for the Advancement of Education, Ford Foundation	
National Science Foundation	GN-408 GS-507 GY-371 NSF-C 424 GN-544
Southwest Regional Laboratory for Educational Research and Development	
State of California	
Department of Education	2591
U. S. Air Force	
Air Force Cambridge Research Laboratories, Office of Aerospace Research	F 1962867C0008 AF 19(628)-5166
Air Force Office of Scientific Research, Office of Aerospace Research	AFOSR-1203-67 AF 19(628)-5166
Electronic Systems Division, Air Force Systems Command	AF 19(628)-5166
Rome Air Development Center	AF 19(628)-5166
U. S. Navy	
Office of Naval Research	Nonr-4421(00) Nonr-4427(00) Nonr-4745(00)

CONTENTS

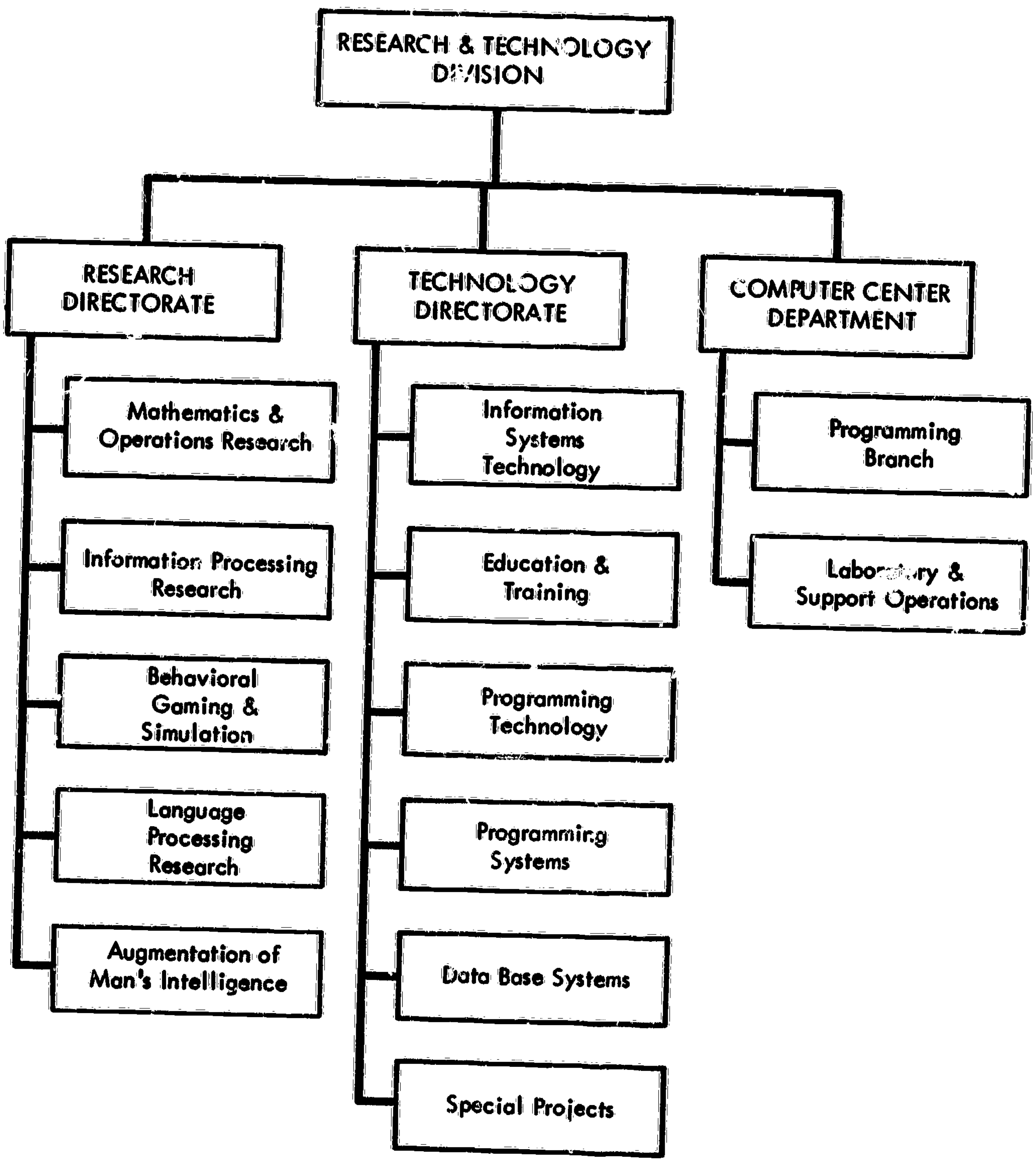


FIGURE 1. ORGANIZATION OF RESEARCH & TECHNOLOGY DIVISION

INTRODUCTION AND OVERVIEW

Mission

The mission of SDC's Research & Technology Division is to provide support for the rest of the Corporation and for its customers. The program consists of extensive research and technology development, particularly in the information sciences, and the provision of a new computer center for the Corporation. The research and technology tasks are intended to develop the tools which SDC's customers will use in carrying out their functions and which the professional SDC employees will use in the fulfillment of their contractual commitments to governmental and other clients. The computer center is being developed in the Research & Technology Division so that the latest technological developments will be available to all SDC professionals at an early stage.

Organization

The program is carried out in three organizational units--the Research Directorate, the Technology Directorate, and the Computer Center Department (see Figure 1). Because of the close relationships among these units, the lines of demarcation are not clear-cut among them. The Research Directorate is generally responsible for programs with long-range implications, for which there may not be an immediate match to existing requirements. The Technology Directorate's work spans the interval between these long-range programs and appliable technology; it is therefore concerned with applied research, pretechnology, and technology development, as well as a certain amount of fundamental research. For much of the Technology Directorate's work, the end objective is pretty well known at the time the project is undertaken and its relationship to SDC and our customers' operations is often fairly clear-cut (although the way of achieving those desired ends may not be quite so well determined). The Computer Center Department has been developing the software for a new computing complex, intended for a wide range of corporate users.

Technical Overview

The research and technology programs are composed of several main threads of activities which are mutually supporting. Typically, each thread includes projects ranging from fairly basic research through the development of technology.

The first of these threads relates to the handling of large files of structured information--the data base problem. In this area, we are concerned with developing a series of tools that SDC's professionals and customers can apply to the operational problems of data management. We want to make it possible for a nonprogrammer to describe, organize, and update his data, and then have the programming system carry out the mechanical functions of processing the data. In addition, we wish to make it simple for the user to ask for data in a manner that is convenient and natural to his way of operation and to have the system understand the details of data storage, organization, and conversion. Our interests in this area include the content as well as the format of the computer output.

The second main thread of our efforts concerns computer programming languages. Most past work in this area has consisted of developing languages for the professional computer programmer. These languages were aimed at relieving him of some of the tedium and bookkeeping necessary to produce computer programs. A variety of such languages has been produced, including the JOVIAL language developed by SDC and used extensively within the Corporation and externally. We have augmented this traditional approach to programming languages with initial efforts at providing languages for the nonprogrammer. The overall work spans the spectrum from basic theory of programming languages through the implementation of compilers for several languages of present interest. Encompassed in this domain are the development of new aids for the programmer and of techniques for managing the computer program development process.

The third thread of our activities concerns the executive systems within which the programs of a computing complex operate. In particular, an innovative technology of time-sharing, which makes it feasible for users to operate on-line in direct two-way conversation with the computer, has been developed through this work. Our efforts are aimed at exploiting the capability inherent in time-sharing systems and in understanding the basic scheduling and allocation processes associated with having a number of programs operating more or less simultaneously in the computer complex. In addition, much of the overall emphasis of our programs in technology and of the techniques that we use in research revolves around our unusual on-line capabilities; we are concerned with the advantages made feasible by having the computer directly accessible to the users and with exploiting these advantages for the various purposes toward which the whole program is aimed.

The fourth thread is concerned with computer processing of natural language, that is, English as it is spoken and written--as distinct from the formatted languages of data base systems or the formal computer languages. A sizeable program of basic research is being devoted to solving the problems of semantic and syntactic analysis of text by computer, toward the eventual goal of providing capabilities for using subsets of natural English as query or command languages for computers, and enabling computers to read, understand, and generate English text. At the more applied end of the spectrum, we are concerned with developing tools for automatic classification and indexing of documents and otherwise automating the storage and retrieval functions of libraries and document centers as well as of individual document holders.

The fifth thread of our activities relates to the processes of education. In this area we are concerned with new technologies for instruction, particularly the use of programmed materials; with the potential offered by the computer for assisting teachers, counselors, and administrators; and with the implications of these new technologies for school administration, including problems of flexible scheduling of resources and personnel. This program is continuing to broaden in scope and to place increasing emphasis on bringing new technologies into use in the public school systems.

Finally, we have programs in mathematics and operations research and in exploring the processes of human decision making, and of man-machine interaction, through behavioral gaming. Much of the mathematical work is of a research nature and relates in a general way to corporate interests in simulation, modeling, and system analysis. The work in behavioral gaming and simulation is leading to a better understanding of the nature of group decision making and the functioning of organizations, and is also resulting in a set of innovative tools for the conduct of behavioral games and computer-based data analysis. In the long run, we hope to see this work lead to improved comprehension of the interaction of people within their society and to the knowledge that makes it possible to improve that society.

It should be pointed out that these basic threads intertwine as we begin to derive appliable technologies from them. The operating systems provide the framework within which all of the computer programs operate. The data base systems are integral to the operating system and will, in our plans, provide a data management function for all system users. The operating and data base systems are written in the programming languages that we have been developing; and our language processors or compilers are embedded in the systems. Although there was a considerable gap between the natural language efforts and those related to the more formal languages, we are beginning to develop systems that combine mixed-mode retrieval capabilities of both structured and unstructured data. The mathematics and operations research personnel provide consulting assistance for many of the other research and technology projects. The work in education has been focusing on the potential of on-line man-machine interaction, made possible by time-sharing technology. The behavioral gaming area is producing novel techniques for human use of computer-based consoles and displays. Finally, the Computer Center Department will be providing--on an operational basis for a variety of users--the data base, time-sharing, and advanced compiler concepts developed by the research and technology program.

Progress

The past year was one of actual or impending transition for many of the 70-some R&TD projects that utilize a computer, as we moved toward replacing existing facilities with a third-generation computing installation. Specifically, the Philco 2000, which since 1961 had supported many SDC's Independent Research projects and the original Systems Simulation Research Laboratory, was sold in September 1966. The IBM Q-32 computer, which has been supporting the balance of the computer-based projects in R&TD on the SDC-pioneered Time-Sh 'ng System, is scheduled to phase out late in '7. Replacing these computers, as well as some others at SDC, is a series of IBM S/360 machines, for which R&TD's Computer Center Department has been developing the complete software. A 360 Model 50 was operational from October 1965 through July 1966; it is being replaced by a time-shared Model 65 which began to achieve a useful capability during the latter part of the year.

The changeover in computers is resulting in mixed consequences. On the one hand, certain experiments have been delayed due to reprogramming, a hiatus in machine availability, and the uncertainties that inevitably accompany the installation of new equipment. On the other hand, many of the projects are taking this opportunity to make long-sought program changes and design improvements afforded by this breathing space and by the more powerful capabilities of the new installation.

Turning to specific projects, the data base area was marked by increased usage of our prototype data management tools. TSS-LUCID, an on-line data management system, was used by about 50 people each month during 1966, including many external users, for a wide range of data management problems. The General-Purpose Display System (GPDS) was successfully harnessed to a particular application, namely the calculation and display of salary maturity curves. Various parts of the new Time-Shared Data Management System--an integrated set of data handling tools based on experience with LUCID, GPDS, and other systems--reached promising stages of design, coding, and checkout. TDMS is scheduled to be a major resource on SDC's 360 Time-Sharing System.

In the realm of programming languages, the techniques of metacompilation received increased emphasis as a means of reducing the cost and time to produce compilers. A compiler system, called META, and an interpretive system, called META5, have been developed and refined; they have found many useful applications including the production of compilers, translation between programming languages, data base conversions, and program reformatting. A version of an advanced list processing language, LISP 1.5, was developed during the year, and work was initiated on LISP 2, which will provide improved capabilities for manipulating complex data structures and performing lengthy arithmetic computations. Design and coding were well under way for an Interactive Programming Support System, intended to give the professional programmer the maximum benefit of the on-line capability of time-sharing. Programs were developed to facilitate automatic recognition of handwritten characters, automatic flow charting, and automatic code improvement. Finally, a handbook to aid managers in preparing estimates of computer program development projects was published.

The executive system area underwent profound changes, with final improvements made to the Time-Sharing System (TSS) on the Q-32, and the development of a new system on the 360. One of the first general-purpose time-sharing systems, TSS became operational in June 1963 and has been continually improved and refined over the years, to the point where its sponsor, the Advanced Research Projects Agency, felt that it had well fulfilled its purpose as a developmental research project. During 1966, the system was made available to interested users on a subscription basis; ARPA support is no longer required and has been redirected to more developmental efforts. The more than 500 users who previously had free access to the system are now limited to those who use it on a paid basis. During the year, a comparison of on-line versus off-line programmer performance resulted in qualified advantages for time-sharing. Additionally, a small step was taken toward the next important area in executive systems--the connecting of several computers into a network--by the linking of the Q-32 computer in Santa Monica to the TX-2 computer at Lincoln Laboratory in Boston.

In the natural language area, several new studies were initiated to complement the existing work on automated language processing. The new studies, sponsored by the Advanced Research Projects Agency, include a computer-based semantic analysis of sense relationships of the words in a dictionary, the development of an on-line transformational grammar tester, and a study of the use of English subsets in query systems. Additionally, Protosynthex III, a fairly complete approach to a natural language processor, emerged from the preliminary design stage. In the library application area, BOLD, a highly automated display-oriented document storage and retrieval system, and SURF, a personal file retrieval system, were further refined and received initial usage. A new method of automatic document classification, called ALCAPP, broke through previously restrictive barriers of cost and storage space. A paper on ALCAPP, as well as one on an SDC study to evaluate document representations, were two of the three prize-winners at the annual meeting of the American Documentation Institute.

The education and training area completed several studies in 1966, including the development of a computer-based simulation of an innovative school; the development of 28 criterion tests that indicate absolute levels of mastery in foreign-language comprehension, speaking, reading, and writing; and a comparison of linear vs. branching strategies in presenting programmed material. Two major new studies were begun during the year: the development of a computer-based educational system for the Southwest Regional Laboratory for Educational Research and Development, and a study to adapt the SDC-designed "empirical trial-and-revision" process to the development of instructional materials and procedures for classrooms serving predominantly Spanish-American students. Considerable progress was made in ongoing investigations into the use of time-sharing in education: in one case to improve the counseling function; in another to improve the teaching of statistical inference at the college level. A very promising outcome of the last-named study is an on-line lesson design and teaching program called PLANIT, which is receiving wide interest and initial usage.

The mathematics and operations research staff continued to solve a number of challenging problems. With partial funding from the Bureau of Public Roads, SDC's Vehicular Traffic Study completed the coding, debugging, and exercising of an initial version of a computer simulation model of a freeway diamond interchange. A new project was begun on ways to compress the enormous amounts of data transmitted from spaceborne hardware. In celestial mechanics, new procedures for existence proofs were derived and applied, and a very efficient numerical technique was devised for solving certain differential equations. Additional work yielded new results in such areas as optimal strategies for item presentation in education, stochastic duels, life-testing, validation of simulation models, factor analysis, and mathematical programming. The algorithmic languages project continued to contribute important insights into formal languages through the noteworthy publication of 12 papers in major journals during 1966.

In the area of man-machine interaction, work continued along a broad front. A robot-like system, capable of following simple commands, was programmed and several films were made of its operation on a display scope. The augmentation of human intellect by machine was furthered by the initiation of an "augmented statistician" project, the development of an interactive problem-solving task called Shimoku, the completion of a set of experiments to test the effectiveness of various display aids to human problem solving, and the development of various on-line data manipulation systems. The continuing research on bargaining and negotiation behavior completed several major experiments during the year, including a transnational study which culminated in a conference of participating researchers from various nations in Santa Monica in November. The Leviathan system, a computer-based model of large social organizations, was used experimentally as a training tool in a management workshop conducted at the University of Southern California for Air Force officers.

As mentioned before, the Computer Center Department completed design and programming for a number of operating systems on a series of IBM 360s: a batch-processing Model 50, which was operational from October 1965 through July 1966; an interim batch-processing system on the Model 65, operational since July; and a time-sharing system on the Model 65, for which an initial version was released in November. In addition, the Department has been developing other supporting aids, including a 360 JOVIAL compiler, on-line and off-line debugging tools, program and text editing capabilities, file maintenance programs, and assembly language processors. The laboratory and planning staffs have been active in the selection and acquisition of a complex of displays and other ancillary equipment to be used in connection with the 360.

Organizational Changes and Appointments

In recognition of his technical and administrative accomplishments as assistant to the R&TD Manager, Bill Barancik was appointed Assistant Division Manager in January 1967.

During 1966, several organizational changes took place in the management and composition of the various areas of the Research and Technology Directorates. In March, Gerald Shure was named head of the Decision Processes Research staff. In August, the Research Directorate underwent a restructuring: the concept of "staff" areas was changed to one of "program" areas; Decision Processes Research was replaced by Behavioral Gaming & Simulation (with Shure as head); and a new program, Augmentation of Man's Intellect, was formed with Research Director Kenneth Yarnold as acting head. In December, a logical bifurcation of the Language Processing and Retrieval staff resulted in a research-oriented Language Processing

Research Program, under Robert Simmons, in the Research Directorate, and an applications-oriented Information Systems Technology staff, under Carlos Cuadra, in the Technology Directorate.

In the Computer Center Department, Al Irvine, formerly head of the UCLA Computer Network project's programming staff, joined SDC to head the CCD's Programming Branch, and Jerry Hanna was appointed Assistant to the CCD Manager, responsible for the nonprogramming aspects of the Department's activities.

Other major appointments during the year included: Andy Gafarian (Mathematics & Operations Research Program) and Gerald Shure to Senior Scientists; and Robert Bleier and Sally Bowman (both of the Data Base Systems staff) to Research Leaders.

Interaction with Other Divisions

During 1966, the Research & Technology Division increased its efforts to communicate with and support other corporate organizations. A survey undertaken in June of 1966 indicated that, during that month alone, approximately 60 R&TD personnel, or about 30 percent of the professional staff, were actively engaged in 23 clearly identifiable, special-purpose, ad hoc activities in support of other SDC organizations. These activities ranged from briefings on R&TD projects to long-term consultation on projects of other divisions. It should be stressed that this survey did not cover the normally ongoing corporate-wide responsibilities of R&TD, which include the development of the Computer Center, maintenance of information centers on information processing and information science, furnishing of statistical services, and, more generally, the overall development of the products of research and technology, which are intended for the use of the other SDC organizations and their customers.

An important advance in corporate communication was the formation of five Interdivisional Technical Steering Committees by SDC President Melahn in September 1966. The committees, chaired by senior corporate managers and composed of high-level technical representatives from the various SDC organizations, are intended to provide a mechanism for the communication of technical information among SDC's divisions and to help solve common problems in the areas of data base systems, education and training technology, executive systems, natural language processing, and programming languages. In addition to active participation on the Committees, R&TD has supported them through initial conception and planning, and provision of recording personnel. During the last four months of 1966 the five Committees met a total of 21 times, and can point to a number of accomplishments, such as the undertaking of new interdivisional projects, improved utilization of common resources, and measurably enhanced communications.

Another innovation in 1966 was the R&TD internship program. Under this program, individuals selected by the other divisions join R&TD for periods of up to six months, to participate actively in research and technology projects. The objective is to provide the participants with a working knowledge of R&TD's technology, which they will ultimately carry back to their own organizations. Eight interns joined R&TD in the last half of 1966. Judging from initial indications, the program is accomplishing its aims and will be continued in 1967.

Also during the year, R&TD provided two 2-day sessions of special briefings and demonstrations expressly for personnel in the other divisions. Approximately 150 middle management and senior technical personnel from throughout SDC received detailed descriptions of the ten most frequently demonstrated R&TD products.

Communications

As in the past, R&TD continued an active program of communication, both internally and externally. Division personnel were coordinators and hosts for a number of local, national and international meetings. Also, 20 research colloquia, featuring both SDC and external speakers, were held during the year; these are open to all SDC personnel and invited outsiders.

One of the most effective ways in which we have communicated to the outside world is through live demonstrations of the programs developed in R&TD. During 1966, several thousand visitors to SDC received first-hand exposure to man-machine interaction under time-sharing, as the computer displayed to them, on scopes or teletypes, the results of the actions taken by the demonstrators or, in many cases, by the visitors themselves. Additionally we gave or supported a large number of remote demonstrations, linked via teletype to the Q-32 computer in Santa Monica, at symposia throughout the country and abroad. During the year considerable effort was devoted to improving our demonstrations, from the standpoints of both presentation and technical support. A number of the more frequently demonstrated programs are briefly described in the Appendix.

During 1966, approximately 40 consultants and selected graduate students participated in the Division's program, over periods ranging from several days to several months. Apart from the technical contributions made by these people, the exchange of ideas gained by these close working relationships has been of great value to SDC and to the visiting personnel.

Another effective medium of communication is the "lend-lease" program, instituted in 1966. Similar to the internship program described above, the lend-lease program enables technical specialists from outside agencies to work on SDC projects, providing a valuable exchange of ideas between these people and R&TD researchers. During 1966, personnel from Shell Oil, Atlantic-Richfield, IBM, and Bolt, Beranek and Newman joined R&TD to participate in the development of projects in data management and programming languages.

As is traditional in science and technology, SDC makes a great effort to communicate its findings to the external community. During 1966, Division personnel published 60 articles in the external literature; this was augmented by over 300 SDC documents, most of which are available through the Defense Documentation Center.

Finally, R&TD personnel gave a total of 175 oral presentations for professional meetings, university colloquia, and invited lectures. In addition, many members of the Division are officers in their professional societies and editors for journals in their field.

Detailed information on all these activities can be found at the back of this report.

Postdoctoral Fellowship Program

In the fall of 1966, R&TD instituted a Postdoctoral Resident Research Fellowship Program. Fellows selected under this program will receive a $9,000 stipend while conducting research of their own choosing in the Research & Technology Division in Santa Monica. Major resources available to Fellows are the knowledge and experience of a multidisciplinary staff of senior investigators, and the facilities of a computer-based man-machine laboratory.

A representative, but by no means exhaustive, list of areas that may be proposed for research includes man-machine interaction, operations research, mathematical modeling, digital simulation, education and training, experimental gaming, decision making, computational linguistics, information management, computer graphics, automata theory, formal and programming languages, programming systems, and the application of information processing to law, medicine, economics, and other fields.

The fellowship program was developed partly as a result of the successful experiences of NSF and NIH fellows who have spent their research periods at SDC.

Research Advisory Committee

The Research Advisory Committee continued to play an important role in the Corporation. During 1966, the Committee met for four 2-day sessions at Santa Monica (including a joint meeting with the Board of Trustees) to consider the work of R&TD and to offer broad guidance on the overall program. During the year the RAC focused on the following areas: R&TD's operating plan, the "augmentation of man's intellect" program, education and training, and executive systems.

In July, President Melahn announced the rotation of the chairmanship of the RAC to Dr. Merrill Flood, succeeding Dr. C. West Churchman who had chaired the RAC since its inception in 1962. Dr. Churchman continues to serve as a valued member of the Committee. The complete list of RAC members is as follows:

Dr. Merrill M. Flood (Chairman)
Professor and Senior Research Mathematician
University of Michigan

Dr. William C. Biel (Secretary)
Associate Dean of the Graduate School
University of Southern California

Dr. C. West Churchman
Professor of Business Administration
University of California, Berkeley

Dr. Harry D. Huskey
Professor of Mathematics and Electrical Engineering
University of California, Berkeley

Dr. John L. Kennedy
Department of Psychology
Princeton University

Dr. Anthony G. Oettinger
Professor of Mathematics and Linguistics
Harvard University

General Earle E. Partridge
USAF, Retired

Organization of Report

In the main, the report that follows has been organized to reflect the several threads of attack indicated in the technical overview. In cases where a project belongs administratively under one area, but fits functionally more clearly in another, the functional relationships have governed. Thus, the administrative structure of R&TD, and in particular the organizational changes that occurred late in the year, are not necessarily reflected in the project descriptions that follow.

If it becomes increasingly difficult for us to pigeonhole a given project--for example, to judge whether the PLANIT language for on-line lesson design belongs under education or man-machine interaction or programming languages or some other area--we take these multiple attachments to be a healthy sign. Our aim is to continue to break down the traditional barriers imposed by different disciplines and skill fields, with the goal of producing tools that draw upon many areas of knowledge to fulfill a wide range of uses.

Donald L. Drukey
Vice President and Manager
Research & Technology Division

ADVANCED PROGRAMMING

E. H. Jacobs, Head

The Research & Technology Division's extensive work in programming technology is being conducted by several staffs. For purposes of this report, these efforts have been integrated in a single section titled "Advanced Programming." Constituting the major part of this section is the work of the Programming Technology staff, headed by E. H. Jacobs. Also included are appropriate parts of the work of the Programming Systems staff (see also page 3-1), the Information Processing Research staff (see also page 2-1), and the LISP development activity.

The area of Advanced Programming embraces study and development of tools and techniques for the computer programmer, the nonprogramming user of computers, and the manager of computing installations. The major areas being explored are compilers and programming languages, aids to programmers, and studies of the programming process.

The first of these, compilers and programming languages, has been a major activity at SDC for some years and the JOVIAL language and compilers have emerged as usable tools. Continuing research aims at finding new techniques of compiler production and at increasing the language capability available to programmers.

In the realm of improved compilers, the techniques of metacompilation are being studied, as a possible means of reducing the time and cost required to produce compilers. These techniques have shown the capability to produce certain parts of compilers, notably the so-called "front-end" which translates from source language to an intermediate language and, in some cases, to machine language. Both a metacompiler and a metalanguage interpreter have been constructed and are being used to explore the problems of "describing" a compiler. The metalanguage interpreter has also been used for other applications, including the translation of one POL into another.

In programming languages, one current project is aimed at extending LISP, an advanced list processing language. List processing languages have been found useful in work involving extensive manipulation of symbols (as opposed to arithmetic computation), but their utility has been blunted by the slow speed of their processors and by their limited capability to handle problems involving both symbolic manipulation and arithmetic computation. The LISP project is producing an advanced processor, with a built-in computational facility. One version of such a processor, called LISP 1.5, was built during the year and used to test ideas. Another version, LISP 2, is being built to implement these ideas in a complete system.

Exploration in another dimension, furnishing new aids to programming, is being vigorously pursued. The work includes finding aids that help the programmer to make better use of the higher level languages, making it easier to get programs written and assisting in code checking.

A major project in this area is the design and development of an interactive programming support system. Such a system will be designed to give the programmer the maximum benefit of the on-line capability available in a time-sharing mode of operation. Compilers and checkout tools specifically designed for a time-sharing environment will have many more points of programmer interaction than are normally found. Further, the language or control structure will be shaped to allow the programmer to switch from function to function (for example, from a compile mode to a debug mode) without having to make a corresponding switch in language.

Several other projects are expected to contribute to the interactive system. The work on compiler construction described earlier will provide the basis for the compiler used in the system. Other contributing projects are Automated Flow Analysis and Graphic Input/Output. The first of these is developing programs to analyze other programs in order to get a machine-produced flow chart. During the year, a program was written that produces a series of flow charts from a program written in JOVIAL. The first chart is very detailed and successive ones present less detailed information, giving a better overall picture. A byproduct of this work was the development of a program to automatically improve code written in JOVIAL.

The Graphic Input/Output project is developing techniques by which a programmer may write his code directly into the computer (as opposed to use of a teletypewriter or keypunch). Initial work is on character recognition routines that can operate rapidly enough to be useful in the on-line environment. At the same time, routines that display lines of code and make deletions and insertions are being developed. Several experimental recognition routines as well as parts of the needed control programs have been written and checked out on the Philco 2000.

Study of the program development process has also been a significant activity. This research is aimed at systematizing and improving control and planning techniques for use by managers of computer program development. The work includes analysis of the process of program development to identify relationships among programming products, resources, and environment. The goals are to identify and develop economical and efficient management methods for realizing programming products and to establish criteria for measuring the quality of these.

The central effort in this area has been the statistical analysis of numerical data, characterizing completed computer programs, to derive improved methods for estimating costs. The work has resulted in publication of a handbook for use by managers in preparing estimates of a computer program development project.

COMPILER CONSTRUCTION TECHNIQUES

Syntax-Directed Compilers and Translators*

E. Book
J. Igawa
M. Schaefer
D. V. Schorre

Description

The objective of this study is to develop techniques that will simplify the task of producing a compiler f-r a new computer. SDC's experience has shown .hat the 9 to 12 months currently required to produce a compiler for command-control systems is a serious delay in the production of operational programs. The use of metacompilers has received much study by the computer community and is the focus of this project. This technique offers not only a possibility for constructing compilers for new machines, but also a way to build compilers for new languages.

The metacompiler makes use of a rigorous formal description of the language for which a compiler is to be produced. In addition, it requires a description of the translation (or compilation) process. (This description includes, either explicitly or implicitly, a description of the target machine language.)

The work has taken two major forms, called META and META5.

META is a compiler system which accepts as input a description of a desired compiler, in a specialized form, and which outputs the desired compiler in an executable form.

META5 is an interpretive system which consists of the META5 language, a META5 compiler, and a pseudomachine which is implemented on the Q-32 computer. The META5 language allows a variety of data structures to be declared and used in the language. It also contains some operations not yet available in a system like META, but useful in data manipulation, e.g., relational and arithmetic operators; an assignment operator; search, concatenate, enter operators; etc.

The META language consists of two sublanguages. The first of these, called SYNTAX, is used to describe the syntax of a desired source language, specifying a mapping of a program in the source language into a tree-structure representation. The second sublanguage, called GENERATORS, specifies the correspondence of the tree structure to a desired target language. This is the semantics of the source language. It is the link between the string of marks input to the computer (the source language program) and the actions to be performed by the computer to achieve the desired result. This language is a cross between a pattern matching notation and a macro notation. It is a new language and will be developed further. The two sublanguages are nonprocedural or descriptive languages which are specially designed for the specification of compilers and/or interpreters.

The subroutines that do the work specified by the SYNTAX and GENERATOR languages are written in a language called MOL (Machine Oriented Language). These routines could have been written in any procedural programming language such as JOVIAL, ALGOL, MAD, etc., or even in assembly language. However, the MOL language was designed and a compiler for it implemented for the reasons specified below.

First, MOL is based on a compromise. On the one hand, it is desirable to take advantage of the computer hardware in producing machine code for various features of the language. On the other hand, it is undesirable to descend to the level of actual machine operations. MOL has an ALGOL-like flavor, but the operands and some operators are concerned with machine registers. Such things as indirect addressing, partial word fields, user control of index registers, and similar considerations are specifiable in this

*Supported in part by the Advanced Research Projects Agency.

language. MOL combines an assembler's vocabulary with a compiler's grammar.

Second, it was felt that more control could be exercised in adding features that would mesh this language into the META system if it were specifically designed and implemented for that purpose.

Third, since MOL is a procedural language, it provides an early test of the ability of the system to describe a conventional compiler.

Progress

The SYNTAX and GENERATOR languages were designed and a description of the syntax of both languages was written in SYNTAX. This also showed the mapping into the internal model (tree structure). The mapping of the internal model to Q-32 code was described for both SYNTAX and GENERATOR in the GENERATOR language.

The procedural language MOL-Q32 was implemented in an earlier version of META which did not have the generator language. The subroutines required to implement SYNTAX and GENERATOR were written in MOL-Q32. This resulted in the entire META system being described in its own language and able to compile itself. This process was actually performed by bootstrapping from a more primitive version of the system.

A MOL for SDC's S/360 System was designed and a compiler for it was described in META. The GENERATOR portion of META was changed to produce IBM 360 code, instead of Q-32 code, from the internal model. The resulting version of the META system for the 360 is currently being checked out. The bootstrap process to move from one computer to another is very well defined using this method.

The META5 system has been useful in describing POL-to-POL translations (p. 1-5), data base conversions, some analysis of a subset of English as used in questions, etc. Specifically, JTS-to-JS, TINT-to-JTS, J3-to-JS, FORTRAN-to-JTS, and JS-to-PL/I conversions have been accomplished. Various data bases have been converted for the LUCID system (p. 4-7). The META5 system has also been used to reformat META5 programs and to write the META5 compiler. A calculation program and an input processor for JOVIAL constants have been written in META5. The last two uses were of particular interest since they were written for on-line interaction between the user and the computer.

As the META5 system became a working tool for POL-to-POL translation, the META5 language was expanded to facilitate string and character manipulation. The META5 system is currently being moved to the S/360 computer and, since the META5 language is totally machine independent, programs written for Q-32 META5 will run with no modification on the 360. The J3-to-JS translator was used to bootstrap the META5 system over to the JS/360 dialect of JOVIAL.

Plans

While the META5 system is being moved to the S/360, improvements will continue to the language and system as needed. Tree-building and manipulation capabilities may be added along with multiple input/output capabilities, to permit the production of compilers. Debugging capabilities will be made available. Study will be made of the feasibility of partial recompilation of META5 programs. More POL-to-POL translators will be written including a PL/I-to-JS translator.

The META system is also being moved to the S/360. When this is completed, improvements will be made. One planned improvement is the addition of a feature to describe and handle a dictionary containing information about data types. The GENERATOR language will continue to be developed. After this, work will split into two directions: (1) further investigation of metalanguages and processors, and (2) the use of META to produce compilers and interpreters for various nonprocedural languages for experimentation.

Project Documentation

1. Oppenheim, D. K. The META5 language and system. SDC document TM-2396. July 21, 1965. 49 pp.
2. Book, E. and Schorre, D. V. A higher-level machine-oriented language as an alternative to assembly language. SDC document TM-3086/001/00. August 12, 1966. 29 pp.
3. Oppenheim, D. K. and Haggerty, D. P. META5: A tool to manipulate strings of data. Proceedings of the 21st National ACM Conference. August 1966, 465-468. (Available as SDC document SP-2243/000/01.)

Translation Between Procedure-Oriented Languages

D. P. Haggerty

Description

A major difficulty hindering the introduction of improved procedure-oriented languages (POLs) into an established computing facility is the incompatibility between the new or improved languages and older languages long in use. It is often desirable to modify old programs by using the new language or to combine routines written in the old language with routines written in the new one. One method of minimizing this difficulty is the design of translation programs between old and new POLs.

This project is exploring the possibilities and limitations of automatic translation of one POL into another.

Several programs designed to accomplish a translation from one POL to another existed at the inception of this project--an SDC program to translate FORTRAN II to JOVIAL (J2) and an IBM FORTRAN II-to-FORTRAN IV translator. Two others were being programmed--a TINT-to-JTS translator and a JTS-to-JS translator.

The first two translators were produced using "traditional" methods, i.e., they were written in the conventional programming languages J2 and FORTRAN, respectively; in contrast, the latter two translators were written in META5, a syntax-directed compiler-writing system (see page 1-3).

The languages selected for translation in this particular study were JOVIAL (JS version) and PL/I (i.e., a JS-to-PL/I translator) and the META5 system was selected to realize the translator.

The language pair was chosen because both would be available on IBM System/360 and a POL-to-POL translator would have practical value The META5 technique was selected because a TINT-to-JTS translation was partially implemented at the time and seemed to indicate that the method could be applied to the more complex JS-to-PL/I translation.

Progress

At present, the translator is operating on the Q-32 under the Time-Sharing System.

The translator was written in the META5 language and debugged on-line under the Time-Sharing System. The use of the META5 language facilitated the writing of the translator in several ways; the specification of the syntactic recognition process for the various JS forms is not only precise and transparently explicit but also succinct. The transformational processes necessary to produce the PL/I equivalents are also expressed quite clearly and briefly.

The language specifications as a basis for writing the translator were, for JS: SDC TM-1682/003/00, JOVIAL (J-6) Grammar and Lexicon; and for PL/I: IBM, File No. S360-29, FORM C28-6571-3, IBM System/360 Operating System PL/I Language Specifications.

Although it is assumed that the JS input text has been found syntactically correct by the 'generator' phase of a JS compiler, the philosophy adopted with respect to the translator has been to produce as complete a translation as possible no matter how incorrect or garbled the input text may be; hence non-JS program segments will produce output, e.g., J3 programs will translate although usually incompletely.

The translator has converted a JS program that was designed to test the ability of a JOVIAL compiler to compile a JS program. The PL/I equivalent of this program has been submitted to IBM for syntax-checking by that part of the PL/I compiler; the program consisted of about 315 statements and 120 data declarations. A much larger J3 program (about 1800 statements and 250 data declarations) has also been translated; this translation should give an indication of how useful the translator is for translating J3 programs (for which it is not designed), i.e., how difficult the postediting job is.

Several PL/I translations have been compiled by the F-level compiler and have revealed errors in the translator.

The project has noted several aspects of programming languages that add to the difficulty of translation (these remarks also apply to the compilation process): (1) any ad hoc devices that a particular compiler uses to "fit" a language to a machine constrict the translation process; (2) the existence of structural connectivity (see E. T. Irons, "Structural Connections in Formal Languages," Communications of the ACM, Vol. 7, No. 2, pp. 67-72, for an exposition of this concept) in a language adds a considerable burden to a translator, and hence should not be introduced unnecessarily.

Plans

The META5 system is being converted to the IBM System 360. When this is completed, study of translation techniques will continue through the application of META5 to other language pairs. In particular, a PL/I-to-JOVIAL translator offers a promising research avenue.

Project Documentation

1. Haggerty, D. P. Use of the JS to PL/I translator. SDC document TM-2823. January 19, 1966. 4 pp.
2. Haggerty, D. P. JSPL: An application of the META5 system. SDC document TM-3003. June 14, 1966. 18 pp.

Compiler Techniques for Paging*

R. J. Dinsmore

Description

The goal of this project is to establish the compiler techniques for producing computer code that takes advantage of "page" and "segmentation" features of new computers such as the IBM 360/67 and the GE 645.

When the current generation of compilers was designed, computer memories generally consisted of a single block of locations whose addresses were simply a continuous sequence from zero to the highest available. The new computers, however, subdivide memory into smal' r blocks called "pages" and "segments," and it appears that special compiling techniques are needed to properly utilize these features. That is, compilers must produce computer programs that will take advantage of the small blocks of memory and minimize the number of times a process is interrupted for the loading of new pages or segments.

Other SDC projects (Automatic Code Improvement and Automatic Flow Charting) have shown that the generator phase of a compiler can obtain a great deal of information about the structure of a program, and it is anticipated that this is the kind of information useful in the paging problem.

Progress

The work plan on this project consists of a study of an existing compiler, the construction of a compiler that includes the new techniques, and a study of gains realized.

This project was instituted 'ate in the year with a study of the JOVIAL compiler for the IBM 360 and some of its output code. A number of possibilities for structuring a program have been hypothesized and are being considered for implementation in a compiler. Also the

*Supported by the Advanced Research Projects Agency.

differences in design between existing compilers and a paging compiler are being investigated.

In addition to the program structure itself, two additional techniques are being considered. One of these is to have the program utilize statistics on its own past behavior in loops and at branch points. The other technique is to have the program furnish notice of its future page needs to the executive.

Plans

Work will continue to find more potentially profitable ways in which programs can be structured. An experimental compiler will be built to test each of these techniques.

PL/I for SDC 360 Time-Sharing System

W. E. Meyer

Description

The purpose of this project is to investigate the difficulties involved in moving a compiler from one third-generation operating system to another. Specifically, the project is investigating the transfer of the PL/I F-level compiler from the IBM OS/360, a system designed for multiprogramming, to the SDC 360 Time-Sharing System.

The dependence of compilers and assemblers on operating systems has been increasing over the last ten years. This increasing dependence implies greater difficulty in transferring compilers from one system to another. With the advent of multiprogramming systems, this dependency is increased substantially because of the desirability of dynamic relocation, standard data structures, and flexible and dynamic linking of program segments.

Because of the widespread interest in PL/I and because of its potential usefulness in the SDC Time-Sharing System, PL/I was chosen as the object of this study. It was written to operate with the IBM OS/360; the SDC 360 System is sufficiently different that the transfer requirements are significant.

Progress

A list of probable problem areas was developed. They are:

1. The compiler interface with the operating system.

2. The compiled program's interface with the operating system.

3. The form of the compiled program as output by the compiler.

4. The form of data aggregates (e.g., files, records, data sets, control blocks, etc.) recognized by the operating system and the compiler.

5. The differences in input/output procedures; this is related to 4, above.

6. Dynamic (execution time) calls on the library. (This may be unique with OS/360.)

7. Possible side effects introduced in moving support programs from one operating system to another.

In the latter part of the year, Program Logic Manuals for the PL/I compiler and the PL/I library were received from IBM, and a study of the compiler and the library was begun. All compiler interfacing with OS/360 is contained in six control programs. All references to OS/360 in these routines have been identified. There are 62 calls, generated by 19 different system macros. This list of references will be used as a basis for the evaluation of the problem area 1, above.

With respect to the object program's interface, the PL/I compiler does not generate code that is operating-system dependent; all operating system interfacing is isolated in the PL/I library. There are 17 library routines using 20 system macros that supply the object code interface with the operating system. These macros are, for the most part, the same as those used in the compiler-OS/360 interface.

The form of the output program is not a problem in the case of PL/I since the "object module" that is output must be processed by the linkage editor, which outputs a program in a form readable by the loader. The need for link-editing arises because all memory allocation, data management, data conversion, and input/output processes have been placed in the PL/I library and the library routines must be linked to the object program by the linkage editor.

Plans

The study of the problem areas will be continued. The variances between the OS/360 and the SDC 360 System data aggregates will be tabulated and the differences between the two system interfaces will be evaluated.

PROGRAMMING LANGUAGES

LISP 2*

SDC: J. A. Barnett, Project Leader	III: L. Hawkinson
D. C. Firth	M. I. Levin
R. E. Long	P. W. Abrahams
E. Book	D. Crandal
R. E. Martin	R. A. Saunders
C. Weissman	E. C. Berkeley
S. L. Kameny	

Description

The LISP 2 Project is a joint development of SDC and Information International, Inc.

LISP 2, which is based on LISP 1.5, is a new programming language for manipulating complex data structures and performing lengthy arithmetic calculations. As in LISP 1.5, programs can be treated as data, and storage can be regained through a compacting technique known as "garbage collection." The LISP 2 Source Language (SL), which resembles ALGOL, is the standard input; the LISP 2 Intermediate Language (IL), which resembles LISP 1.5, is used for programs that are to be treated as data. Type declarations are available for efficient compilation of arithmetic operations. LISP 2 includes bit operators and an open subroutine capability. The most general form of a datum is a symbolic expression; other forms include numbers, functions, strings, and integer-indexed arrays. All of the system programs are themselves written in LISP 2. The I/O package transforms input into a stream of characters which are converted into tokens by the finite-state machine. The supervisor controls the various LISP 2 operations. SL is translated into IL by the syntax translator; IL is translated into assembly language by the compiler; and assembly language is translated into machine language by the LISP 2 assembler, LAP. Machine mobility is achieved through core image generation. (See Figure 1-1.)

Presently implemented on the Q-32 computer, LISP 2 has two components: the language itself, and the programming system in which it is embedded. The system programs that define the language are accessible to and modifiable by the user; thus the user has an unparalleled ability to shape the language to suit his own needs and to utilize parts of the system as building blocks in constructing his own programs.

While it provides these capabilities to the do-it-yourself programmer, LISP 2 also provides the complete and convenient programming facilities of a ready-made system. Typical application areas for LISP 2 include heuristic programming, algebraic manipulation, linguistic analysis and machine translation of natural and artificial languages, analysis of particle reactions in high-energy physics, artificial intelligence, pattern recognition, mathematical logic and automata theory, automatic theorem proving, game playing, information retrieval, numerical computation, and exploration of new programming technology.

The LISP 2 programming system provides not only a compiler, but also a large collection of run-time facilities. These facilities include the library functions, a monitor for control and on-line interaction, automatic storage management,

*Supported by the Advanced Research Projects Agency.

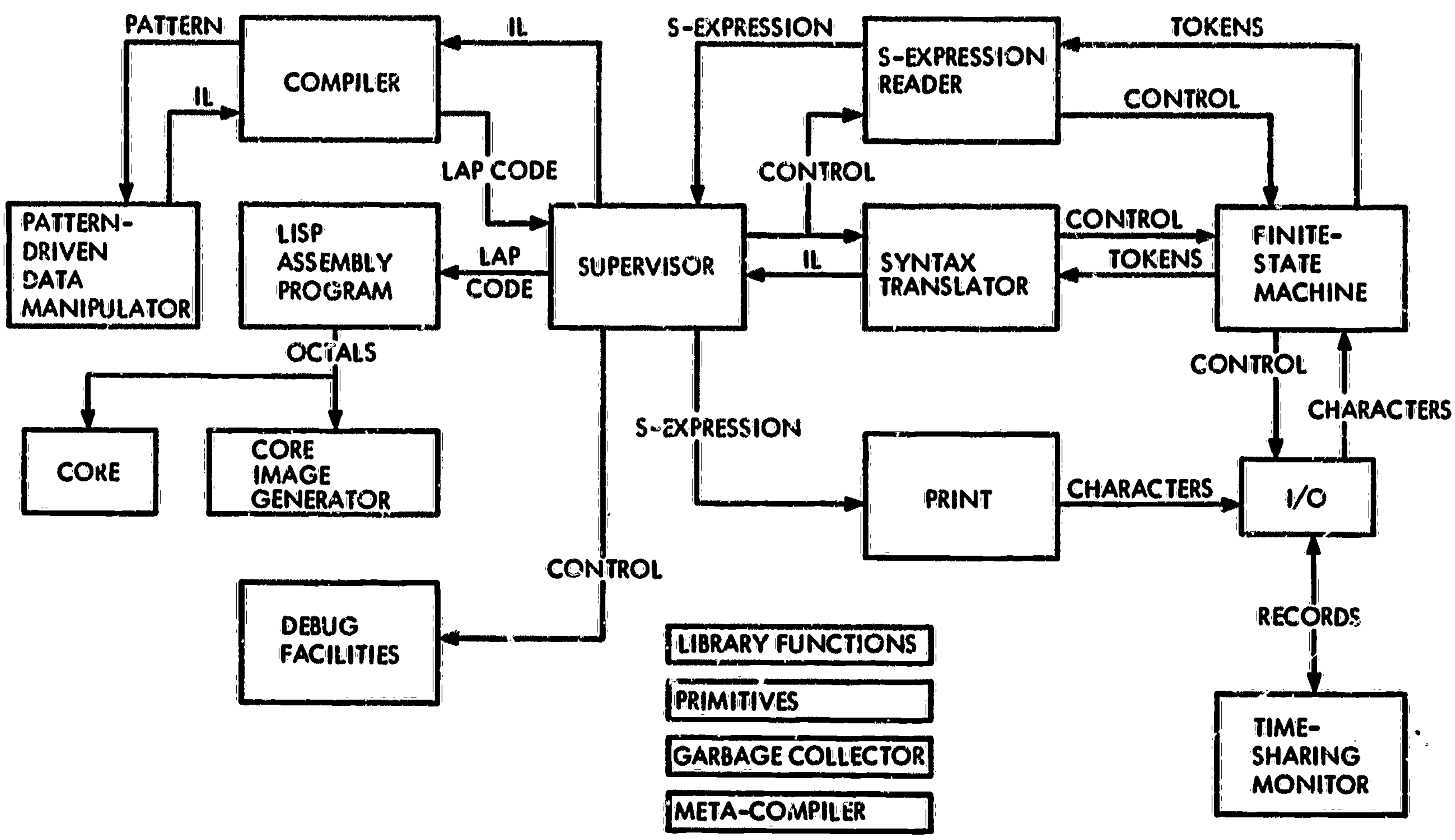


FIGURE 1-1. LISP 2 SYSTEM COMPONENTS AND INFORMATION FLOW PATHS

and communication with the monitor system of the machine on which the system is operating.

A particularly important part of the program library is a group of programs for bootstrapping LISP 2 onto a new machine. (Bootstrapping is the standard method for creating a LISP 2 system on a new machine.) The bootstrapping process, achieved by the technique of generating a complete LISP 2 core image on the target machine (core image generation), represents a major milestone in the implementation of large systems. The LAP assembler, a part of the core image generator, produces relocatable binary code. During the bootstrap process, this code is assigned an absolute core address and relocated as necessary as if it were being loaded. Associated data structures are assigned locations and their binary core images generated. The code, data structures, and binary images are then placed on an external output device such as magnetic tape. The bootstrapping capability is sufficiently powerful so that the new machine requires no resident programs other than the standard monitor and a non-relocatable binary loader to read the tape produced.

LISP 2 includes and extends the capabilities of its ancestor, LISP 1.5, which is notable for its mathematical elegance and symbol-manipulating capabilities. However, LISP 1.5 lacks a convenient input language and efficiency in the treatment of purely arithmetic operations.

LISP 2 was designed to maintain the advantages of LISP 1.5 while remedying its deficiencies. The first major change has been the introduction of two distinct language levels: Source Language (SL) and Intermediate Language (IL). The two languages have different syntaxes but the same semantics (in the sense that for every SL program there is a computationally equivalent IL program). The syntax of SL resembles that of ALGOL 60 while the syntax of IL resembles that of LISP 1.5. IL is designed to have the same structure as data, and thus to be capable of being manipulated easily by user (and system) programs. An advantage of the ALGOL-like source language is that the ALGOL algorithms can be utilized with little change.

The second major change has been the introduction of type declarations and new data types, including integer-indexed arrays and character strings. At a future time, packed data tables, which can presently be simulated through programming techniques, will be added. Type declarations are necessary to obtain efficient compiled code, particularly for arithmetic operations, but by using the default mechanisms (whereby the system automatically makes type declarations), a programmer may omit type declarations entirely (albeit at the cost of efficiency).

Figure 1-2 shows an example program, called RANDOM, in its Source Language format, in the equivalent Intermediate Language program as produced by the syntax translator, and in the LAP output from the compiler.

Progress

A first LISP 2 system was implemented on the Q-32 and demonstrated in May of 1966. The LISP 2 Intermediate Language (IL) was used for all programming of LISP 2 including the system primitives; IL was found to be powerful enough so that little or no machine language or assembly language code was required. A few minor changes were made in IL on the basis of experience obtained by project personnel in using IL. The system was produced through the core image generation process using LISP 1.5 on the Q-32 as the bootstrapping vehicle. A syntax translator for translating from source language to IL was written using metacompiling techniques (see page 1-3). A simple pattern matching routine and a LISP "pretty print" have been programmed in LISP 2 to aid in system checkout. Also, several extensions to LISP 1.5, including a context editor, were required.

<SOURCE LANGUAGE>

```
REAL SECTION TEST:
%R RANDOM COMPUTES A RANDOM NUMBER IN
%R THE INTERVAL (A,B)
DECLARE (Y←1) INTEGER OWN Y:
REAL FUNCTION RANDOM(A,B):
DO   Y←3125*Y;
     Y←Y\67108864; %R \ DENOTES REMAINDER
     RETURN A+Y*(B-A)/67108864.0;
END;
```

<INTERMEDIATE LANGUAGE>

```
(SECTION TEST REAL)
(DECLARE (Y INTEGER OWN 1))
(FUNCTION (RANDOM REAL)((A REAL)(B REAL))
  (BLOCK ()
    (SET Y (TIMES 3125 Y))
    (SET Y (REMAINDER Y 67108864))
    (RETURN (PLUS A (TIMES Y
      (QUOTIENT (DIFFERENCE B A) 67108864.0))))))
```

<Q-32 LAP LANGUAGE>

```
(LAP
 (FUNCTION ((RANDOM . TEST) REAL)
  ((A REAL LEXICAL VALUE)(B REAL LEXICAL VALUE))
  (STF TOP.)
  (BEGIN)
  (LDA (Y . TEST))
  (MUL 3125 (L567.7 R S))
  (ARGS)
  (STB PUSHA.)
  (LDA (NUMBER 67108864) S)
  (CALL (REMAINDER . LISP))
  (STF (Y . TEST))
  (LDC A)
  (FAD B)
  (FDV (NUMBER 67108864.0))
  (FMP (Y . TEST))
  (FAD A)
  (END)
  (RETURN))
 (((REMAINDER . LISP) FUNCTION
    (FUNCTIONAL INTEGER INTEGER INTEGER)VALUE)
  ((Y . TEST) OWN INTEGER VALUE))
TEST)
```

FIGURE 1-2. A SAMPLE LISP 2 PROGRAM IN SL, IL, AND LAP

Plans

The Q-32 LISP 2 system will be polished and a mechanism for swapping binary programs from disc storage will be installed. An operational Q-32 system will be completed in the first quarter of 1967.

Implementation specifications for putting LISP 2 on an IBM 360-like machine will be written. Multiple register usage, virtual memory management, and paging techniques will be included in the design work.

Several problem areas need to be resolved. The first is that LISP 2 must be polished to reduce its size. Also, the swapping mechanism has been designed to help alleviate this problem, particularly in view of future "page-turning" operating systems. The second problem involves finding a target system to which LISP 2 may be bootstrapped. Neither the IBM 360 nor the PDP-6 time-sharing system has progressed fast enough to be used as an operational vehicle for LISP 2. Lacking a stable, reliable time-sharing environment, mobility of LISP 2 onto other systems is being compromised.

Project Documentation

1. Abrahams, P. W. (III), Weissman, C. (SDC), et al. The LISP 2 programming language and system. SDC/III document TM-3163. September 26, 1966. 28 pp.

2. Kameny, S. L. and Weissman, C. The Q-32 LISP 1.5 mod. 2.6 system: Operating system, input-output, file, and library functions. SDC document TM-2337/103/00. April 11, 1966. 27 pp.

3. Levin, M. I. and Berkeley, E. (III). LISP 2 primer. SDC document TM-2710/101/00 (Draft). July 15, 1966. 165 pp.

4. Firth, D. C. and Kameny, S. L. Syntax of LISP 2 tokens. SDC document TM-2710/210/00. August 25, 1966. 11 pp.

5. Kameny, S. L. (SDC) and Hawkinson, L. (III). LISP 2 intermediate language. SDC document TM-2710/220/01. July 5, 1966. 56 pp.

6. Barnett, J. A. SIM, an s-expression pattern-matching function. SDC document TM-2710/260/00. June 29, 1966. 6 pp.

7. Saunders, R. A. (III), Barnett, J. A. and Firth, D. C. (SDC). The LISP 2 compiler. SDC document TM-2710/320/01. February 1, 1966. 55 pp.

8. Book, E. The LISP 2 syntax translator. SDC document TM-2710/331/00. April 15, 1966. 27 pp.

9. Howard, M. V. Operating instructions for the LISP 2 supervisor in the LISP 2 core image. SDC document TM-2710/510/00. October 14, 1966. 5 pp.

Data Base Oriented Programming Language*

E. B. Foote
R. G. Howard

Description

Data management systems, such as LUCID (see p. 4-7), have several features distinctly different from procedure-oriented compiler systems such as FORTRAN or JOVIAL. The first is the English-like restricted language of the former. This language is valuable because it is user-oriented and easily learned. However, it lacks certain power inherent in the procedure-oriented languages. The second difference--not as apparent to the user but equally significant--is the organization of the data in storage. Although languages like FORTRAN and JOVIAL allow flexibility in what is stored, and in the hierarchical structure between data elements, they put the material away, in core or tape or disc, in a relatively rigid format. Data management systems like LUCID, on the contrary, have a very flexible format, dictated by the data itself. The goal of this project is to study the possibility of obtaining in one system the power of the procedure-oriented language and the capability offered by the data structure used in the data base management system.

Progress

The language is being investigated in the context of SDC's Time-Shared Data Management System (TDMS)--(see p. 4-9). The first step has

*Supported by the Advanced Research Projects Agency.

been to study the possibility of building a convenient bridge for the user between a data management system and a procedure-oriented language; that is, to enable a user who has built a data base using the manipulative features of TDMS to use the same data base for more extensive computations. For example, a user who has built and used an inventory of bomb components might wish to use linear programming to determine the most bombs he could build out of components in his inventory. Such a lengthy computation exceeds the capacity of TDMS. The user would need to use something like JOVIAL but would prefer not to have to convert his data base to a new format.

The use of the TINT* system on the Q-32 computer was considered as the procedure-oriented language base since TINT is more readily modifiable than available compilers. Several new language forms were devised to carry data back and forth between the formats of TDMS and TINT. However, taking material from the tree structure used in TDMS into tabular format of TINT can result in wasting of a great deal of space, because the tree structure does not allow space for absent elements while the table must allow space for the largest potential entry. Other approaches were considered and several appear promising. One of these is to add a facility to a compiler such as JOVIAL that would enable a user to get elements from a data base as he needs them. Then the user can perform computations on the elements immediately or construct tables based on his knowledge of the data.

Plans

The JOVIAL compiler for SDC's S/360 Time-Sharing System will be studied to determine if it can be matched to the data retrieval package of the TDMS system. If this appears feasible, the compiler will be modified to furnish a vehicle for experiment and further development.

*Teletype INTerpreter--an on-line assembly-oriented language system.

AIDS TO PROGRAMMING

Interactive Programming Support System*

M. B. Bleier
H. Bratman
E. R. Clark
J. S. Hopkins

Description

The goal of this project is to develop a programming support system that (1) takes advantage of the interactive capability of time-shared systems, (2) facilitates the programmers' work by providing an integrated capability in system language, information, and function, and (3) makes advantageous use of a tabular display in program composition, program editing, program debugging, and program and system documentation.

Among the functions that will be provided in the system are:

Program Composition:	Original preparation of source-language program
	Alteration (temporarily or permanently) of source-language program
	Maintenance of program and data files
Program Compilation:	Grammar checking
	Partial recompilation
	Compilation of optimum object code
Program Testing:	Display of variables that are selected by symbolic name
	Symbolic alteration of values of variables
	Tracing
	Assistance in preparing and storing data for program parameter testing
	Display of variables controlled by conditional statements

*Supported in part by Rome Air Development Center.

Documentation:

- Production of "set-used" information and glossary of names used in the program
- Reformatting of source-language program text
- Production of flow charts containing variable levels of program details
- Assistance to user in learning the system

The system is designed to give the user an integrated language for the program production processes of composition, editing, compilation, execution, testing, and documentation. The commands for performing a particular action are all in the same format, even though the action may occur in different functions. For example, once a programmer has learned the commands to insert and delete statements in a program, he is able to perform simple kinds of file maintenance without learning new or contradictory commands.

Also, in developing a program, the user will deal with only a single entity, the system itself, instead of being concerned with a compiler, an editor, a debug package, etc. He can perform actions as needed without awareness that different functions are being used. For example, if the user gives a command to execute a program and that program has not been compiled, the system will compile it for him. The system is designed for programmers rather than problem solvers. However, the system and its language are designed to be learnable in steps, so that a novice with a simple problem would need to learn only a small part of the system.

In addition to providing an integrated language, the system exploits the capability of time-shared interaction, especially during compilation. During compilation the compiler usually makes some arbitrary decisions. For example, when there are more index variables than registers, the compiler uses predetermined rules to assign registers. In an on-line, interactive situation, the compiler can ask the programmer for advice. If the programmer has information about the frequency of use of some or all of the indexes, the compiler can use this information to make more efficient assignments.

Another feature of the system is improved coordination of programs, providing better service to the user. Increased efficiency in operating time is expected, because programs within the system do not regenerate information that has already been computed. In particular, the compiler provides a great deal of information that is useful in debugging. An example is the dictionary generated during compilation, which is currently being used by some debugging routines. The flow chart and glossary programs need much the same information about the structure of a program generated by a compiler with line-at-a-time or partial recompilation capability. Figure 1-3 shows the relationship between system components.

Progress

Investigation is proceeding along three paths:

. Design of the system language and the techniques of user-system interactions.

. Design of the system functions that support the system capabilities.

. Design of system control and program integration.

1. Design of the System Language. The conversational requirements of various components of the system, such as program composition, compilation, testing, and documentation are being studied. As the requirements are formulated, an integrated language is being developed. The development of the system design concurrent with the language design ensures that the system contains all the capabilities implied by the language. A document has been published that describes the system from a user's point of view [3].

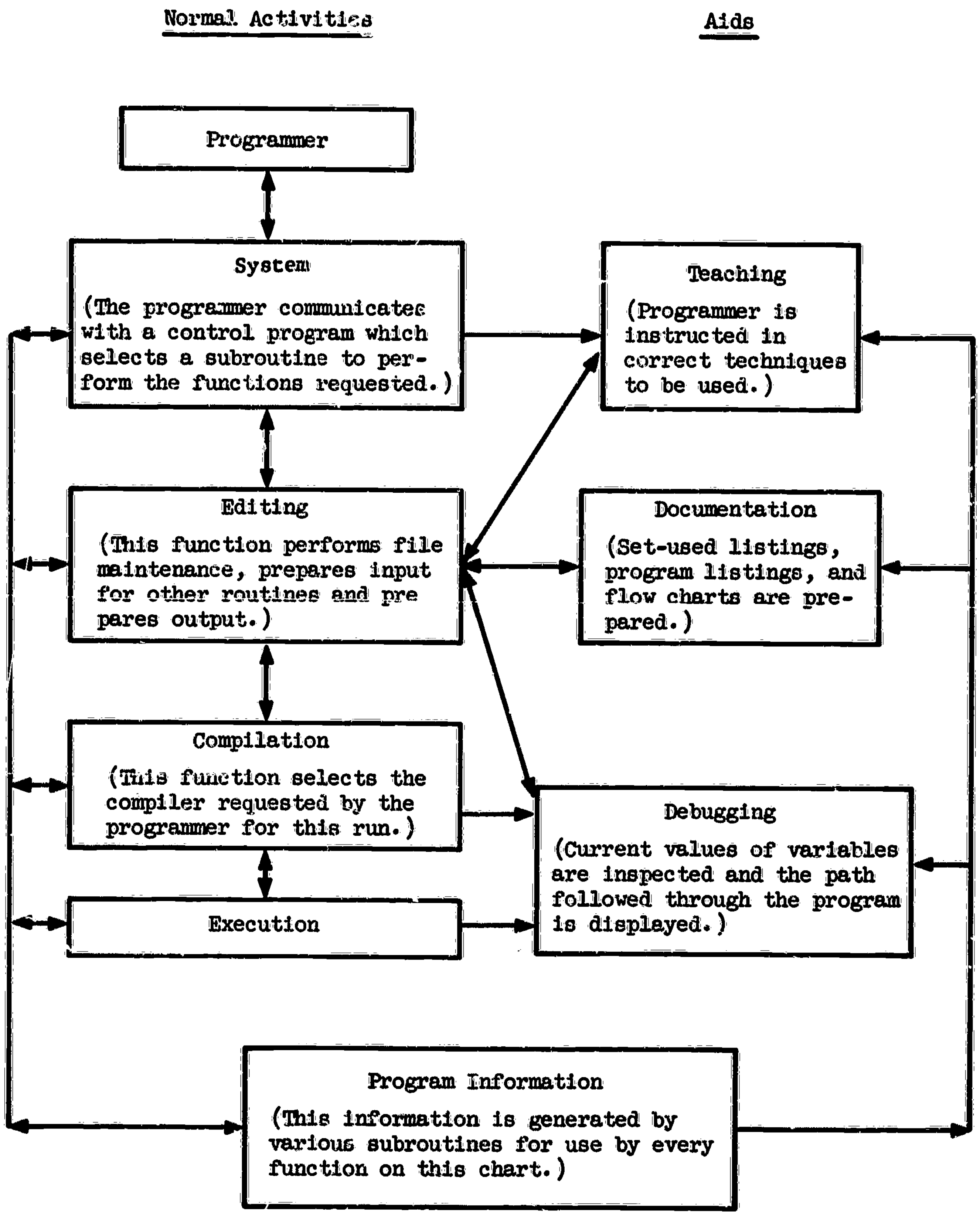


FIGURE 1-3. COMPONENTS OF THE INTERACTIVE PROGRAMMING SUPPORT SYSTEM

2. Design of Various System Functions and Tables. A version of the syntax analyzer is being checked out on the Q-32. This component performs the following functions:

. Analyzes syntax of all inputs to see if they are correct and indicates location of error if they are not.

. If the input is a system command, encodes it in binary and searches the dictionary for any variable names in the command.

. If the input is a JOVIAL statement, removes redundant blanks from the statement and makes entries in the dictionary for variables used or defined.

The syntax analyzer is being written in JOVIAL and uses techniques developed by the META and META5 metacompilers (p. 1-3). This will result in an efficient program that can be easily modified for changes in input syntax.

Formats that facilitate the attaching and detaching of program units and system commands to the main body of the program have been proposed for the Text Structure Directory (TSD) and the Program Structure List (PSL). Algorithms have been devised to attach, locate, and trace program units in the proposed TSD format. A scheme to encode system commands, consistent with the proposed TSD format, has been devised.

Research is being conducted in the area of partial recompilation, which combines the problems of program editing and incremental compilation with those of modifying executable programs. The program text is maintained in a list form to facilitate program editing. Statements are compiled into independent subroutines whose computation sequence is described in the PSL. An execution monitor is being designed to operate on the output of the compiler. This component executes the independent program segments in the proper sequence and directs the operation of any program test requests made by the user.

3. Design of System Control and Program Integration. A central controller and input/output control are being designed to operate within SDC's S/360 Time-Sharing System. A study of the Time-Sharing System is under way to determine its optimum use with the Interactive Programming Support System. A prototype controller will be written to operate on the 360 with dummy system components.

Plans

Work on the Interactive Programming Support System will concentrate on completing the system design and building the framework of the system. The aim will be to build a mockup of the system, with certain parts "running" and certain parts only represented, in order to meet the various constraints on resources. The emphasis will be on developing the system language and showing the forms of user interaction. The aim is to gain experience in using the system and testing the effectiveness of the program composition and editing language and function.

The tasks to be worked on in the immediate future include:

. Development of the system language.

. Design and construction of the program composition and editing programs.

. Completion and revision of the syntax analyzer.

. Design and construction of the system control program.

. Modification of the existing JS S/360 compiler to be operable under the control program, match the output of the editing program, provide some interactive capability, and match existing testing programs.

Project Documentation

1. Jacobs, E. H. An interactive programming support system. SDC document TM-2819. January 1, 1966. 4 pp.

2. Beeler, R. G., Jackson, C. W., Hopkins, J. S. and Jacobs, E. H. Interactive programming support system. SDC document TM-2819/001/00. February 21, 1966. 9 pp.

3. Bratman, H. and Hopkins, J. S. Initial design specifications for the interactive programming support system. SDC document TM-2819/002/00. July 20, 1966. 54 pp.

Automated Flow Charting

L. Fine

Description

The flow chart is a traditional programming tool used in program design, production, checkout, maintenance, and documentation. It provides a means for correlating what a program actually does with what it is meant to do. The level of detail desired in a flow chart is a function of its intended use. For some purposes a generalized flow chart, giving an overview of a program, is most useful, while for other purposes a detailed flow chart, giving the exact flow of a program, is most appropriate. However, the production and maintenance of such flow charts is often in itself an overwhelming, and hence rather neglected, task. For this reason, a program that takes a symbolic program and from it automatically produces multilevel flow charts can serve an important purpose. In addition to relieving the programmer of the tedious aspects of flow diagramming, such a program produces flow charts that have a consistent degree of accuracy and a standardized format.

This project is developing a method for automatically producing multilevel flow charts. The most detailed flow chart is first produced and then, by iteratively applying a set of algorithms, it is successively condensed to produce flow charts at various levels of detail.

Progress

As a start on the problem of analyzing symbolic code, a program (SURE) was written which would take a program written in JTS (JOVIAL for Time-Sharing) and produce complete set/used information. This program was then augmented to perform more detailed analysis and was found to be useful in improving the symbolic code. That is, SURE can reformat a program so that it will be shorter and will execute faster. This work appeared so promising that a separate project aimed at automatic code improvement was started (see p. 1-21) while the automatic flow charting work continued.

.. method of producing multilevel flow charts has been implemented in JOVIAL. It is currently operating on the Q-32 computer, and will be adapted to operate on SDC's S/360 Time-Sharing System. The program takes a source program written in a subset of JOVIAL, determines the function of each source program statem^nt, and assigns to each statement (other than a branch-type statement) a separate box that indicates its function. Text that describes the action of the source statement is placed in each box, and flow information is associated with each box. Finally, certain adjacent boxes are grouped together, thus producing the most detailed flow chart.

A study of the number of entries to, and exits from, each box, as well as some information about program flow, indicated that certain configurations of boxes can be grouped together without essentially altering the picture of program flow. This study led to the development of a set of seven rules that, when applied to the flow chart, condense it by grouping together certain configurations of boxes and thus produce a more generalized flow chart. By reapplying the rules to this flow chart the next-level chart is produced. The seven rules specify how different boxes are combined, how the text is modified, and how the flow information is updated. Initially, simple program structures are grouped together and then more complex structures are combined, until finally the flow chart is in its most generalized form.

The method and the program that implements it are still in the developmental stage. The program checkout has been completed ; it very little experimentation has been done. The results of applying the rules are encouraging but the rules do not reduce all source programs to a most generalized flow chart consisting of a single box.

The output of the flow charting program is a printer-drawn flow chart. No attempt has been made to produce a well-drawn picture, since this project is primarily concerned with the analysis techniques. A number of researchers are working on the "drawing" problem and it is expected that this project will utilize their results.

Plans

The next step will be to generate a number of computer-produced flow charts to be used in evaluating and refining the method. A study will be made to see if the seven rules can be extended to reduce all source programs to one-box representations. Further investigation will also be directed toward determining what constitutes a flow chart level.

The present output will be replaced with a more sophisticated "picture-drawing" capability, preferably by adapting one of the flow charting programs already implemented outside of SDC, and an on-line display console will be used as an output device. Long-range plans include an investigation of user interaction with the program, a feature that is not possible now.

Project Documentation

1. Fine, L. Automated flow diagramming--boxes and text. SDC document TM-2969/001/00. April 18, 1966. 67 pp.

2. Fine, L. Automated multilevel flow diagramming. SDC document SP-2629. October 26, 1966. 19 pp.

Graphic Input/Output*

M. I. Bernstein

Description

The primary purpose of this project is to extend on-line programming facilities and techniques through the use of graphic input/output equipment--namely, the RAND Graphic Input Tablet, CRT displays, and their associated hardware. The attainment of this goal requires the development of character and shape recognition routines that function efficiently in an on-line time-shared environment and the integration of these routines into both current on-line programming systems and extended programming systems. To be useful, the character recognition routines must not only be able to recognize inputs from a large variety of users, but must provide the users with a relatively large character set. The extended systems include languages for which the placement of characters is more meaningful and natural in a two-dimensional format; for example, in flow charts, automated analysis and manipulation programs, and theorem-proving and game-playing programs.

Progress

The central ideas of the character recognition techniques being developed are that maximum use be made of the serial nature of the real-time input data and that the principal unit of information is the stroke. The program extracts descriptors in the form of "feature" strings from each stroke as it occurs. Multistroke characters require that the strokes constituting them have the proper spatial relations.

In order to tailor the character recognition programs to a given user, they are being constructed to allow each user to provide his own input vocabulary of characters and to associate these with the desired output character. During this procedure a control program constructs a

*Supported in part by the Advanced Research Projects Agency.

dictionary for the user and allows him to test the level of achievement in recognition. If the user is unsatisfied with the results, he may add samples until the recognition rate is acceptable.

To date, four feature extractors and two methods of analyzing the spatial relations between strokes have been programmed for the Philco 2000 using a RAND Graphic Input Tablet for input and its associated display for output.

One extractor generates a "feature" each time it detects a local minimum or maximum in the X or Y coordinates (i.e., each time the writing instrument reverses direction of its left-right or up-down motion). Another divides the stroke area (the rectangle circumscribing the stroke as defined by the absolute minimum and maximum values of X and Y) into four equal areas upon completion of the stroke. The "feature" list consists of the subareas through which the stroke passed in order of occurrence. A third "feature" extractor divides the stroke area into five subareas by superimposing a diamond-shaped area at the center of the rectangle. The fourth extractor fits straight-line segments to the stroke path and measures local curvature in the path by computing the angular change between adjacent line segments. "Features" in this case are a function of the collected local angular changes that are all in the same direction of rotation (clockwise or counterclockwise). Collection is terminated by a change in direction of rotation, by a sharp angular change regardless of direction of rotation, or by the end of the stroke. The "features" themselves are classifications based upon the amount and direction of curvature.

Of the two methods of analyzing the spatial relations between strokes, only one has proved useful to date. It classifies the relationship of a pair of consecutive stroke rectangles into one of three classes: coincident, proximate, or unrelated. To those classified as proximate or unrelated, a quantized direction is added. For this purpose direction is quantized into eight headings.

The other analyzer, which generates the relationship between the imaginary line segments formed by joining the stroke end points, has been temporarily laid aside until more work can be done to provide better discrimination.

The feature extractors and the first spatial analyzer have been integrated in a basic control package which handles the input/output chores and the construction of the stroke-character dictionary. The dictionary is a tree. All characters with common beginning strokes have the same path in the tree, no branch occurring until a difference occurs. Thus, a "1," a "P," and an "R" occupy one path in the tree, with the "1" and the "P" as intermediate and defined sub-elements of an "R."

Tests with each of these techniques showed a variety of shortcomings.

The feature extractor based upon the detection of local minima or maxima in X and Y was overly discriminating and required too many samples for adequate recognition. The area feature extractors were unable to consistently discriminate between all the 99 characters in the test set. The extractor based upon measurements of path curvature and rotation was not completed but appeared to be overly discriminating.

Several techniques for better smoothing and filtering were tried in order to improve the min-max extractor, in the belief that some of the problem was due to noise. This did not prove to be the case and further work on this extractor has been suspended.

Using similar techniques on the curvature feature extractor showed that noise was not its principal problem, but rather that there was some inconsistency in determining the proper termination point for each feature in complex characters. In an attempt to solve this problem, a corner

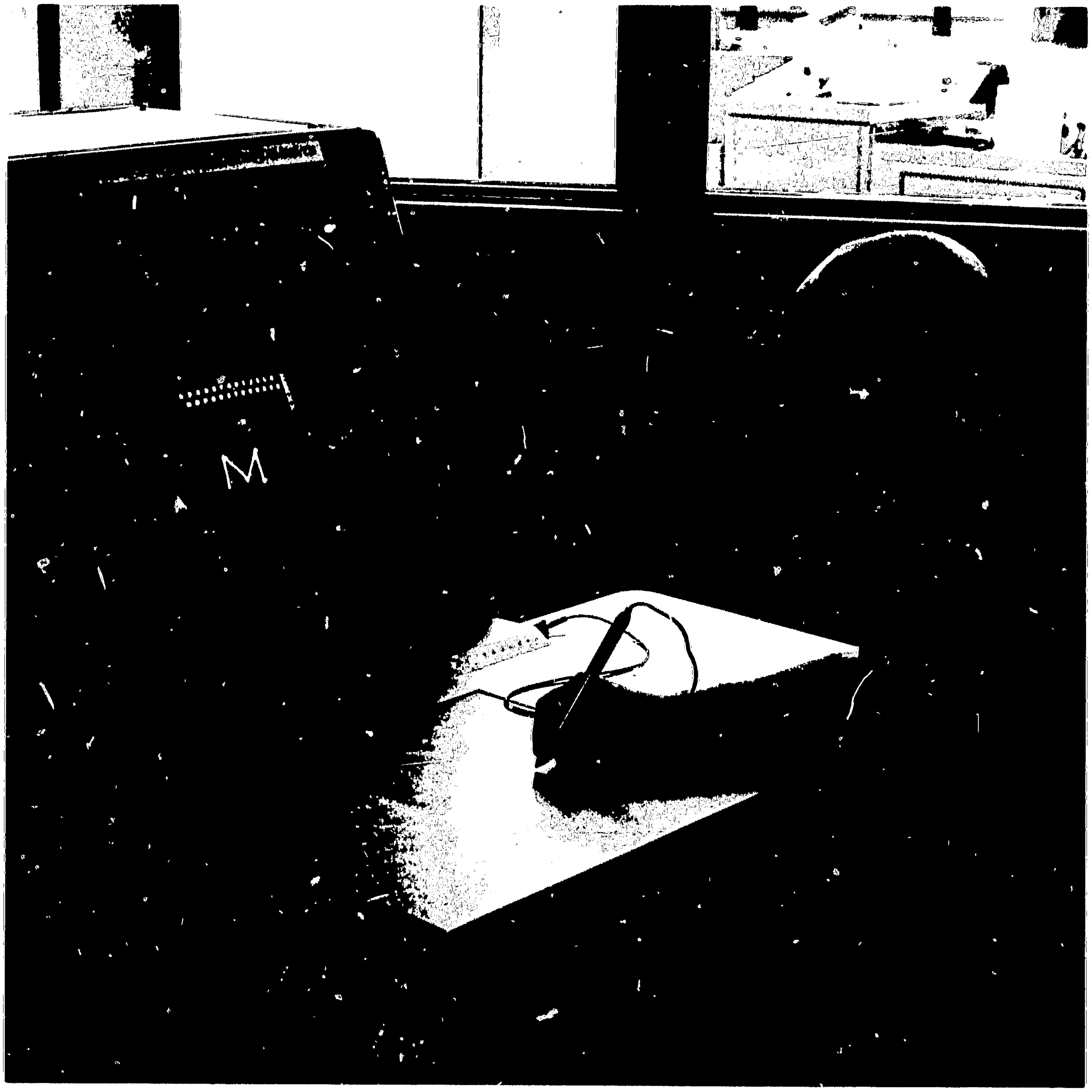


FIGURE 1-4. GRAPHIC INPUT/OUTPUT EXPERIMENTATION

Experimenter writes character on RAND Tablet with stylus. The display scope shows the large hand-drawn character, plus the computer-recognized matching character immediately above it.

detector was developed that looked at both the local geometry and path velocity for clues in locating a corner. The corner detector proved successful and was incorporated in both the area feature extractor (using five subareas) and the curvature feature extractor. This addition enabled the area feature extractor to provide adequate recognition for at least one user on a set of 99 characters that includes the upper and lower case Roman alphabet, lower case Greek alphabet, 10 digits and 13 punctuation marks. The computer on which the work was being done (the Philco 2000) was turned off before completion of modifications and testing could be done on the curvature extractor.

Plans

In an attempt to smooth the transition from the Philco 2000 to the IBM S/360, some RAND Tablet outputs of drawn characters were recorded on tape and converted to IBM format. This tape will permit preliminary work to proceed on the S/360 without full hardware capability and will provide a controlled mechanism for program improvements and testing.

Both the curvature and area feature extractors will be programmed for the IBM S/360 to work under the SDC Time-Sharing System, initially incorporating the most obvious improvements. When the programs are brought to the appropriate state, the existing versions will be tested for various capabilities, with a typical set of potential users. Work will also begin on imbedding the recognizer in a system that allows a programmer to prepare a computer routine directly on the RAND Tablet.

Project Documentation

1. Bernstein, M. I. An on-line system for utilizing hand-printed input. SDC document TM-3052. July 11, 1966. 19 pp.

2. Bernstein, M. I. Some system considerations for on-line character recognition in a time-sharing system. SDC document SP-2636. October 28, 1966. 11 pp.

Automatic Code Improvement

E. R. Clark

Description

Almost any program of a certain minimal complexity can be improved. Improvement may mean making the program shorter in storage space required, faster in execution time, or rearranged so that the program logic is easier to understand. Undoubtedly, these improvements can be done most effectively by an experienced programmer who understands what the program is to do. Since such people are not always available, and, even if they are, the process is time consuming and expensive, a processor has been written to analyze a program automatically and attempt to improve it. The processor needs no understanding of what the program is supposed to do; its object is to produce a better program that will do the same thing.

Progress

A program called SURE (Set-Used REformatter), has been developed which accepts a program written in JOVIAL, automatically looks for any of 11 specific situations, and improves the program if any of these are found. The particular methods implemented were selected by weighing the difficulty of detecting and improving a given situation against the likelihood that such a situation will occur. The improvements made in a well-written program may not be very significant, but the processing time of SURE is not great, and the processed program is permanently improved since the changes are made at the symbolic language level. A poorly written program can, of course, benefit even more.

The original version of SURE generated an intermediate language which was processed for possible improvements. SURE was rewritten to make its improvements by working directly on the symbolic statements. This version makes the same improvements as before but, not requiring an intermediate language, is about one-third shorter. SURE originally accepted the JTS version of JOVIAL as input. It now accepts the full J3 language in addition to the special primitives belonging to JTS.

Plans

SURE will be modified to run under SDC's S/360 Time-Sharing System. Additional methods of automatic improvements might be implemented if the need for a particular improvement is demonstrated.

Project Documentation

1. Clark, E. R. Executive service: Set/used reformat processor (SURE). SDC document TM-2708/205/00. April 7, 1966. 13 pp.
2. Clark, E. R. On the automatic simplification of source-language programs. SDC document SP-2389. April 11, 1966. 20 pp.

COMPUTER PROGRAMMING MANAGEMENT

Program Cost Analysis*

T. Fleishman
V. LaBolle
E. A. Nelson
G. F. Weinwurm
H. J. Zagorski (Defense Systems Division)

Description

The aim of this project is to develop techniques, standards, and guidelines for managers of computer programming projects, to aid them in planning, controlling, and estimating the costs of computer programming jobs. This work has been characterized by three major steps, namely: (1) the collection of data that detail the costs and cost factors for completed programming jobs by means of a questionnaire, (2) the validation of these data to eliminate errors in the data, and (3) the use of statistical techniques, e.g., multivariate regression, to derive numerical relationships, primarily linear equations, for estimating the costs of proposed computer programming projects.

*Supported by the Air Force Electronic Systems Division, Directorate of Computers.

Progress

Using the foundation provided by earlier results* from an analysis of data on 74 programming projects completed at SDC, project members did additional work to derive estimating equations that were both easier to use and more accurate. To make the equations easier to use, the logarithmic transformation adopted earlier for some variables was dropped--especially for the major cost measures used as dependent variables, i.e., man-months, computer hours, and elapsed time in months. As a result, a derived statistic such as the standard error of estimate for man-months could be measured in those units rather than in their log, and the meaning of the results could be readily interpreted. To improve the accuracy of the equations, seven data points with excessive cost measures were eliminated. This truncated sample was used to derive equations that reduced the strong influence of these outliers. To further increase the statistical precision, the remaining total sample (N = 67) was divided into three subsamples based on the program size measured in man-months.

Concurrent with this work, an effort was under way to validate additional data collected earlier to form a more homogeneous and representative sample from sources other than SDC. These new data were to be merged with the SDC data to form a larger data base that would be analyzed in similar ways. The new analysis was to stress subsampling as a way to derive more accurate equations.

In the latter part of 1965, over one hundred data points from completed programming efforts were collected from 8 industrial organizations and 4 U. S. Air Force agencies. After authentication of these data in the spring of 1966, the newly acquired data points were combined with the SDC

*See SDC document TM-2712, Research into the Management of Computer Programming: A Transitional Analysis of Cost Estimation Techniques, November 12, 1965.

data on hand, to form a total data base of 169 data points as inputs to the analysis.

Four subsamples were analyzed as meaningful ways to divide the data in terms of costs: (1) Programming Application--a division of programs into categories--Business, Scientific, Utility and Support, and Other (a miscellaneous category such as command and control, research and development); (2) Program Source Language, a separation between programs written in machine-oriented languages and procedure-oriented languages; (3) Production Computer Size, a division based on equivalent purchase price of small, medium, and large computers; and (4) Stand-Alone versus System Program, "one-shot" programs produced as single entities versus programs created as integral parts of a larger information processing system.

Within each of the four groups, statistical tests of significance were performed on the means of the following variables: Man-Months, Computer Hours, Elapsed Time in Months, Number of Object Instructions, Source and Object Production Rates (Instructions/Man-Months), and Source and Object Computer Usage Rates (Computer Hours/1000 Instructions). The results of these tests on the data (see Table 1-1) showed the following:

. Utility and Support programs are more costly to produce than the other three applications.

. Business (file-oriented) programs are less costly than the three other program applications.

TABLE 1-1. MEAN COSTS, PRODUCTION RATES, AND COMPUTER USAGE RATES BY APPLICATION

Application	Number of Points	Mean Man-Months	Mean Cmptr Hrs	Mean Object Instr/Man-Month	Mean Computer Hrs/1000 Object Instr
Business	79	13.2	73	1521	12
Scientific	27	42.0	137	882	18
Utility and Support	28	92.7	766	410	57
Other	35	54.7	263	292	30

. Procedure-Oriented Languages are more effective (see Table 1-2), i.e., have lower resource use, object instructions, and computer usage rates, than Machine-Oriented Languages.

. The average expansion ratio is approximately 3.3 Machine Language Instructions to one Procedure-Oriented Language Instruction.

TABLE 1-2. MEAN COSTS, PRODUCTION RATES, AND COMPUTER USAGE RATES BY PROGRAMMING LANGUAGE

Application	Number of Points	Mean Man-Months	Mean Cmptr Hrs	Mean Object Instr/Man-Month	Mean Computer Hrs/1000 Object Instr
Machine-Oriented	123	48	289	610	30
Procedure-Oriented	46	18	99	1977	10

The data for the other two subsamples, Computer Size and Stand-Alone/System, showed no conclusive results.

Several sets of estimating equations were derived for the total sample and for a number of subsamples. These results, as well as others, were summarized in a handbook for estimation of computer programming costs. The equations for the subsamples presented in the Management Handbook [3] exhibit better statistical precision than the equations for the total sample.

The Management Handbook also contains other material based upon technical literature and the experience of project members. These guidelines are intended to help managers estimate the cost of computer program development. In the handbook the computer programming process is divided into six categories: Preliminary Planning and Cost Evaluation; Information System Analysis and Design; Computer Program Design, Code, and Test; Information System Integration Test; Information System Installation and Turnover; and Computer

Program Maintenance. Each of these process steps is described in terms of tasks, inputs, and outputs. For each step a number of cost factors are listed together with some statistical and/or intuitive indication of their influence on costs. Planning factors such as unit costs or percentage-of-other-item costs are also given for the process steps. Examples of forms for recording cost estimates are also included.

This Handbook should be interpreted as an initial effort to present the manager with a comprehensive set of applicable guidelines. If feedback indicates that guidelines in this form are useful, the Handbook should be supplemented and revised as more information becomes available through further research and development.

Plans

No further analysis of the collected data is scheduled at this time; however, the analytical results to date and the data base could be used in future project endeavors, such as the one planned for the first part of 1967, namely, the continued development of an operational system to collect cost data during the program development process.

Project Documentation

1. LaBolle, V. Development of equations for estimating the costs of computer program production. SDC document TM-2918. April 5, 1966. 49 pp.
2. Fleishman, T. Current results from the analysis of cost data for computer programming. SDC document TM-3026/000/01. July 26, 1966. 97 pp.
3. Nelson, E. A. Management handbook for the estimation of computer programming costs. SDC document TM-3225. October 31, 1966. 141 pp.
4. LaBolle, V. Development of aids for managers of computer programming. Journal of Industrial Engineering. 1966, 17 (11), 564-571.

A System for Reporting Cost Data for Computer Programming*

L. Farr (Advanced Systems Division)
T. Fleishman
V. LaBolle
E. A. Nelson
C. L. Starkey (Defense Systems Division)
G. F. Weinwurm

Description

In the work to develop estimating equations for computer program development costs (see p. 1-22), one major difficulty was obtaining data that relate products to costs. Even when these data were available, they showed the poor quality of "after-the-fact" data--unstructured, ambiguous, and disorganized. Recorded data were aimed mainly at organizational or contractual accounting and not at planning and control of computer programming projects. The reporting system being developed is intended for use during a computer programming project. In addition to promoting the recording of uniform data that can be compared from project to project, the system should provide information for cost control of individual projects and inputs to a data bank that can be analyzed to supply improved planning factors.

Progress

The initial effort in the development of a reporting system was completed during the first part of 19 6. This work used a definition of the computer programming process as a context in which to identify and define proposed data elements to be collected in the system.

The programming process was broken down into the following seven steps: Information Processing Analysis; Information Processing Design; Computer Program Design; Computer Coding and Checkout; Computer Program Functional Test; Information Processing Integration Test; and Information Processing Installation and Implementation.

*Supported by the Air Force Electronic Systems Division, Directorate of Computers.

Each step identifies a stage in which a subset of the proposed set of data elements is to be collected.

Several forms were developed to collect the data at each process step. Separate forms were proposed for (1) collection of both cost data and technical data, (2) tracking of estimated and actual costs and cost factors through the life cycle of a computer programming project, and (3) formation of a quantitative history of resource expenditure patterns.

The system is intended to provide a basis for collection of comparable (uniform) cost and technical data from computer programming development projects whether performed "in-house" by the Air Force or by a subcontractor.

The first version was designed to be compatible with existing management and budgetary systems, e.g., the Program Budget, Cost Information Reports (CIR), and System Program Management Procedures, as described in the AFSCM 375 series. As part of a task to consult for the SAC Airborne Data Automation Project, members of the Programming Management Project made recommendations for data collection on computer programming work.

Plans

Review of the completed work by managers has suggested areas of potential improvements, such as clarifying the definitions of the cost and technical data items. The continued work is aimed at making these improvements. Plans for the early part of 1967 call for testing the feasibility of the system in an Air Force agency that is responsible for a large number of programming projects.

Project Documentation

1. Weinwurm, G. F. Data elements for a cost reporting system for computer program development. SDC document TM-2934/000/02. July 6, 1966. 83 pp.

2. LaBolle, V. and Fleishman, T. Programming management project consulting for the SAC airborne data automation project. SDC document TM-3094. August 18, 1966. 24 pp.

COMPLETED STUDIES

The following studies in the Advanced Programming area were completed prior to 1966 and are not described in this report.

Compiler Construction Techniques

1. Book, E. The LISP version of the META compiler. SDC/III document TM-2710/330/00. November 2, 1965. 11 pp.

2. Book, E. and Bratman, H. Using compilers to build compilers. SDC document SP-176. August 31, 1960. 11 pp.

3. Book, E., Bratman, H., Schwartz, J. I. A one-pass JOVIAL compiler. SDC document TM-970/001/00. January 22, 1963. 3 pp.

4. Book, E., Bratman, H., and Schwartz, J. I. JOVIAL-X.2, the language of the one-pass JOVIAL compiler. SDC document TM-970/002/00. January 31, 1963. 11 pp.

5. Cohen, V. L. and Hopkins, J. S. Phase I of the JOVIAL generator (NGENI). SDC document TM-555/021/01. March 17, 1965. 343 pp.

6. Dobrusky, W. B. Design for JOVIAL compiler for the small computer. SDC document TM-739. August 7, 1962. 9 pp.

7. Englund, D. E. and Clark, E. R. The CLIP translator. Communications of the ACM, 1961, 4, 19-22. (Also available as SDC document FN-4455.)

8. Oppenheim, D. K. The META5 language and system. SDC document TM-2396. July 21, 1965. 49 pp.

Programming Languages

1. Foote, E. B. and Sandin, N. A. JTS users' manual. SDC document TM-1577/000/01. April 8, 1965. 63 pp.

2. Isbitz, H. A formal description of CLIP. SDC document TM-543. October 14, 1960. 18 pp.

3. Kameny, S. L. LISP 1.5 reference manual for Q-32. SDC document TM-2337/101/00. August 9, 1965. 85 pp.

4. Kameny, S. L. Input-output file and library functions: The Q-32 LISP 1.5 mod. 2.5 system. SDC document TM-2337/102/00. September 22, 1965. 14 pp.

5. Perstein, M. H. The JOVIAL (J3) grammar and lexicon. SDC document TM-555/002/04. October 20, 1965. 138 pp.

6. Shaw, C. J. A comparative evaluation of JOVIAL and FORTRAN IV. Automatic Programming Information, 1964, (22), 1-15.

7. Weissman, C. LISP primer: A self-tutor for Q-32 LISP 1.5. SDC document TM-2337/010/00. June 14, 1965. 166 pp.

Computer Programming Management

1. Farr, L., LaBolle, V., and Willmorth, N. E. Planning guide for computer program development. SDC document TM-2314. May 10, 1965. 179 pp.

2. Farr, L. Quantitative analysis of computer programming cost factors: A progress report. SDC document SP-2036. August 26, 1965. 19 pp.

3. LaBolle, V. Office of naval research/computer program implementation process: Final report. May 1965. SDC document TM-1954/004/00. June 3, 1965. 25 pp.

4. Nelson, E. A. Research into the management of computer programming: Some characteristics of programming cost data from government and industry. SDC document TM-2704/000/00. November 15, 1965. 43 pp.

5. Peach, P. Quality control for computer programming: A final report on an initial study. SDC document TM-2313/001/00. September 9, 1965. 17 pp.

6. Weinwurm, G. F. Research into the management of computer programming: A transitional analysis of cost estimation techniques. SDC document TM-2712/000/00. November 12, 1965. 203 pp.

INFORMATION PROCESSING RESEARCH

T. B. Steel, Jr., Head

The activities of the Information Processing Research staff for the past year fall into three main categories. First, there are studies in the area of formal models of information processing concerned with the development and analysis of theoretical descriptions of various aspects of information processing systems. Second, there is an effort to study and develop information processing systems intended to promote more effective cooperation in man-machine teams in problem-solving contexts. Finally, some work is continuing in attempts to advance the frontier of support technology in the area of programming languages. This last work is reported under the topic of Advanced Programming (see p. 1-1 et seq.).

The importance and relevance of the first class of projects derive from the phenomenal growth of the information processing field. As a result of this growth, most developments in technique have been ad hoc and empirical; theory is virtually nonexistent. The history of science strongly suggests that, unless the development of a theoretical framework begins to catch up with ad hoc studies, and organize them, progress will slow down. Thus, the principal objective of these projects is the pursuit of theoretical investigations into the nature of information processing procedures. While the means of investigation may occasionally be empirical and particular, the objectives remain theoretical and general.

A common theme among this first set of studies is the development of abstract models, covering machines, programming languages, procedures, and algorithms. Although the individual models are disjointed and sometimes even mutually contradictory, the employment of common research techniques is intended to lead to a rational attempt to locate integrating principles among the various theories. This latter endeavor, while not formulated as an explicit study, is an underlying research objective.

The principal achievement of this research is the development of fundamental insights into information processing principles. In particular, continuing study of the relationship between the phrase-structure grammars developed by linguists and the procedure-oriented programming languages is important not only to a comprehension of the general structure of language, but to the improvement of ways in which languages of all kinds may be developed and processed. Additionally, the growing understanding of several models of data processing in terms of formal logic, in particular a model of question asking, shows promise of providing substantial tools for integrating a variety of information processing problems in a form susceptible to analysis.

The second category of studies--in the area of man-machine partnership--is evolving in a significant manner. In previous years these studies were a disparate collection whose principal connection was the fact that they were all concerned with "artificial intelligence." During this past year the Information Processing Research staff, together with representatives of other staffs, conducted a study into the meaning and utility of artificial intelligence research both in the general research community and in the particular context of SDC's mission. The study concluded that "pure" artificial intelligence research was not of central interest to SDC's mission, and, indeed, was becoming a less attractive research area in general. However, exploration of the possibilities of man-machine interaction with the objective of augmenting man's intellect was not only of key importance to SDC, but also held promise of great general value and interest.

As a result of this conclusion, together with the recognition that the skills and techniques required are similar to those previously being employed in the artificial intelligence area, a gradual redirection of this research is underway. The change in title of the "Research in Adaptive Programming" project to "Problem Solving and Learning by Man-Machine Teams" reflects this movement. A new study, "Augmented Statistician," is an example of a direct attack in the new direction. As time goes on, this new area will come to characterize this part of the staff's work.

During 1966 several projects previously under investigation were terminated (see Completed Studies, p. 2-14). One significant development has been the continuing withdrawal of SDC involvement in several projects where the principal investigator has been a consultant to SDC. In these cases, termination of SDC's participation has not meant a cessation of the research but merely a shift in the institutional arrangements for the study.

In summary, during 1966 the Information Processing Research staff has reoriented its activities somewhat toward an emphasis on problems more central to SDC's mission. It has also worked toward bridging the gap between research and application, both by better communication and by research emphasis. One of its chief aims continues to be the development of a science of information processing.

Note: The work of several of the following members of, or consultants to, the Information Processing Research staff is described elsewhere in this report:

S. Y. Sedelow, T. L. Ruggles - Stylistic Analysis
(see under Language Processing & Retrieval - p. 5-17)

D. P. Haggerty - Translation Between Procedure-Oriented Languages

W. E. Meyer - PL/I for SDC 360 TSS
(see under Advanced Programming - p. 1-1 et seq.)

FORMAL MODELS OF INFORMATION PROCESSING

Theory of Algorithmic Languages*

J. Doner
S. Ginsburg
T. N. Hibbard
G. F. Rose

Consultants: S. A. Greibach, Harvard University; M. A. Harrison, University of California, Berkeley; E. H. Spanier, University of California, Berkeley; J. S. Ullian, Washington University

Description

A serious drawback in the application of modern data processing systems is the cost and time consumed in programming these complexes. The user's problems and their solutions are described in a language such as English. To use the services of a data processor, this descriptive language must be converted into machine language; that is, into program steps. In recent years, attempts have arisen to bridge gaps by constructing programming languages that are:

1. Rich enough to allow a description of the solution of a wide range of problems.
2. Reasonably close to the user's ordinary language of description and solution.
3. Formal enough to permit a mechanical translation into machine language.

The purpose of this investigation is to accomplish the following:

1. Conduct research designed to develop a theory for algorithmic (programming) languages.
2. Develop suitable mathematical models of currently used mathematical languages such as ALCOL, COBOL, and JOVIAL.
3. Use the mathematical models to answer questions of interest about these languages.

*Supported in part by the Air Force Cambridge Research Laboratories, Office of Aerospace Research, and the Air Force Office of Scientific Research, Office of Aerospace Research.

Progress

Computer algorithmic languages are formal languages; that is, they consist of a formal syntax for deriving the meaningful units of expression such as words, clauses, sentences, arithmetic expressions, etc. Several grammatical systems for deriving the syntax of formal languages are in the literature. These give rise to various families of formal languages such as the recursively enumerable sets, context sensitive languages, context free languages, and regular sets. Formal languages are also defined by special kinds of acceptors such as the finite-state acceptors and the pushdown acceptors. Both grammatical and acceptor-based methods of defining formal algorithmic languages have been investigated as have the languages obtained by these methods.

Of all the models used to consider programming languages, the most universally accepted one is that of the context free language (i.e., a language defined by Backus normal form). Four of the five technical reports written during 1966 concern this model. In [1] context free grammars are considered in which indexed brackets are inserted around the right-hand side of the rules in the grammar. The resulting language, called "bracketed," appears to be a natural component in the theory of transformational grammars, a topic of concern in natural languages. In the report, an algebraic condition is given for one bracketed language to be a subset of another. It is also shown that the intersection and difference of two bracketed languages with the same brackets and terminals is a context free language. Report [2] concerns a special family of context free languages that arose from mathematical considerations. In it, two characterizations of bounded regular sets are given. In addition, certain connections with items of mathematical interest are noted. In [4], partial algorithms for context free grammars are considered. The partial algorithms considered here are of the following form: "Suppose a certain problem is known to be recursively unsolvable, but in a particular case is known to have a solution. Can an algorithm (called a partial algorithm) be found to determine the solution?" Among the results obtained are the following: There is no partial algorithm for finding, given context free grammars G_1 and G_2, a generalized sequential machine (complete sequential machine) which maps the language generated by G_1 onto the language generated by G_2. In [5], a new device is given which recognizes exactly the context free languages. (This device, unlike a pushdown acceptor, is a special kind of linear bounded acceptor.) The languages recognized by the deterministic form of the device result in a larger class of unambiguous languages than that given by deterministic pushdown acceptors.

Report [3] concerns itself with a device that is more powerful than a pushdown acceptor. This device, called a one-way stack automaton, has the same features as the pushdown acceptor with the additional feature that it can go into its push-down store and read, but not write. It is thus more realistic in modeling current compilers. The family of languages accepted by these acceptors is then studied. In particular, various closure properties and solvability questions are considered.

Plans

Future work is expected to extend the research to more realistic acceptors and the languages accepted by these devices. Studies are also under way to find new kinds of grammars and new ways of using old grammatical rules.

Project Documentation

1. Ginsburg, S. and Harrison, M. Bracketed context free languages. SDC document TM-738/023/00. January 4, 1966. 35 pp. (To appear in the Journal of Computer and System Sciences.)
2. Ginsburg, S. and Spanier, E. H. Bounded regular sets. Proceedings of the American Mathematical Society, 1966, 17, 1043-1049.
3. Ginsburg, S., Greibach, S., and Harrison, M. One-way stack automata. SDC document TM-738/025/00. April 22, 1966. 58 pp. (To appear in the Journal of the ACM.)
4. Ullian, J. Partial algorithm problems for context free languages. SDC document TM-738/027/00. October 10, 1966. 34 pp.
5. Hibbard, T. N. A generalization of context free determinism. SDC document TM-738/028/00. November 21, 1966. 67 pp.
6. Ginsburg, S. and Ullian, J. Ambiguity in context free languages. Journal of the ACM, 1966, 13, 62-89.
7. Ginsburg, S. and Spanier, E. H. Semigroups, Presburger formulas, and languages. Pacific Journal of Mathematics, 1966, 16, 285-296.
8. Ginsburg, S. and Rose, G. F. A characterization of machine mappings. Canadian Journal of Mathematics, 1966, 18, 381-388.
9. Ginsburg, S. and Rose, G. F. Preservation of languages by transducers. Information and Control, 1966, 9, 153-176.
10. Ginsburg S. and Ullian, J. Preservation of unambiguity and inherent ambiguity in context free languages. Journal of the ACM, 1966, 13, 364-368.
11. Ullian, J. Failure of a conjecture about context free languages. Information and Control, 1966, 9, 61-65.
12. Ullian, J. S. and Hibbard, T. N. The independence of inherent ambiguity from complementedness among context free languages. Journal of the ACM, 1966, 13, 588-593. (Also available as SDC document TM-738/010/00.)
13. Greibach, S. A. The unsolvability of the recognition of linear context free languages. Journal of the ACM, 1966, 13, 582-587. (Also available as SDC document TM-738/015/00.)
14. Ginsburg, S. and Spanier, E. H. Finite-turn pushdown automata. Journal of SIAM on Control, 1966, 4, 429-453. (Also available as SDC document TM-738/020/00.)

The Logic of Questions

T. B. Steel, Jr.

N. D. Belnap, Jr., Consultant at SDC
(University of Pittsburgh)

Description

A large body of literature concerning the logic of inference is in existence. The logic of inference has a considerable impact on the information processing sciences, even though it is essentially a logic of declarative sentences.

Men rarely assert things to machines; rather, they command or question them. Clear desiderata, then, are a logic of imperatives and a logic of interrogatives. Neither exists.

Initial focus is upon the logic of questions on the grounds that it appears easier to formulate. The first objective of the study is to formulate suitable criteria for deciding what kind of formal system will be acceptable as a logic of questions. An informal example of such a criterion is: it must be effectively decidable about any piece of language whether it is a question or not and, if it is, it must be effectively decidable what pieces of language count as answers (although not necessarily true answers).

Given a satisfactory set of criteria, the next task is to develop one or more specific examples of a formal logic of questions in order to search for fruitful theorems. Apart from its intrinsic interest such a logic is imperative as a basis for the design of processors for true problem-oriented languages. In such languages, the user will simply describe a problem for machine solution, in contrast to the present practice of providing a technique for solution via a procedure-oriented language.

Progress

A set of criteria has been developed and a rather general formal system has been analyzed subject to the criteria. Notions allied to what is loosely meant in ordinary language by "question," "direct answer," "complete answer," "partial answer," and even "rhetorical question" have been defined. The fallacy of many questions ("Have you stopped beating your wife?") is disposed of neatly when subjected to this analysis. As a result of these analyses, the criteria have been modified to sharpen their impact.

The key element in this approach that distinguishes it from previous attempts at an interrogative logic is that a question is treated as identical with a declarative sentence on both the syntactic and semantic levels of analysis. It is recognized as a question only on the pragmatic level, by observing that the user intends the statement as a question.

A detailed document elaborating the theory has been published [1]. Under revision is a paper intended to outline the connection of this logic with data processing applications. In particular, the paper discusses the reduction of English questions to the necessary formalism. This paper includes a correlation of the work in this project with the Synthex project (see p. 5-7) and other query systems. In addition, it elaborates and modifies the quantification analysis found in [1] on the basis of study completed during the year and establishes several new notions pertinent to the logic of questions.

Project Documentation

1. Belnap, N. D., Jr. An analysis of questions: Preliminary report. SDC document TM-1287. June 3, 1963. 160 pp.

AUGMENTATION OF MAN'S INTELLECT

Problem Solving and Learning by Man-Machine Teams*

A. M. Hormann, Principal Investigator
T. L. Ruggles
S. S. Shaffer

Description

This study seeks to develop a system of computer programs that can exhibit some "adaptive" and "intelligent" behavior in a variety of problem-solving situations, and thus to develop a machine capable of playing the role of "partner" to man in his intellectual/creative endeavor. The first stage of the research has been concerned with how to develop an adaptive, intelligent, problem-solving system--Gaku [1 and 2]. The second stage has been to design a task environment that can be

*Supported in part by the Office of Naval Research, U. S. Navy.

controlled by the experimenter as he presents problem situations differing in kind and complexity [3 and 4]. In this environment, a human will attempt to solve a given problem by interacting with Gaku, using its capabilities in various aspects of problem solving and learning.

One of the reasons for stressing the "learning" capability of Gaku is the researcher's belief that an intelligent, adaptive machine will be an appropriate partner for man in those problem situations in which the man does not have a clear idea of, or complete information about, how to solve a problem, how to find answers, or how to perform a task. In such situations, detailed decision making in advance is impossible or infeasible. Man may start with incomplete information and vague ideas about solution methods and strategies (and, therefore, insights can be gained only during the course of problem solving and interaction with the machine). The human user cannot decide in advance exactly and completely what machine capabilities he will need, or what techniques, methods, concepts, and terms he will use. He must decide as he proceeds and must teach the machine, thus allowing it to "grow" intellectually with him.

Although the development of Gaku is far from the desired sophistication level indicated above, it has been decided to use the system in a man-machine context in order to gain new insights into certain research problems. By this means, Gaku's limitations and capabilities will be discovered, and ways to improve the learning mechanisms and the communication means will be explored. It may then be decided to make either major design changes or relatively simple modifications and additions.

Progress

The design of a task environment for man-machine interactive problem solving has been completed and its implementation on SDC's S/360 Time-Sharing System has begun. A brief description of the environment called Shimoku follows; it includes little discussion on Gaku since Gaku has been described in last year's annual report, and in [1] and [2].

The basic design of Shimoku is a four-in-a-row game in which marked counters are used in a 4x4, 4x4x4, or 4x4x4x4 board or playing surface (the size of the board is one of the factors determining complexity). Four-in-a-row positions are determined in the same manner as are those of 2-D, 3-D, and 4-D tic-tac-toe (see Figures 2-1 and 2-3). The counters are marked with numbers and suits, similar to a set of playing cards (in Shimoku, however, four shapes of counters are used instead of suits). The scoring rules include elements that are similar to poker (see Figure 2-2), and the action rules consist of "placing," "sliding," and "exchanging" counters on the playing surface of the given board.

In [3], a detailed description of Shimoku is given and some sample problem situations are discussed to show how a human player might exploit the capabilities of Gaku to aid him in man-machine interactive problem solving. In this environment, the man faces conflicting subgoals while attempting to reach a main goal; he must weigh the consequences of different courses of action, including compromises and local sacrifices, for the purpose of overall gain; he must make decisions in the face of incomplete and inexact information; and, most important, he must formulate policies or strategies for long-range planning. In all these, the player can be assisted by Gaku, to whatever degree his own cleverness and Gaku's development permit.

In [4], one variation of Shimoku, interpreted as an assignment task (one type of operations research problem), is described. Within this task context, a player starts with an initial capital and a given configuration of counters on the Shimoku board, and works through three contractual periods, each of which has a different

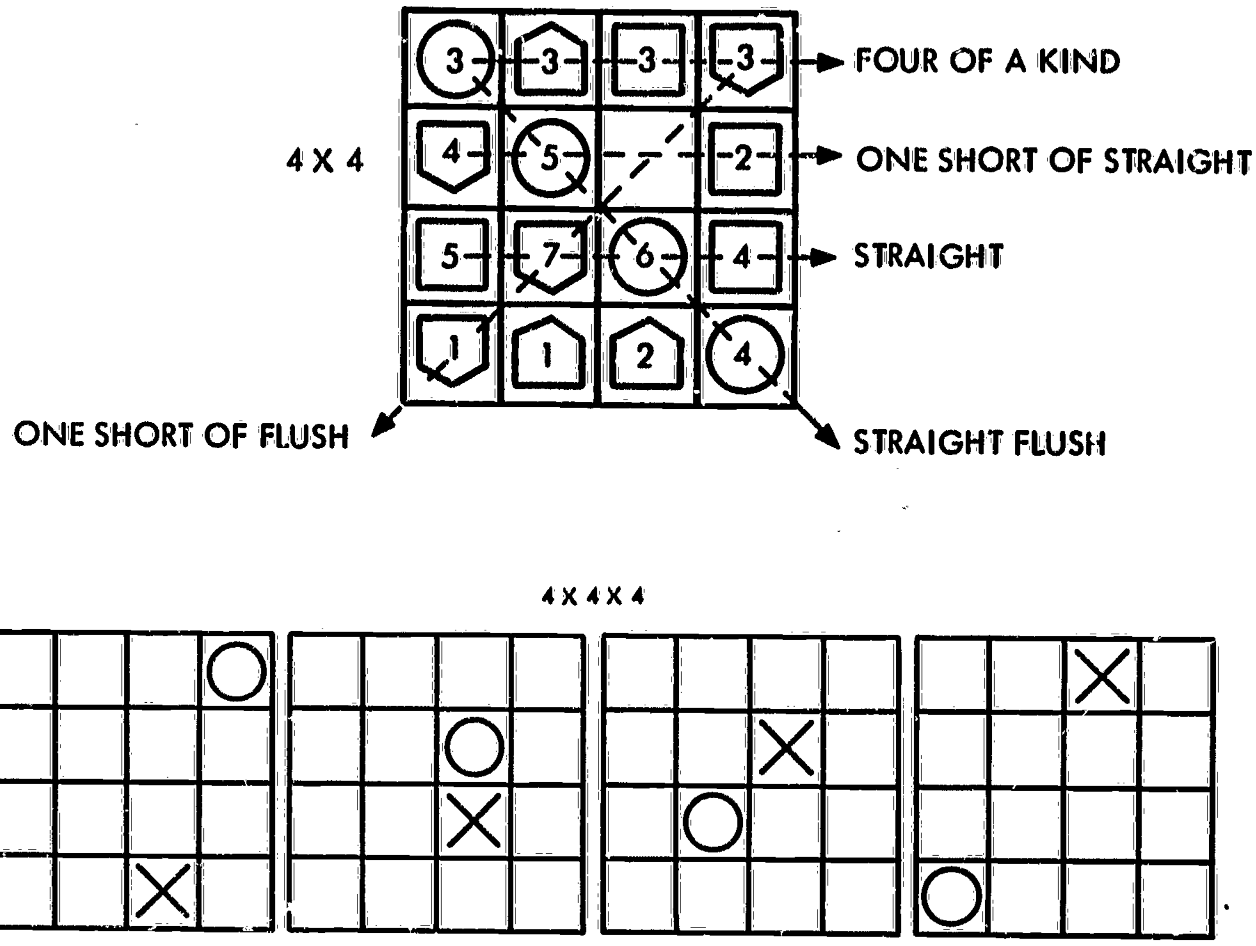


FIGURE 2-1. 2D AND 3D SHIMOKU

PATTERNS	EXAMPLES				SCORE
STRAIGHT FLUSH (SF)	5	6	7	8	10 POINTS
FOUR OF A KIND (4K)	9	9	9	9	8 POINTS
FLUSH (FL)	5	1	8	3	6 POINTS
STRAIGHT (ST)	3	4	5	6	4 POINTS

FIGURE 2-2. FOUR PATTERNS FOR SCORING IN SHIMOKU

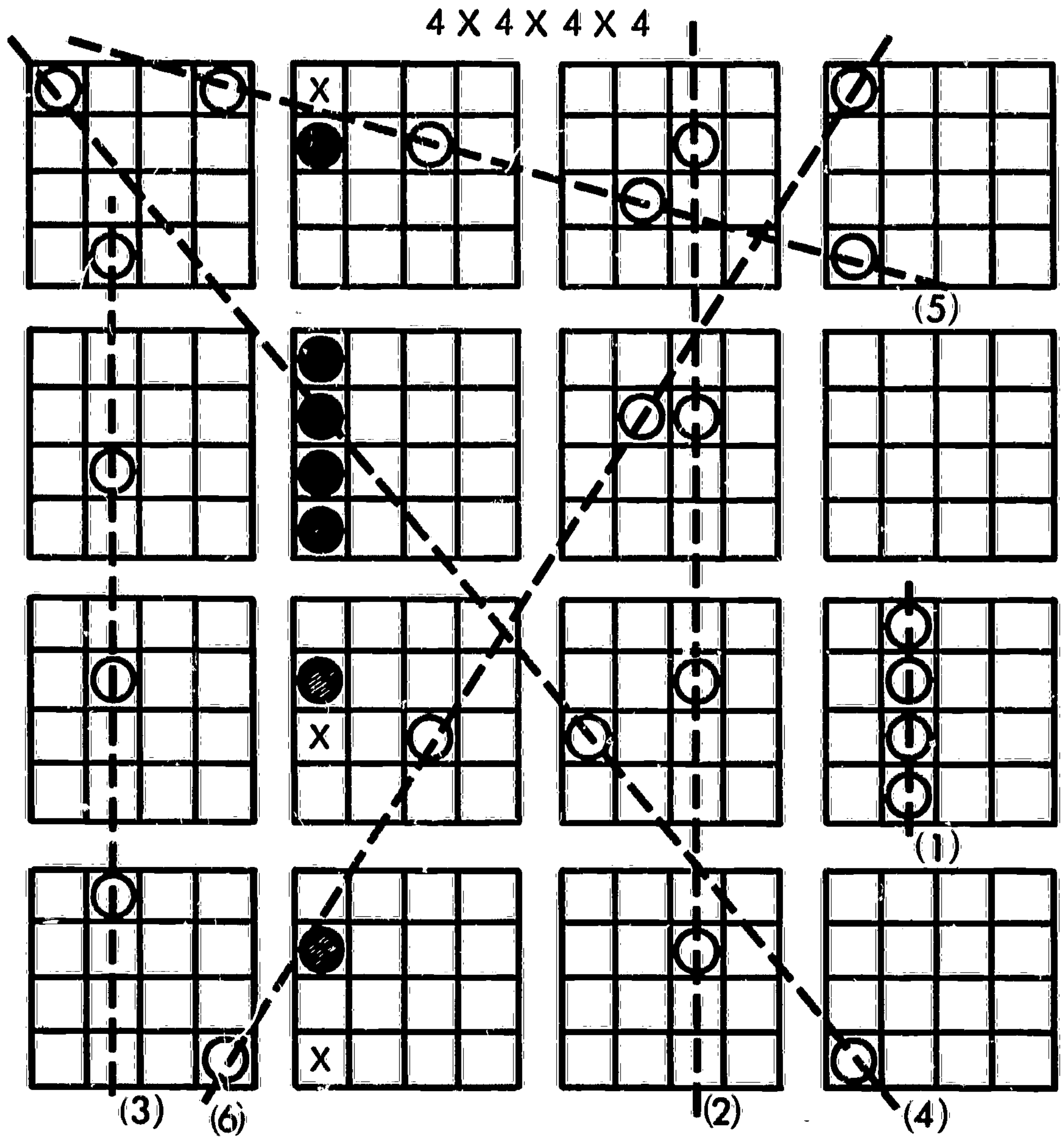


FIGURE 2-3. 4D SHIMOKU

set of contractual demands for certain products. The player is also given a set of action rules, determining how the counters (resources) may be bought and manipulated (assignment actions), and stipulating their corresponding costs. His assigning performance in each period is evaluated in terms of payoff value--depending on how many contractual demands are fulfilled. This value is then added to whatever is left of the capital for that period; the new capital and configuration at the end of the period then become the starting conditions of the next period with a new set of contractual demands. The objective is to score (in terms of the amount of capital) as high as possible at the end of the third contractual period.

The initial effort in implementation of Shimoku has been concentrated on the visual display of Shimoku configuration, using an IBM Model 2250 display unit and a Graphic Input Tablet, in order to provide near-"natural" interaction between a human player and the environment.

Plans

Implementation of the Shimoku environment must be carefully planned, as many variations are to be incorporated in one package of programs. The experimenter must be able to present problems of different types and complexity. Preliminary experiments with a human player will then begin, first independently of Gaku. Gaku will have to be newly implemented since the work done so far has led to many design changes and added features, some of which could not have been incorporated using the old facility (the Philco 2000). The implementation, however, will not be started until a suitable programming language becomes available for the 360 in its time-sharing mode (JOVIAL is being used for Shimoku, but a list processing language such as LISP 2 will be desirable for Gaku). In the meantime, communication between the human problem solver and his environment (eventually including Gaku) will be studied along with the display techniques for effective interaction. A flexible and "natural" communication means is needed to help the human problem solver move smoothly from idea-getting (conceptual stage) to trying-out (experimentation stage) to result-judging (evaluation stage)--and then usually back to the first stage again. The user's effectiveness as a problem solver can be influenced greatly by the nature of communication allowed; his own learning, in turn, will influence Gaku's secondary learning, which can only come about by communication.

Project Documentation

1. Hormann, A. M. Gaku: An artificial student. Behavioral Science, 1965, 10(1), 88-107.
2. Hormann, A. M. Designing a machine partner--prospects and problems. SDC document SP-2169/000/01. October 15, 1965. 55 pp.
3. Hormann, A. M. A new task environment for Gaku teamed with a man. SDC document TM-2311/003/00. May 27, 1966. 26 pp.
4. Hormann, A. M. Shimoku posed as an assignment task. SDC document TM-2311/004/00. September 28, 1966. 13 pp.

Augmented Statistician

R. E. Dear
R. L. McCornack
W. E. Meyer
T. B. Steel, Jr.
E. A. Stefferud

Consultants: N. S. D. DuBois, University of California, Los Angeles; H. Solomon, Stanford University

Description

The development of modern statistical theory began roughly at the beginning of this century. Its products have had an impact on measurement in diverse disciplines ranging from anthropology to zoology. Statistics has aided measurement through its various methodologies, each of which is designed to explore the total variation in a system that has random elements or a combination of random and fixed elements; and through its concepts for estimations, it has been used for tests of hypotheses, and for more generalized decision-making structures. To make the latter (estimations aspect) more concrete, we may list the notions of confidence intervals, power functions, and risk functions which include a number of cost elements; for the former (measurement aspect) we may note the analysis of variance, regression analysis, correlation analysis (all of which are interrelated), sample survey design, acceptance sampling, sequential analysis and optimal stopping strategies, and time series analysis and spectral analysis.

Much effort on statistical issues by a large number of scholars has found its way into the literature. This work now fills many journals and many volumes, providing a reservoir of research techniques for the investigator. The concepts and methodologies were initiated and developed without any aid from the digital computer. Advances in computer hardware and software have been swift, and this has led to inquiries by staff members in SDC's Research Directorate about the desirability of dynamic interaction between a statistician and a computer and the possible consequences for research in all intellectual activities. The issues involved are, at best, amorphous at present, but enough work has been devoted to the topic to indicate that "the Augmented Statistician" is worth experimental study.

Progress

A feasibility study and an analysis of the need for such a system have been undertaken. The conclusions reached are: (1) an Augmented Statistician is technically feasible, (2) it will meet an existing need, and (3) its cost/effectiveness cannot be determined without building a prototype.

The initial purpose of the Augmented Statistician will be to provide the professional statistician and the qualified user of statistical techniques a dynamic, on-line, display-oriented capability for data analysis.

Data analysis was chosen as the area for initial emphasis for three reasons. First, the arguments of several statisticians (e.g., Tukey) for the use of on-line data processing in data analysis, plus the SDC experience with the TRACE system (see p. 6-16), indicate a higher potential utility for a data analysis system than for other possible application areas. Second, much of the programming and system design required for the data analysis capability is prerequisite to other more specialized areas such as time-series analysis, geometric probability and clustering techniques. Each of these is a candidate for the next step. Finally, the opportunities for innovation in information processing technology seem largest in this area. The feedback requirements and language design problems do not appear as obvious in this application as for some others.

At the beginning three major program components are required: (1) a control and sequencing program, (2) a package of computational subroutines, suitably selected, and (3) a general display capability. The control program will require the greatest ingenuity. Much of the subroutine package can be borrowed from existing libraries with little modification beyond that required to insure data set compatibility. New display equipment in SDC's Research & Technology Laboratory will provide the necessary display facility. Compatibility with TDMS (see p. 4-9) will allow it to be coupled with other existing tools for data handling.

In order to insure usability of the system, certain constraints need to be applied to the programming technology employed. By using SDC's IBM S/360 Time-Sharing System, and at the same time insisting on compatibility with the IBM operating system, the potential body of users is large. Similar remarks apply to the programming languages used.

There should be no question that an Augmented Statistician can be built along the lines indicated.

The question that cannot be answered without further work is whether a real quantum jump in effectiveness can be attained.

Plans

The next objective of the project is to design and implement a prototype system, concentrating on data analysis. This will be done within the framework of the SDC 360 Time-Sharing System. Given this prototype system, an evaluation with users having real data and real problems will be undertaken to provide guidelines for future work.

Project Documentation

1. DuBois, N. S. D., Jr. An interactive statistical system. SDC document TM-3043. July 1, 1966. 18 pp.
2. Steel, T. B., Jr. The augmented statistician. SDC document TM(L)-3201/000/01. November 11, 1966. 17 pp.

Synthesis of Behavior

L. Friedman, Principal Investigator
S. S. Shaffer

Description

The primary goal of this investigation is the formulation of a theory of instinctive brain mechanisms. Information processing in brains has so far resisted analysis. In order to correlate neurophysiological structure with behavioral output, a theory of functional operation of brain processes has been formulated, based on experimental observations of ethology and neurophysiology. To validate the operation of the proposed mechanisms, a simulated robot, which embodies the postulated mechanisms and organization in its simulated "brain," is being programmed.

Three basic groups of mechanisms are assumed. These are the Behavior Unit (BU), the Releasing Mechanism (RM), and the Selector of Releasing Mechanism (SRM). The BU is a functional unit which produces behavior directed toward the accomplishment of a single goal, set at some arbitrary level. The organism must be provided with a set of BUs that can express all possible behavior by being activated either succesively or in concert. An Executive Control operates by activating the BUs just as a piano player produces a melody by striking keys. The Executive Control consists of hierarchies of RMs triggered by inputs from the environment and competing for activation by SRMs. The detailed structure of this complex organization has been worked out [1,2,3].

The simulated robot, called ADROIT, is programmed to be displayed on a computer-controlled cathode ray tube. The experimenter activates the actions of the robot with a stylus operating on a Graphic Input Tablet (Figures 2-4 through 2-6).

Progress

All the BUs for the simulation have been programmed and were operational on the Philco 2000 before that computer was replaced in September 1966. These BUs are:

SEEK designated type of object

GO TO object (or some designated place)

MOVE ARM TO object

GRASP object

RELEASE object

A decision making strategy was incorporated in the "GO TO" BU to enable the robot to attain the desired position despite arbitrarily placed obstacles which it had to avoid en route. A film showing operation of the BUs was made on February 26.

In addition, an early version of the Executive Control has been implemented. A complex pattern of behavior units, demonstrating a nest-building capability, was actually produced by the robot. During this demonstration, the Executive Control showed one of the basic characteristics of animal behavior: The robot exhibited a sensitivity to unexpected environmental changes* while it was accomplishing its objectives. Thus, if a new desirable object for the nest were placed directly

*As distinct from features randomly placed in the field, but not changing during the run.

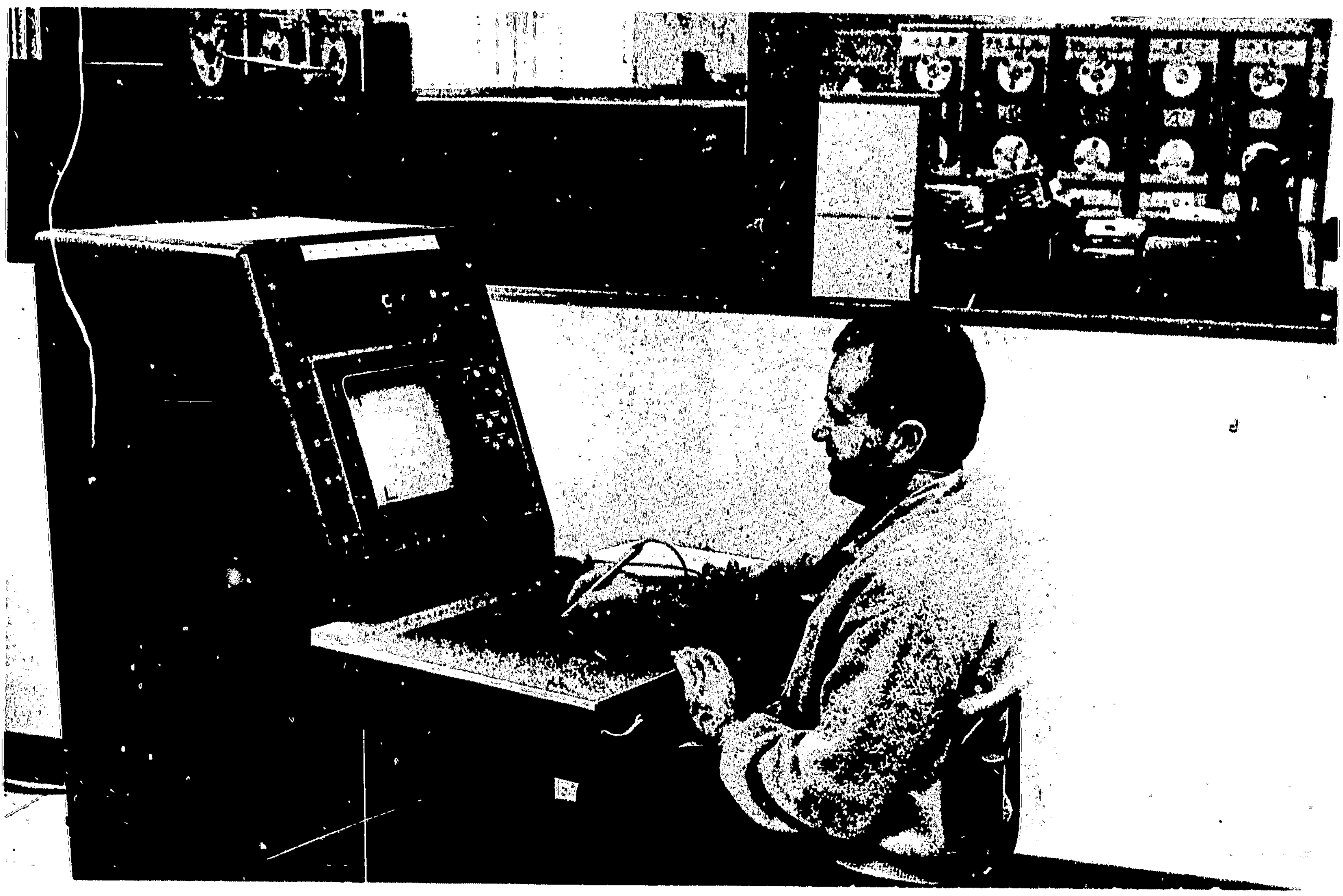


FIGURE 2-4. EXPERIMENTER USING DISPLAY SCOPE AND GRAPHIC INPUT TABLET TO ISSUE COMMANDS TO ROBOT (PHILCO 2000 COMPUTER IN BACKGROUND)

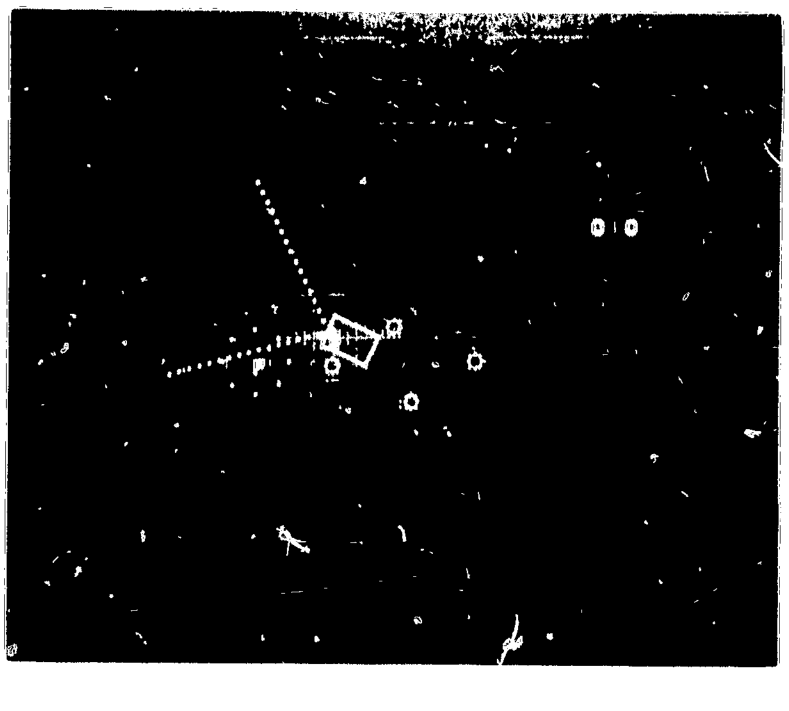


FIGURE 2-5. SCOPE DISPLAY OF ROBOT MAKING ITS WAY THROUGH OBSTACLES IN RESPONSE TO "GOTO" COMMAND

FIGURE 2-6. CLOSE-UP DISPLAY OF ROBOT RESPONDING TO "MOVE ARM TO" COMMAND

in its path, the robot promptly grasped it, abandoning the attempt to get to a more distant object. If the experimenter introduced a building object in a correct position in the growing structure of the nest, while the robot was away gathering another element of the nest, the robot would, upon its return, place the newly acquired element correctly next to, rather than on top of, the one introduced by the experimenter. Such a pattern was filmed on September 28.

During the course of this work, close contact has been maintained both with researchers dealing experimentally with brain function and structures, and with groups engaged in the construction of hardware robots.

Plans

Work will continue on elaborating performance of the Executive Control to permit the robot to display various phenomena of instinctive behavior dealing with conflict situations, such as displacement, oscillation, alternation, and suppression of conflicting drives. In addition, higher-level decision making mechanisms will be investigated.

Project Documentation

1. Friedman, L. A model of goal-directed behavior. Artificial Intelligence, Publication S-142, Winter General Meeting IEEE, January 1963, 66-82.

2. Friedman, L. Theories of instinctive behavior and their computer analogues. SDC document SP-2292. December 10, 1965. 42 pp.

3. Friedman, L. Instinctive behavior and its computer synthesis. Behavioral Science, in press.

Steps Toward Validating a Computer Model of Social Influence*

J. T. Gullahorn and J. E. Gullahorn — Principal Investigators
Consultants at SDC
(Michigan State University)
S. S. Shaffer

Student Associate: M. Useem, Harvard University

Description

Previous steps toward validating portions of HOMUNCULUS, a computer model of social behavior, have concerned relatively static sections of the model in which decisions were made on the basis of given values. During 1966 emphasis was shifted to validating parts of the model enabling simulated persons to modify their own behavior and to exert social influence intended to increase their rewards from face-to-face interaction.

Progress

The current research involves a computer simulation of Harold Gerard's small-group experiment on the anchorage of opinions. In Gerard's study subjects first read a case involving a union-management dispute and predicted the outcome on a 7-point scale. Gerard then selected subjects for three-man discussion groups, varying the degree of agreement on the predictions. Cohesiveness was also manipulated systematically through instructions to the subjects. In the first simulation runs of this experiment, independent variables were level of liking, amount of disagreement among opinions, and degree of confidence of each simulated person in his opinion.

When Gerard assigned groups to the "mildly disagree" or "strongly disagree" conditions, he considered only the average differences among

*This project received its early support from SDC's Independent Research program. In 196‹ it was supported in part by SDC, where the principal investigators were Visiting Scientists, and in part by a National Science Foundation Senior Postdoctoral Fellowship to John T. Gullahorn and a National Science Foundation research grant to Jeanne E. Gullahorn.

opinions and did not consider relative positions along the 7-point continuum. Since data from other studies indicate a positive relationship between extremity of opinion and confidence, and since high confidence would make subjects less amenable to group influence (in terms of one of the propositions operating in our simulation model), we decided to conduct a series of sensitivity runs to see whether different distributions of initial opinions would produce different outcomes of the simulated group interaction. In general, these sensitivity runs demonstrate that initial opinion distribution is more than a minor noise variable. Indeed, Gerard's findings concerning opinion change can be reproduced by holding all parts of our model constant and varying only the initial opinion distribution. For example, in runs simulating discussion by subjects in the "strongly disagree--high attraction" condition, an average of 26 percent of the simulated subjects changed toward another in their groups when initial opinion positions on the 7-point continuum were 2, 4, and 6. With initial positions 1, 4, and 5, however, 43 percent changed--a result matching the 44 percent change reported by Gerard.

A small-group experiment, a variation of Gerard's "The Anchorage of Opinions in Face-to-Face Groups," was designed and pretested during the summer of 1966. While incorporating Gerard's stimulus situation and general design, we have added several categories to classify activities (helpful if social exchange theory is to explain influence) and a questionnaire designed to measure the importance to the subject of maintaining consonance and integrity.

Plans

Work on HOMUNCULUS is continuing in five areas: (1) refining the analysis of approaches to validation of computer simulation models; (2) testing additional hypotheses in simulation runs and further validating the role conflict model; (3) refining and replicating the "anchorage of opinion" experiments with small groups; (4) refining and validating the computer simulation model of the "anchorage of opinion" experiments; and (5) documenting the relationship between the computer models and specific assumptions and hypotheses from social psychological theory. In addition, the computer simulation model of the "anchorage of opinion" experiment is serving as the core of a graduate seminar in computer simulation of social behavior in the Department of Sociology, Michigan State University.

COMPLETED STUDIES

The following studies conducted by the Information Processing Research staff were completed prior to 1966 and are not described in this report.

Theory of Programming Languages

1. Ginsburg, S. Sets of tapes accepted by different types of automata. Journal of the ACM, 1961, 8(1), 81-86.
2. Ginsburg, S. and Rice, H. G. Two families of languages related to ALGOL. Journal of the ACM, 1962, 9, 350-371.
3. Ginsburg, S. and Rose, G. F. Some recursively unsolvable problems in ALGOL-like languages. Journal of the ACM, 1963, 10, 29-47.
4. Ginsburg, S. and Rose, G. F. Operations which preserve definability in languages. Journal of the ACM, 1963, 10, 175-195.
5. Ginsburg, S. and Spanier, E. H. Quotients of context free languages. Journal of the ACM, 1963, 10, 487-492.
6. Ginsburg, S. and Hibbard, T. N. Solvability of machine mappings of regular sets to regular sets. Journal of the ACM, 1964, 11, 302-312.
7. Rose, G. F. An extension of ALGOL-like languages. Communications of the ACM, 1964, 7, 52-61.
8. Ginsburg, S. and Spanier, E. H. Bounded ALGOL-like languages. Transactions of the American Mathematical Society, 1964, 113, 333-368.
9. Ginsburg, S., Hibbard, T. N., and Ullian, J. S. Sequences in context free languages. Illinois Journal of Mathematics, 1965, 9, 321-337.
10. Ginsburg, S. and Spanier, E. H. Mappings of languages by two-tape devices. Journal of the ACM, 1965, 12, 423-434.

11. Ginsburg, S. and Greibach, S. A. Deterministic context free languages. SDC document TM-738/014/00. May 7, 1965. 43 pp.

12. Greibach, S. A. A note on pushdown store automata and regular systems. SDC document TM-738/016/00. August 30, 1965. 13 pp.

13. Ginsburg, S. and Greibach, S. A. Mappings which preserve context sensitive languages. SDC document TM-738/018/00. October 27, 1965. 29 pp.

14. Ullman, J. Pushdown automata with bounded backtrack. SDC document TM-738/022/00. December 3, 1965. 33 pp.

Theory of Programmed Processes

1. Hibbard, T. N. A simple sorting algorithm. Journal of the ACM, 1963, 10(2), 142-150.

2. Hibbard, T. N. An empirical study of minimal storage sorting. Communications of the ACM, 1963, 6(5), 206-213.

3. Stockmal, F. J. On the inverse of a test matrix. Communications of the ACM, 1965, 7(10), 615.

4. Stockmal, F. J. Calculations with pseudo-random numbers. Journal of the ACM, 1964, 11, 1-12.

UNCOL: Universal Computer Oriented Language

1. Steel, T. B., Jr. A first version of UNCOL. Proceedings of the Western Joint Computer Conference, 1961, 19, 371-378.

2. Steel, T. B., Jr. UNCOL. Datamation, 1960, 1, 18-20.

3. Steel, T. B., Jr. UNCOL: The myth and the fact. In R. Goodman (Ed.), Annual Review in in Automatic Programming. 2. New York: Pergamon Press, 1961.

4. Steel, T. B., Jr., et al. The problem of programming communication with changing machines--a proposed solution. Communications of the ACM, 1958, 1(8), 12-18; 1958, 1(9), 9-15.

Pattern Recognition and Language Learning

1. Prather, R. C. and Uhr, L. M. Discovery and learning techniques for pattern recognition. Proceedings of the 19th National Meeting of the ACM, 1964, D2.2-1--D2.2-10. (Also available as SDC document SP-1561/000/01.)

2. Uhr, L. M. Pattern-string learning programs. Behavioral Science, 1964, 9(3), 258-292.

3. Uhr, L. M. Recognition of letters, pictures, and speech by a discovery and learning program. Symposium on Pattern Recognition, IEEE WESCON, Los Angeles, California, August 1964.

Binary-Choice Model

1. Feldman, J. and Hanna, J. F. The structure of responses to a sequence of binary events. SDC document SP-1664. May 22, 1964. 20 pp.

Stimulus-Sampling Models

1. Marzocco, F. N. Simulation of learning processes. Proceedings of the Fifth National Symposium on Human Factors in Electronics, 1964, New York: IEEE, 77-80. (Also available as SDC document SP-1597.)

2. Marzocco, F. N. Stimulus sampling pattern recognition. Bionics Symposium, 1963, Contributed Paper Pre-Prints. Wright-Patterson Air Force Base, Ohio: Air Force Systems Command, 1963. (Also available as SDC document SP-986/000/01.)

3. Marzocco, F. N. Computer recognition of hand-written first names. Preprint No. 8.2, 1964 Western Electronic Show and Convention, Los Angeles, August 1964.

Theorem Utilizing

1. Travis, L. E. Experiments with a theorem-utilizing program. Proceedings, 1964 Spring Joint Computer Conference. Baltimore, Md.: Spartan Books, 1964, 339-358. (Also available as SDC document SP-1402/000/01.)

Sensorimotor Development

1. Greene, P. H. Superimposed random coding of stimulus-response connections. Mathematical Biophysics, 1965, 27, 191-202. (Also available as SDC document SP-2071.)

2. Greene, P. H. and Ruggles, T. L. CHILD and SPOCK. IEEE Transactions on Military Electronics, 1963, MIL-7(2-3), 156-159.

Theory of Information Handling

1. Steel, T. B., Jr. A formalization of semantics for programming language description. In T. B. Steel, Jr. (Ed.) Formal Language Description Languages for Computer Programming. Amsterdam: North-Holland, 1966. (Also available as SDC document TM-2043.)

PROGRAMMING SYSTEMS

C. Weissman, Head

The Programming Systems staff is responsible for the invention and synthesis of programming technology into integrated systems that extend man's use of computers and make him more productive in that use. In striving toward this objective, the staff has become involved in a number of design and development activities: some reflect the success of past products; others look forward toward future systems. In the former category, emphasis has been on the maintenance, study, and improvement of TSS, one of the first general-purpose time-sharing systems (as reported in previous years). This system--which is now available (and fully subscribed) on a limited subscription basis to the user community--has continued to prove its utility as an effective, on-line, interactive programming vehicle. The second category includes projects for developing programmer- and user-oriented languages and executive systems for tomorrow's information processing needs. Foremost among these activities are the development of META compiler and LISP 2 systems (described in the Advanced Programming Section of this report) and the experimental design and development of a network of time-shared computer systems.

An example of SDC's synthesis of past experience in the programming systems field with the future needs of the Air Force is the recently undertaken production of an advanced development prototype (ADP) time-sharing data management system for the Department of Defense. This work stems from an earlier SDC-designed system (called GENISYS) and will be performed in cooperation with the Data Base Systems staff.

Thus, the focus of the Programming Systems staff is upon time-sharing systems and networks as the vehicles that permit close user-computer coupling, and upon language systems that permit a user-oriented dialog to control the computer and effect more productive use of these machines.

In particular, efforts in the time-sharing area have been directed toward the development and refinement of new tools and techniques that improve management and user control of system resources and that yield improved reliability, quality, and responsiveness of the services offered by the system. Since its inception in 1963, TSS has been unique as the only large-scale time-sharing system to service a large and varied remote community of users. In support of non-SDC and in-house SDC users, the staff has resolved problems of scheduling user programs that are often large (25K words average) and require extensive computing time. Also, they have improved large-file management and editing tools, improved debugging capabilities, increased disc-file executive commands, and provided on-line system documentation. Some of the changes made are not obvious to users. These include adding the ability to load system components as object programs for on-line modification and debugging under time-sharing; reassignment of disc inventory programs and tables to drum for better system responsiveness; and redesign of the PDP-1 executive program to increase the number and type of console channels handled by the system. The channel capacity is now up to 53.

These changes could not have been possible without an intensive analysis of the behavior of the operating system, and the needs of the user community. The time-sharing analysis efforts of prior years have continued; they have been an excellent source of statistical information on computer facility job descriptions and user characteristics. Both simulation and empirical data collection studies were used in the analysis.

Another method of measuring the effectiveness of the time-sharing system involved experimentation and testing of user performance. An experimental comparison of on-line versus off-line programmer debugging performance was concluded, analyzed, and documented during this period. Though the results of this experiment may be interpreted statistically as favoring time-sharing, possibly more importantly, the study highlighted the difficulties of controlling such an experiment because of the enormous range of individual performance differences. However, the experiment was a meaningful first step in attempting to quantify measures in an area previously supported only by subjective, qualitative testimonials.

Finally, an understanding of the needs of users was gathered through feedback from the more than 500 authorized users of the system, plus reactions by the large number of visitors to briefings and demonstrations of TSS and its programs.

Looking toward the future, the staff has participated in two efforts to design a network of time-sharing systems. The first effort, performed in conjunction with the University of California, Berkeley; Bolt, Beranek and Newman, Inc.; and Stanford Research Institute, resulted in a design for a network of SDS 940 computers. The design specifications, edited and published by Wayne Lichtenberger of UCB, are now being circulated in the industry for comment.

The second effort has reached fruition as an operating system. Less sophisticated than the proposed 940 network, this system couples the TX-2 computer at Lincoln Laboratory in Boston to the Q-32 computer at SDC/Santa Monica by commercial dial-up telephone lines. This austere first step in networking is an encouraging example of technological synthesis. Programs on either machine may "log in" and operate programs on the other machine. In contrast with the 940 network of like machines, the TX-2/Q-32 network uses dissimilar machines <u>and</u> monitors, and yet provides a medium whereby the running programs of one researcher may be used directly on his machine by others not using his equipment. (This method could prove to be the only way for the data processing community to truly share the programs and the work of others.)

TIME-SHARING SYSTEM DEVELOPMENT*

S. M. Aranda, Project Leader
L. E. Durham
C. E. Fox
R. R. Linde
E. P. Myer
D. E. Perry

Description

Since the Time-Sharing System (TSS) became operational in June 1963, it has been continually extended. Today there are 53 console channels that can handle approximately 48 physical terminals simultaneously; the largest number of simultaneous users to date has been 31, with an average in the mid-twenties during daily working hours. This expansion can be attributed mainly to an increase in drum storage (the swapping media) of approximately 35 percent, and an increase in the authorized system users to over 500. Many of these uses have been located in the nearly 50 remote organizations having system access. These organizations encompass a large community--the military, universities, law, urban and Federal Government groups, research organizations, and private contractors--with the common ties of performing research and development under ARPA sponsorship.

The time-sharing executive system provides the users with a variety of services, all accessed by on-line executive commands, and implemented by execution of system programs. As with any general-purpose system, the user may also run his own private programs.

Regardless of the program's origin, users may load, run, stop, debug, and edit the programs. The system handles accounting, scheduling of jobs, input/output message communication, file storage, management status and control, and error recovery. Users may use disc files, tape, teletype, and local cathode-ray-tube displays and RAND Graphic Input Tablets, interactively, in an on-line fashion. They may also submit jobs to the console operator to be run as "background" tasks in a fashion analogous to "job-shop" operation. Some of the language systems embedded in TSS include TINT, JOVIAL, LISP, META5, SCAMP (assembly language), EDIT, EDTXT--to name but a few of the more popular tools.

Progress

The past year has been primarily a period of consolidation and refinement of the system. This is so for two reasons: (1) The system runs well and error-free, providing an extensive range of capabilities, and further perturbations would not significantly improve it; (2) the system was previously scheduled for retirement in 1966, and even though its lifetime has been extended to 1967, plans were formulated upon the prior schedule. Even with these considerations, substantial system improvements have been realized.

TSS Executive. The major changes to the TSS Executive include:

. Scheduler - Inclusion of a new three-queue algorithm; allowing preferred users a larger share of central processing unit (CPU) time; inclusion of control logic to flash status indicator lights for certain consoles.

. Debug - The ability to display, on teletype (TTY), JOVIAL items in their symbolical defined formats; the ability to display, on TTY, machine registers in floating-point, hexadecimal, octal or integer format; special provisions to "patch" the system itself.

. Input/Output - Recovery from tape-read parity errors; implementation of automatic and manual control of the disc file "purging" algorithm to delete dated "junk" files;

*Supported by the Advanced Research Projects Agency.

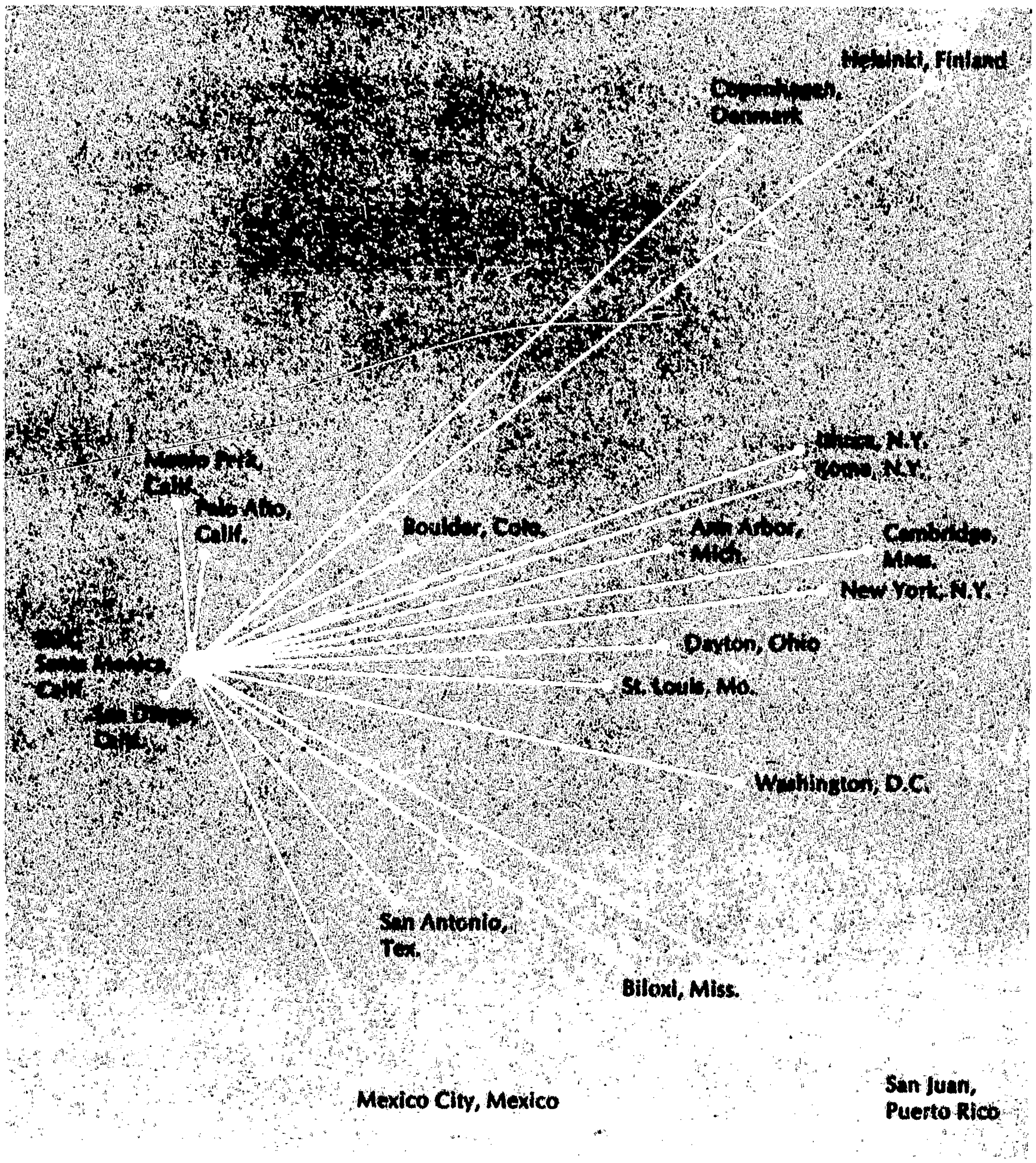


FIGURE 3-1. REMOTE USERS OF TSS

SDC's Time-Sharing System provides remote computing capability which makes the Q-32 computer accessible to users in various parts of the United States, its possessions, and Mexico. Only a few of the nearly 50 remote users are represented above. TSS has also been used intercontinentally from Copenhagen, Denmark, and Helsinki, Finland.

implementation of a dump of disc files onto a backup tape (the IFTAPE) as a user-controlled alternative to file purging; redesign of the PDP-1 executive to handle 53 channels and to handle IBM 1050 and TX-2 input; implementation of reload of disc files from the IFTAPE; provision to handle a RELOG call that combines the functions of QUIT, LOGIN, and (optionally) LOAD calls; ability of the system to load some services as object programs to yield an extended executive capability; ability to allow a program to alter its size dynamically (ALTER call); implementation of dynamic drum repacking to process ALTER calls.

. Miscellaneous - Inclusion of various frequency counters to determine areas of high system activity, e.g., system DISPATCHER calls, SERVICE calls, IFTAPE calls, etc.; relocation of internal disc inventory tables from disc to drum in order to speed system access and improve system responsiveness.

PDP-1 Executive. The PDP-1 computer is used as a real-time communications processor in the system. The PDP executive program has required numerous updatings to install major and minor additions and corrections. The two major modifications to the PDP-1 were those implementing the 1052 keyboard/printer and the processing of TX-2 communications. Both of these required rewriting other components and reassembly to make space available for the new functions in the PDP main core. Some of the other changes made during the year are:

. A provision to handle nine additional teletype stations.

. A provision to handle model 37 teletypes.

. Upper- and lower-case alphabet for the Soroban typewriter and the model 37 teletype.

. Facility to specify maximum line length on teletype output.

. An alternate method of processing RAND Tablet inputs, so that immediate Tablet inputs are processed by the PDP-1 and displayed as stylus tracks, i.e., are "inked."

EDTXT. A new system service, EDTXT, was designed and is now operational, complementing EDIT in the editing of disc files. EDTXT allows the user to edit his files by context, as well as by line number. He may request the insertion or deletion of any character string (up to 72 characters) in the file. This may be done within user-controlled columns and for one or all occurrences. EDTXT, in conjunction with EDIT and LIBRY, provides an improved capacity for symbolic file management.

LIBRY. To assist users in managing their own symbolic and binary files, the program LIBRY was made available during this period. LIBRY enables its user to manage files on his own master tape, by directing LIBRY via interactive commands to dump disc files of all forms (e.g., binary, program, symbolic, etc.) onto a single tape. The user may also purge, rearrange, or insert other files subsequently onto this master tape, which is created, inventoried, and controlled by LIBRY.

The need for LIBRY became critical during the year as the system load expanded. Because the limited disc storage remained static at 4 million words while the need for storage increased, disc files became more transitory and disappeared by automatic and manual file purging. LIBRY solved part of this problem by enabling users to save all their files on a single master tape, rather than saving them on separate tapes, one file per tape. Thus, LIBRY, the IFTAPE, and the disc-purging algorithms have combined to make TSS a working system that manages large files and allows users to manage their own files, without undue delays or hardships.

HELP. Because TSS documentation is voluminous, the HELP program was developed to allow users to query the system, on-line, for information found in the TSS documents. HELP gives the user information on TSS commands and error messages, JTS error messages, and DEBUG capabilities. In conjunction with HELP, which enters into a user-system dialog (see Figure 3-2), the EXPLAIN command provides an answer--on a "one-shot" basis--to any given query of the data base. HFL" and EXPLAIN are first experimental steps toward promoting complete user freedom from dependence on the off-line documentation of a time-sharing system.

LISTF. With the system control of disc files, a user has the ability to list all his current files in the system inventory. LISTF helps the user to housekeep the disc by deletion of obsolete files.

Computer-Aided Secretary. The goal of this project is to provide secretaries and technical personnel with a set of programs to speed up and facilitate the work of producing documentation. This entails providing the capability to enter text into a file, to edit the file, and to provide a set of commands as part of the file which dictate the final format of the document. (The commands, of course, will not appear in the final document.) With this capability, documentation of a program should not lag far behind its implementation, thus making the program available sooner to persons who may be interested in its capabilities. Experimental programs are being designed and written in JOVIAL and checked out on the Q-32, based upon similar work at MIT with the TYPSET and RUNOFF programs, and IBM's Administrative Terminal System (ATS). Eventually, these programs will be transferred to the S/360 TSS where they will be available as a system resource.

TSS Monitor. The program developed last year that displays, on a cathode-ray tube, the status and activity of all TSS users and an up-to-date display of the disc inventory, in both gross and detailed terms, has been a significant contribution to the operational management of TSS (see Figure 3-3). The program has been modified to reflect system changes and to meet the needs of the console operators who use the display for "tactical" decision-making. The strategic decisions are management's prerogative, reflected in the system access, scheduling, and priority given different users.

The combination of administrative procedures, a detailed status display of events and storage capacities within the system, and the ability of operators to take remedial action represents a comprehensive management scheme that could be usefully applied elsewhere. The modifications to the monitor program include:

. Work order number has replaced man number on the main display.

. The program now readily reflects the deletion of files from disc and the IFTAPE, as well as the effect of such deletions on available disc space.

. Two display modes are now provided: an abbreviated mode to prevent loss of characters on the display when 22 or more users are in the system; a detailed mode to display more detailed information about any one user.

. Inclusion of a light-pen "UNLOAD" capability by which the operator can place disc files on the IFTAPE.

Liaison and Demonstrations. Approximately 500 time-sharing briefings and/or demonstrations were given during 1966. Some of the presentations were given to small and informal groups and proceeded from chalk talks about the configuration and operation of the system to tours of the R&T Laboratory and teletype demonstrations of programs of interest to the visitors.

There were also many briefings and demonstrations given for large groups, both at SDC and during conferences in major cities, supported by such visual aids as closed-circuit television,

USER: !HELP

SYSTEM: THE FOLLOWING COMMANDS ARE RECOGNIZED BY HELP, AND CAN BE USED AT ANY TIME OR PLACE IN THE PROGRAM BY MERELY TYPING IN THE COMMAND WORD:

LIST	ALLOWS YOU TO OBTAIN A LIST OF SYSTEM COMMANDS OR SERVICES.
TSS	PRINTS AN EXPLANATION OF THE TIME-SHARING SYSTEM.
COMMANDS	ALLOWS YOU TO OBTAIN AN EXPLANATION OF SYSTEM COMMANDS.
RESPONSES	ALLOWS YOU TO INQUIRE ABOUT SYSTEM RESPONSES.
DEBUG	ALLOWS YOU TO INQUIRE ABOUT DEBUGGING CAPABILITIES.
JTS	ALLOWS YOU TO OBTAIN AN EXPLANATION OF JOVIAL COMPILATION ERROR NUMBERS.
DISPATCHER	ALLOWS YOU TO INQUIRE ABOUT DISPATCHER CALL MESSAGES.
EXIT	ALLOWS YOU TO EXIT TO THE TOP LEVEL OF HELP.
FINISHED	INDICATES TO HELP THAT YOU ARE THROUGH.

WHICH OF THE ABOVE ARE YOU INTERESTED IN?

USER: LIST

SYSTEM: WHAT DO YOU WANT LISTED--SERVICES OR COMMANDS?

USER: SERVICES

SYSTEM: THE FOLLOWING ARE THE SYSTEM SERVICE PROGRAMS:

DBUG	EDIT	EDTXT	SURE	TINT
LIBR	DTAP	JTSBIG	JTSMAL	

NOW WHAT DO YOU WANT TO KNOW ABOUT?

USER: TSS

SYSTEM: THE SDC TIME-SHARING SYSTEM ALLOWS SIMULTANEOUS ACCESS TO THE Q-32 COMPUTER BY MANY USERS. THE EXECUTIVE PROGRAM CONTROLS THE SYSTEM. TO COMMUNICATE WITH THE EXECUTIVE, YOU MUST PRECEDE COMMANDS WITH AN EXCLAMATION MARK (!). TO ENTER THE SYSTEM, USE THE "LOGIN" COMMAND, FOLLOWED BY YOUR MAN NUMBER AND WORK ORDER NUMBER. AFTER YOU ARE LOGGED IN, YOU MAY LOAD AN OBJECT PROGRAM OR SYSTEM SERVICE PROGRAM BY USING THE "LOAD" COMMAND, FOLLOWED BY THE PROGRAM NAME OR TAPE NUMBER. ONCE YOUR PROGRAM IS LOADED, YOU MAY START IT BY TYPING IN THE "GO" COMMAND, AND STOP IT BY TYPING IN THE WORD "STOP". TO LEAVE THE SYSTEM, TYPE IN THE WORD "QUIT".

NOW WHAT DO YOU WANT TO KNOW ABOUT?

USER: COMMANDS

SYSTEM: ENTER THE COMMAND TO BE EXPLAINED.

USER: LISTF

SYSTEM: LISTF ALLOWS YOU TO OBTAIN A LIST OF YOUR OWN FILES BY TYPING IN THE WORD "LISTF". THE SYSTEM PRINTS THE NAME AND FILE SPECIFICATION FOR EACH OF YOUR FILES.

NOW WHAT DO YOU WANT TO KNOW ABOUT?

USER: JTS

SYSTEM: TYPE IN THE JTS ERROR NUMBER.

USER: 32

SYSTEM: THERE IS AN ERROR IN AN "IF" STATEMENT.

FIGURE 3-2. SECTION OF "HELP" DIALOG BETWEEN USER AND SYSTEM

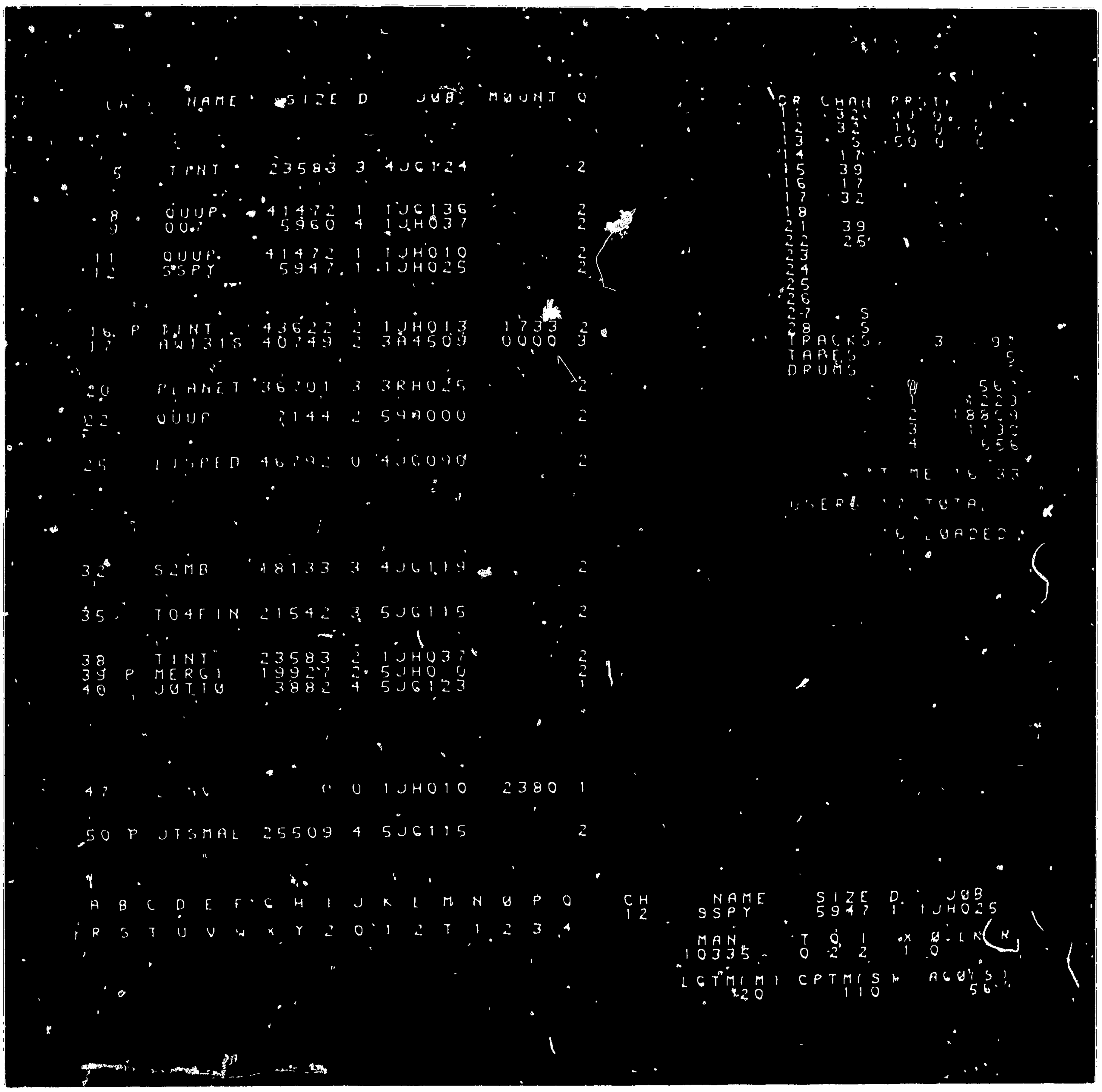


FIGURE 3-3. TSS MONITOR PROGRAM STATUS DISPLAY

The columns in the display indicate to the computer operator the following items: channel number (CH), program name (NAME), program word size (SIZE), drum number on which the program is stored (D), work order number and priority (JOB), magnetic tape reel number (MOUNT), and the status of the program in the job queue (Q). Also indicated are the tape drives being used by the various channels (DR and CHAN), the number of jobs placed in a production stack (PRSTK), the availability of tape and drum tracks,and other useful information. By light-penning the letters and numbers in the lower left corner of the display, the operator can request a display of current files sorted in various ways. The information in the lower right-hand corner gives additional data on a current system user (in this case, the person using channel 12).

slides, multiple teletype installations, and documentation. Some of the major external demonstrations included: American Management Association in New York City; New Jersey Hospital Association in Princeton, New Jersey; the Commanders' Conference, Air Force Logistics Command, Hill Air Force Base, Ogden, Utah; and the National Industrial Conference Board Seminar, New York City.

Among the year's visitors to SDC were representatives from: ARPA, ARADCOM, Office of Naval Intelligence, Pacific Air Command, Aerospace Medical Research Labs, Adjutant General School, U. S. Marine Corps, Office of Secretary of Defense, Joint Chiefs of Staff, Naval Electronics Laboratory, NORAD, Royal Canadian Air Force, Air Force Systems Command, College of the Armed Forces, Office of Anti-Submarine Warfare Programs, and the National Range Division.

Representative of the many visitors from the field of education were those from the American Council of Education, Cornell University, National Educational Association, Pennsylvania State University, the Claremont Colleges, UCLA, USC, Canadian Ministry of Education, Japanese Ministry of Education, and the Assistant to the Deputy Administrator of Education, Ontario, Canada.

Other briefings and demonstrations were given for personnel from NASA, U. S. Department of State, Office of Aerospace Research, National Science Foundation, U. S. Geological Survey, U. S. Civil Service, Bureau of the Census, and Oak Ridge National Laboratory.

Interest from abroad was evidenced by visitors from Cegos Informatique, France; Computer Center, Israel; the Japanese Air Force; IBM, Sweden; and NATO.

The inventory of demonstrable programs, developed within R&TD, is given in the Appendix, page A-1.

Plans

As of November 17, 1966, the Q-32 TSS has been placed into service on a limited subscription basis to a subset of those users, local and remote, previously using the system. Purchase of a share entitles a subscriber to the best preferred service available on one shift--four console hours a day, five days a week. There are four shifts daily. It is anticipated that few or no changes will be made to TSS, except as required by periodic and emergency maintenance. Some additional services may be attempted as object programs without disturbing the basic system executive. In particular, some changes can be expected in the PDP-1 executive to satisfy current commitments in the TX-2/Q-32 network link, and in experimental exploration of improved display "tracking" and "inking" of RAND Tablet input.

Project Documentation

1. Schwartz, J. I. The SDC time-sharing system, part I. Datamation, 1964, 10(11), 28-31; part II, 1964, 10(12), 54-55. (Also available as SDC document SP-1722.)
2. Bain, H. Time-sharing: its users and uses. SDC Magazine, November 1966, 9(11), 1-15.
3. Linde, R. R. and Chaney, P. E. Operational management of time-sharing systems. SDC document SP-2392, June 6, 1966, 30 pp.

RESEARCH IN TIME-SHARING*

C. Weissman, Principal Investigator
W. J. Erikson
B. L. Jones
R. A. Totschek
P. A. McIsaacs
G. H. Fine

Description

The projects undertaken in this area cut across various lines of time-sharing activity.

*Supported in part by the Advanced Research Projects Agency.

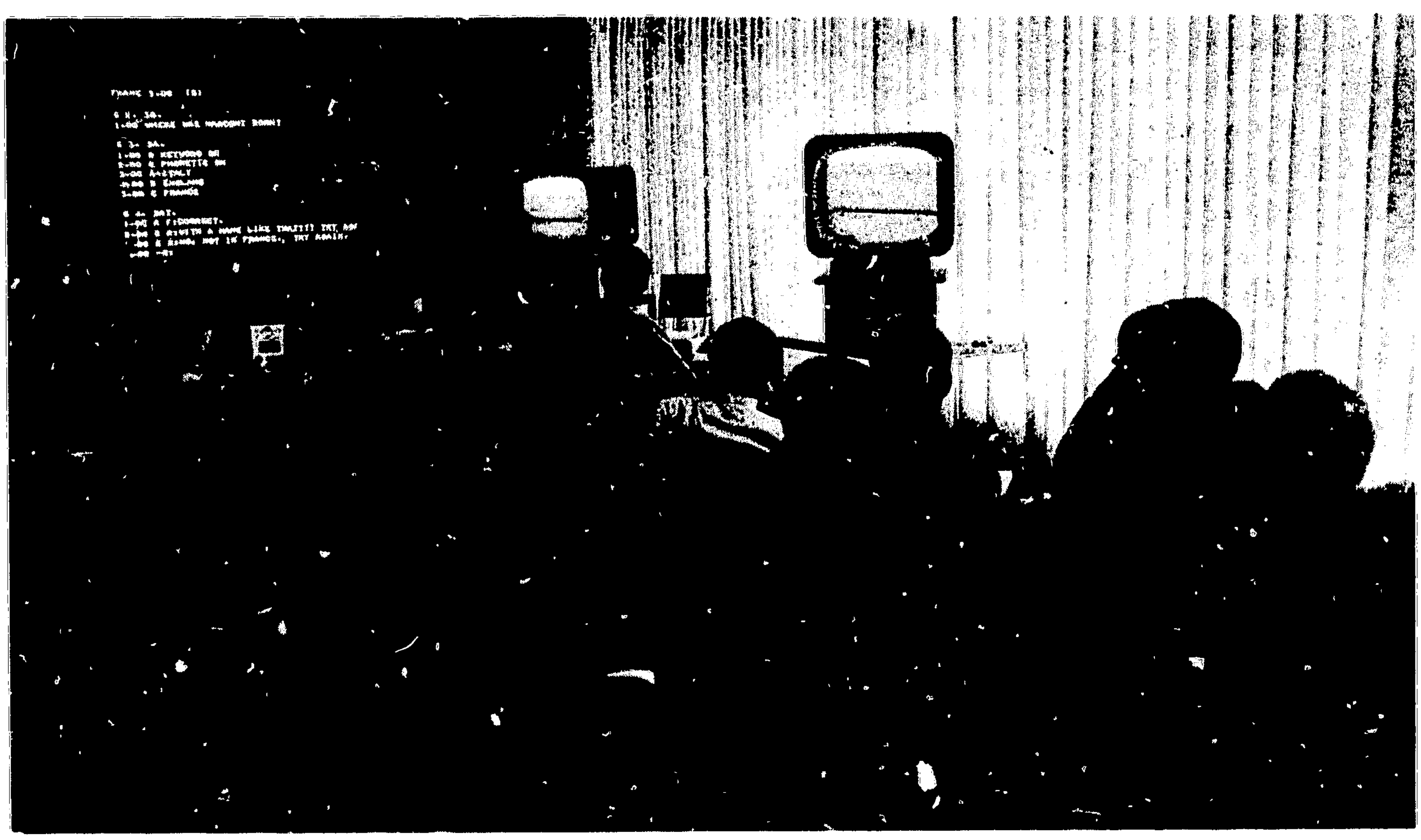


FIGURE 3-4. BRIEFINGS AND DEMONSTRATIONS ON SDC'S TIME-SHARING SYSTEM AND PROGRAMS

They all relate to further understanding of the behavior of the SDC Time-Sharing System (TSS), and to formulating plans for improving or extending the scope of time-sharing. To this end, the analysis of time-sharing carried out in prior years continued this year. The emphasis has shifted, somewhat, from analysis to simulation and modeling, and to experimentation and empirical data collection. Some of these activities reached their goals and have been terminated; others continue to explore the area of interest more deeply.

Progress

Empirical and Simulation Analyses of Time-Sharing. Since publication of the results of earlier analysis of TSS [3], the system has undergone considerable change--more drums, improved disc file management, more and improved user-oriented programming and language tools, and extended system access. To update this analysis and to reconsider scheduling strategies based on the updated analysis, additional simulation and raw operational data were collected and processed. To improve the quality of simulation data, the simulators were driven by input consisting of job- and user-characteristics collected from previous operational data. New data were collected that supported the validity of the old data, and hence the validity of the simulation runs. The simulators output data according to various options: plotted cumulative response time as a function of queue discipline; frequency and length of CPU requests (from programs); and number of quanta serviced per unit time (throughput).

The results of analysis of newly collected operational data and simulation were used as the basis for improving TSS operation. They suggested a new, three-queue algorithm; lowering system overhead by moving the disc inventory tables and processing programs to the drum (as noted earlier in the section on TSS development); and adjusting the number of quanta given each program when it reaches the head of one of the three queues. Figure 3-5 shows the overall functional flow of the present three-queue scheduling algorithm. With the publication of the final report on these studies [2] (and with TSS now in its new subscribed role), this project has been completed.

Paging Study. The development of second-generation time-sharing systems based on virtual-memory addressing schemes (where secondary storage is mapped into real core memory in "pages") has created concern over the dynamic program behavior in the "paged" environment of such systems. To date, only qualitative predictions have appeared in the literature. This study has attempted to provide quantitative data on the behavior of a variety of current time-sharing system programs operating in a simulated paging environment. An attempt was made to find the tradeoff points in system overhead between "demand paging"--fetching a new page whenever required--and complete program swapping. From data on extreme conditions, it was hypothesized that more meaningful algorithms could be generated for less extreme conditions, such as set-of-pages swapping. Formulas were derived for expressing core occupancy in page-per-millisecond units for both full swap and paging-on-demand techniques. Other formulas were used to evaluate these data as a function of program size and other parameters of interest. Data for use in these formulas were collected by embedding an interpretive executive in various TSS object programs, e.g., JTS, TINT, LISP, EDIT, etc. This executive executed the object program for typical user applications and

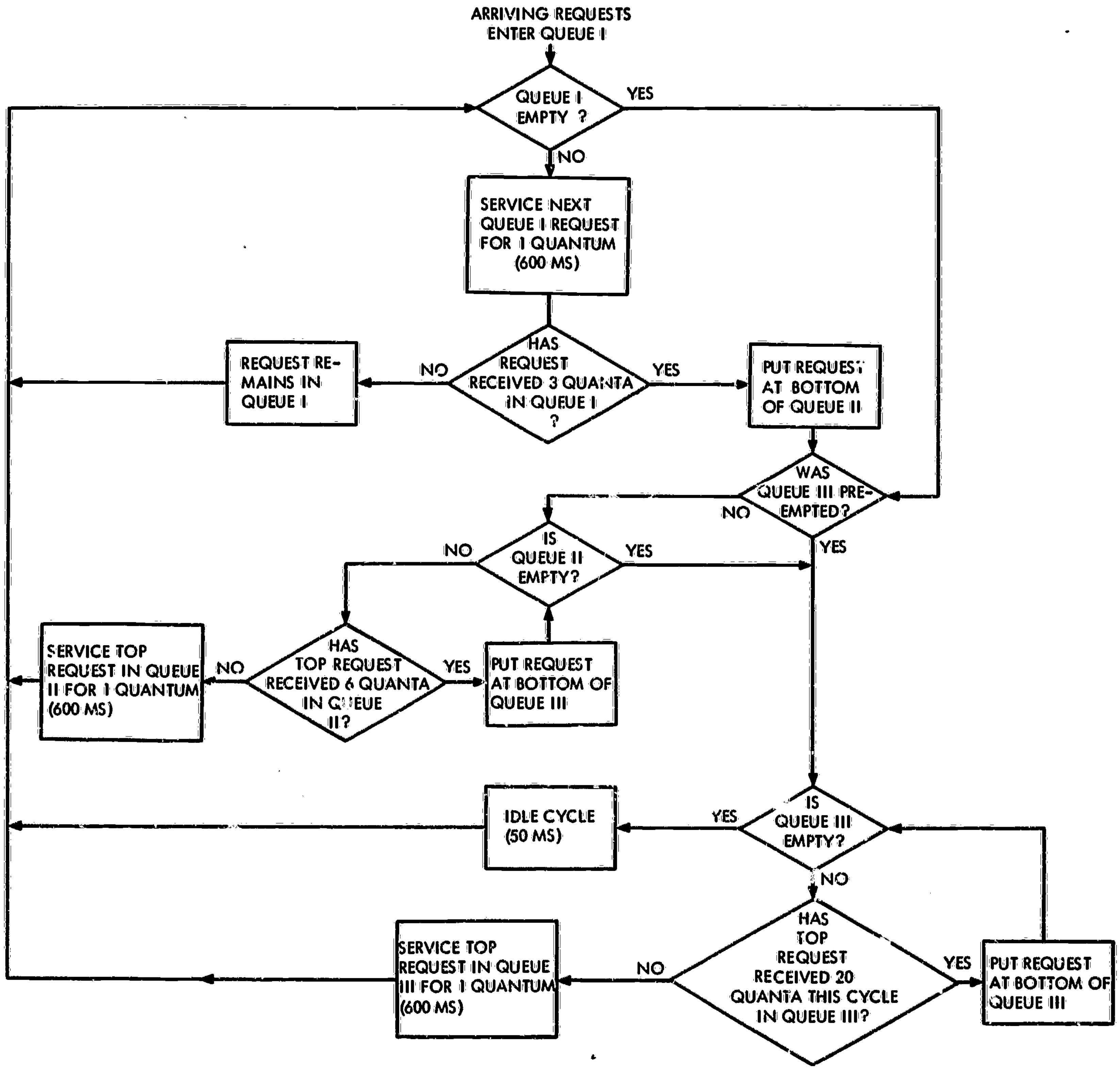


FIGURE 3-5. FUNCTIONAL FLOW OF TSS SCHEDULING LOGIC

collected various statistical data on paging parameters.

The data obtained seem to indicate that the handling of programs similar to the ones examined may be difficult in a time-sharing environment utilizing a paging-on-demand strategy. Attempting to alleviate these difficulties by reorganization of the programs creates new problems in determining what strategies can be used to automatically perform the reorganization. The collected and analyzed data summarized in Figure 3-6 have been supplied to interested second-generation time-sharing system developers. The project was concluded with the presentation of the findings at the 21st National ACM Conference in August [1].

Comparative Economics of Computer Systems. This project has been concerned with determining the value of different types of computer systems to the managers of these systems. The primary measure of value has been total cost, where total cost includes the costs of both the system and the users of the system. The study of one aspect of this problem (the differences between interactive and noninteractive program debugging) has taken a significant portion of the year.

With regard to the overall problem, cost models were developed that allow comparison of different systems. Also, a set of computer system models has been derived for each of the computer systems under study (time-sharing, batch processing, personally operated, and multiple-console open shop). These models have been employed to determine how the costs of the different systems vary as system and user parameters are varied. However, while these models allow interesting comparisons to be made between the different types of systems, they do not adequately describe the process of program debugging. Because of this, an experimental pilot study was designed to determine the effects of interactive versus noninteractive program debugging [4]. The experiment was conducted to obtain data for the larger study to help determine the economics of different types of computer systems. In addition, the data would permit an analysis of changes in the performance characteristics of users who were debugging programs on different types of computer systems.

This experiment is similar in many respects to the study "Programmer Performance Under On-Line and Off-Line Conditions," described below. Both experiments used the Q-32 TSS for one of the test conditions (on-line, or interactive). However, the off-line condition in the study described below involved a simulated batch-processing system, while this study used a simulated multiple-console, open-shop system for the noninteractive condition. In addition, different languages were used by the subjects of the two experiments; the test problems were much more difficult in the study described below; and the subjects in this study were inexperienced programmers. However, even with these wide differences, these two independent studies reinforced each other by indicating the same basic conclusions:

. The subject groups using interactive time-sharing consistently took substantially less time to debug their programs.

. Very large differences in individual performance were noted between programmers.

Plans

With regard to the first two projects reported --the analyses of time-sharing and the paging study--no further work is contemplated. There is a considerable amount of work that remains to be done, however, on the comparison of computer system costs. Before conclusions can be presented concerning the overall cost

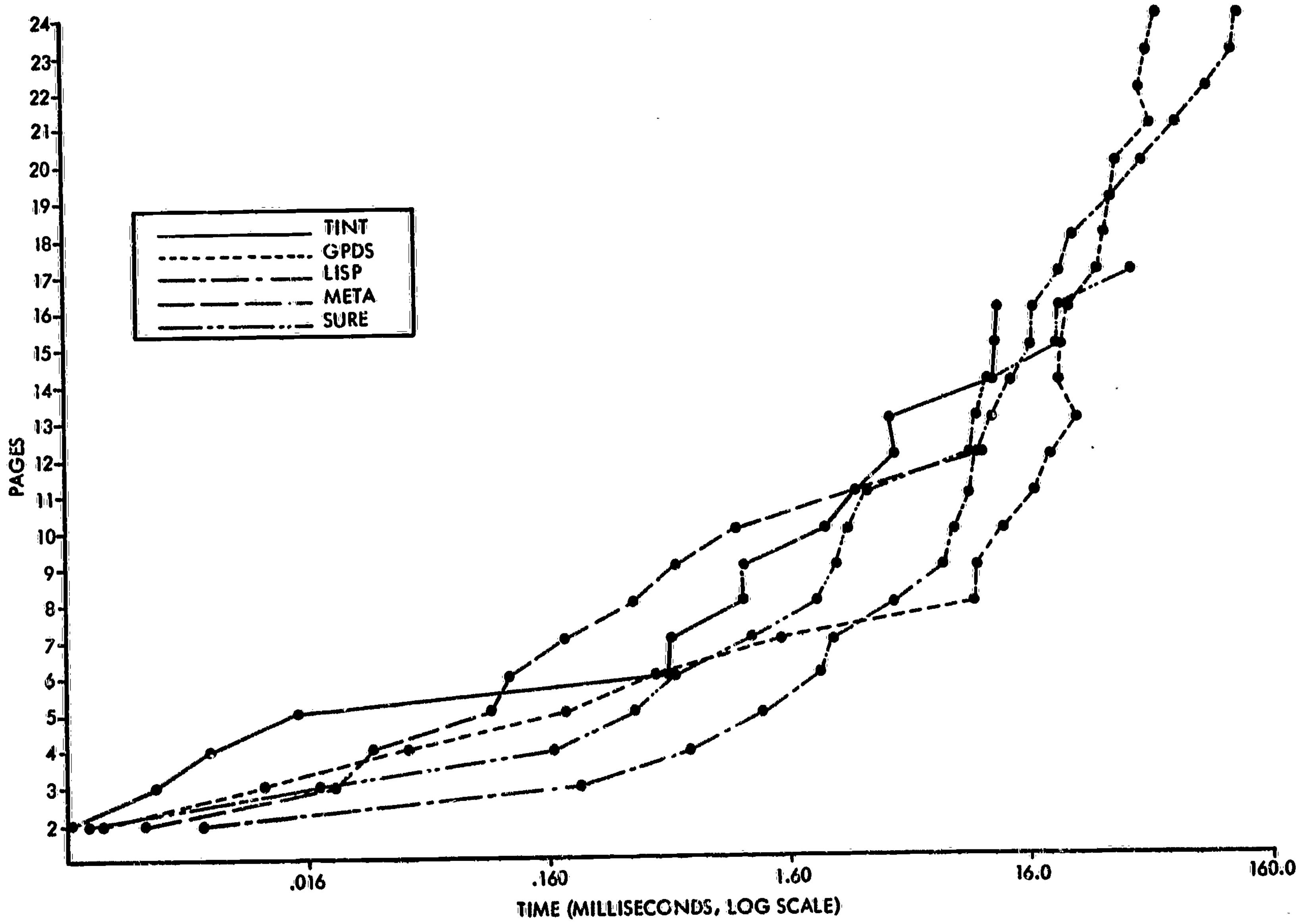


FIGURE 3-6. PAGE DEMAND STUDY

The graph indicates cumulative total number of pages referenced by various programs as a function of elapsed time from initial program execution.

differences between the various types of computer systems under study, several preliminary questions must be answered. The large differences among individual programmers as well as difficulties in defining and measuring job characteristics have led to the recommendation that further study be directed toward determining the characteristics of the users (programmers), the jobs, and the systems used. Then a more successful attempt could be made to determine how these and related considerations interact and how much they cost in the performance of a specified job for a particular type of user. Documentation of this work is in progress.

In general, the major effort in the time-sharing area will be one of consolidation and transfer of Q-32 supported research to the System/360 Time-Sharing System.

Project Documentation

1. Fine, G. H., Jackson, C. W., and McIsaac, P. V. Dynamic program behavior under paging. SDC document SP-2397. June 16, 1966. 19 pp.
2. McIsaac, P. V. Job descriptions and scheduling in the SDC Q-32 time-sharing system. SDC document TM-2996. June 10, 1966. 28 pp.
3. Totschek, R. A. A statistical analysis of the SDC time-sharing system. SDC document SP-2191/001/00. January 18, 1966. 18 pp.
4. Erikson, W. J. A pilot study of interactive versus noninteractive debugging. SDC document TM-3296. December 13, 1966. 21 pp.

PROGRAMMER PERFORMANCE UNDER ON-LINE AND OFF-LINE CONDITIONS*

E. E. Grant
H. Sackman

Description

This study was conceived as an experimental comparison of on-line versus off-line programmer debugging performance. The SDC Time-Sharing System was used for the on-line condition and for a simulation of the off-line condition. A set of rules and procedures was developed to make the off-line simulation as comparable as possible with the facilities of current batch-processing systems. Turnaround time in the off-line mode was held constant at two hours, a response time considerably faster than turnaround times for most off-line installations.

The criterion performance measures were man-hours for program debugging, recorded by the experimenters, and the central processor time (CPU time), recorded by the computer for each subject. These criterion measures were collected for two programming problems--an algebra problem involving solution of equations, and a maze problem requiring determination of the correct path through a given maze. The experimental programs were considered completed when they successfully processed standard inputs.

The experiment was run in accordance with a latin-square design with repeated measurements. Two groups of programmers, with six subjects in each group, solved the two problems under on-line and off-line conditions for debugging (as indicated in Figure 3-7). The total sample of 12 subjects had an average of seven years' programming experience.

	On-Line	Off-Line
Group I	Algebra (6)	Maze (6)
Group II	Maze (6)	Algebra (6)
Totals	(12)	(12)

Figure 3-7. Programmer Debugging Performance Experimental Design

The experimental design has the advantages of pooling all 24 measurements for each criterion variable (resulting in a larger sample), and of using each subject as his own

*Supported by the Advanced Research Projects Agency.

control. The analysis of variance with this design permits separate tests for on-line/off-line and problem differences.

Data on additional performance measures were collected; these included coding man-hours, program length, and the running time of the completed programs. Data on several control measures were also collected: general programming experience, SDC Time-Sharing System experience, type of programming language used, and scores on the two parts of the Basic Programming Knowledge Test, recently developed at the University of Southern California.

Progress

The experimental data raised some unanticipated problems. The main problem was the enormous range of individual performance differences, usually varying over an order of magnitude between best and poorest scores. These individual differences were typically observed in positively skewed frequency distributions, and were so large that they tended to overshadow on-line/off-line treatment effects--an important finding in its own right. These problems were partly resolved by the use of statistical controls.

The results of this study [2] showed debugging man-hours to be significantly shorter for the on-line condition at the .025 level. There was a nonsignificant tendency toward greater CPU time under the on-line condition. In terms of mean raw-score values, debug man-hours off-line ranged from 50 percent larger for the algebra problem to approximately 300 percent larger for the maze problem (50:35 and 12:4, on-line versus off-line debugging man-hours, respectively). It was concluded from these results, under the conditions of this experiment, that on-line debugging was significantly and substantially more efficient than off-line debugging in terms of programmer man-hours, and that there was some tendency for more CPU time to be required for the on-line condition.

To analyze the dimensions of individual differences, the correlation matrix of 15 performance and control variables was subjected to an exploratory factor analysis. The results showed two general and fairly well-defined factors that were consistent with trends in the empirical correlations. The larger factor, showing heaviest loadings with criterion performance and coding speed variables for both problems, and with programming language, was designated "programming speed." The second factor, showing heaviest loadings with program size and running time, was described as "program economy." General programming experience showed some relation with this factor, indicating a tendency for more experienced programmers to write shorter code associated with faster program run time.

In the interpretation of the data, limitations and constraints of the findings were noted. It is not known how representative the algebra and maze problems are with respect to problems "typically" faced by programmers. The experiment was conducted at a single facility with on-line operations the normal mode, and with simulated off-line operations. Only two-hour turnaround time was tested. The effects of varying levels of turnaround time were not systematically investigated and could only be tentatively extrapolated from the results. These extrapolations did indicate, however, that the performance difference between on-line and off-line operations for debugging man-hours tends to disappear as turnaround time approaches zero for the off-line condition (see Figure 3-8). The small size of the experimental groups, combined with large individual differences, led to large error variances in experimental

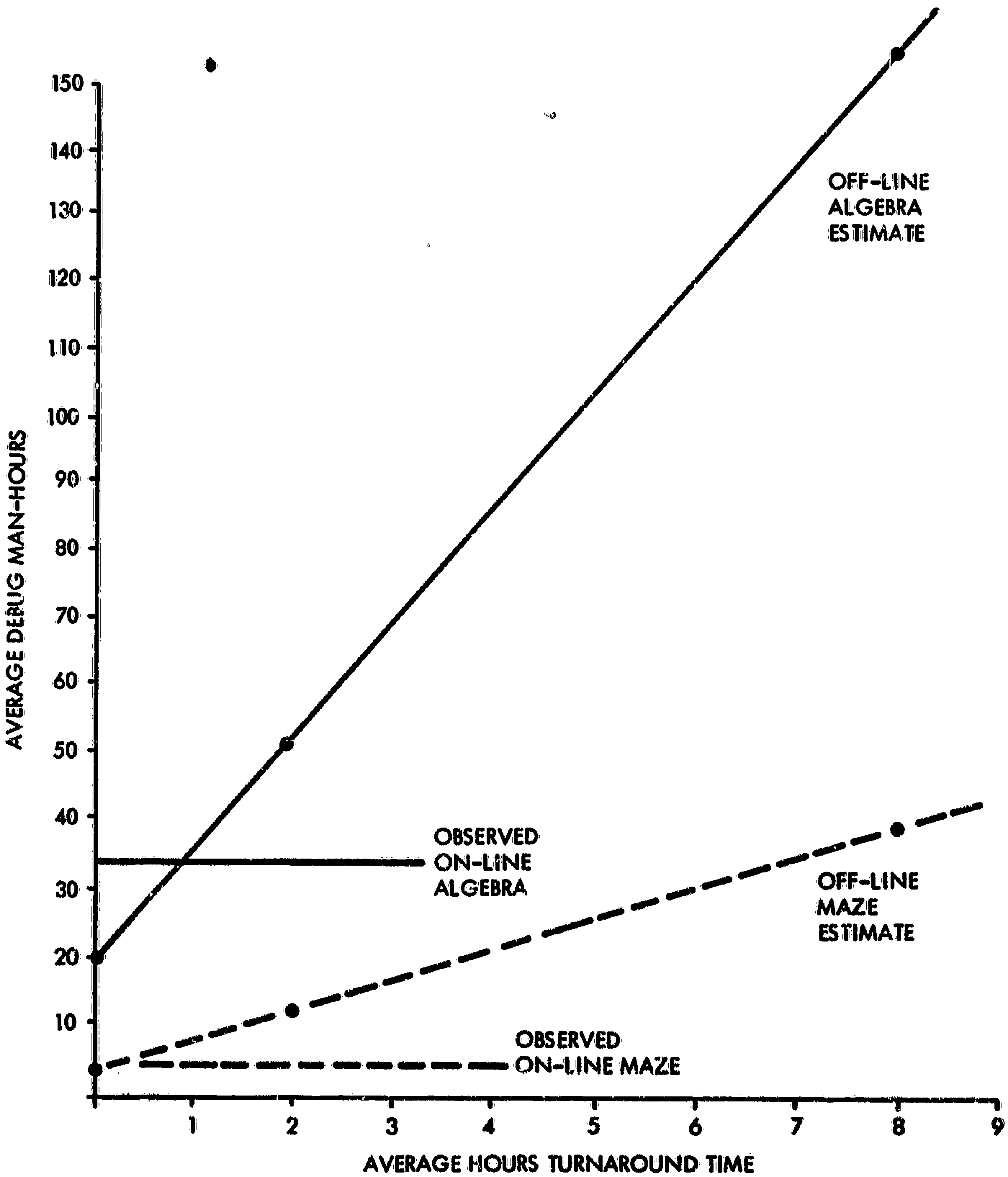


FIGURE 3-8. DEBUGGING MAN-HOURS ESTIMATED AS A FUNCTION OF TURNAROUND TIME

The slope of the curve indicates that the rate of increase of debugging man-hours is a function of turnaround time. For the on-line condition, turnaround time is (effectively) zero; for the off-line condition, turnaround time was an experimentally fixed parameter (from which these curves were extrapolated).

measures and to some ambiguous results. There were problems in describing the subject sample in relation to empirical programmer populations. Independent experimental studies were recommended, utilizing other facilities, with other program problems, on larger and more diversified programmer samples, and with varying levels of turnaround time.

Plans

This study indicated that valid generalizations of user performance in time-shared computing facilities are contingent upon an established body of empirical findings on users--their needs, tasks, problems, resources, experience, skill levels, motivation, and interactive problem-solving behaviors. Since these empirical data are virtually nonexistent, it was apparent that the piecemeal, haphazard pattern of current user studies in time-sharing should be restructured along the lines of an organized conceptual framework for the field as a whole, leading toward a balanced, long-range program of applied research to anticipate and meet user needs. Accordingly, plans for future work involve the collection, review and evaluation of experimental studies on user performance in time-shared computing systems, followed by the construction of a conceptual framework for long-range development and growth of experimental studies on user performance.

Project Documentation

1. Grant, E. E. An empirical comparison of on-line and off-line debugging. SDC document SP-2441. May 18, 1966. 16 pp.
2. Grant, E. E. and Sackman, H. An exploratory investigation of programmer performance under on-line and off-line conditions. SDC document SP-2581. September 2, 1966. 36 pp.

TIME-SHARING COMPUTER NETWORKS*

C. E. Fox, Project Leader
L. E. Durham
G. H. Fine
L. Gallenson
R. F. von Buelow

Description

Many consider the next logical step in the evolution of time-sharing computer systems to be the linking of these systems into networks. Such networks would greatly increase the machine facilities available to cooperating users, as well as allowing widely separated computer installations to share programs, data, and costs with other, similar installations. Preliminary studies of time-sharing networks indicate that they are today technically feasible.

SDC has become involved in two distinct approaches to establishing computer networks. The first involves a network of dissimilar machines (the TX-2 computer at Lincoln Laboratory and the Q-32 computer at SDC); the second involves a network of similar machines (SDS 940 computers). At the present time, the first network has been implemented; the second is awaiting design approval.

The link between the TX-2 and the Q-32 is via a Western Union 1200 BPS line. The TX-2 connects to the SDC PDP-1 computer teletype interface; in all respects (except for higher speed), the TX-2 operates like a teletype on the Q-32. Thus the TX-2 can log in, load, and execute Q-32 programs (and vice versa). This effort involves personnel from SDC, Computer Corporation of America, and Lincoln Laboratory in writing protocol documents, testing programs, and incorporating modifications to the PDP-1 and TX-2 executive programs.

*Supported by the Advanced Research Projects Agency.

SDC has likewise been involved in the design of a network of SDS 940 computers. A design team, headed by Wayne Lichtenberger of the University of California, Berkeley, met at SDC for a week in June 1966 to summarize a proposed design for such a network. The summary, edited by Dr. Lichtenberger, was submitted to the Advanced Research Projects Agency in September.

Progress

The TX-2/Q-32 link is currently being used for "shakedown" and demonstration. Object programs have been written for both machines to interpret the protocol and react to the transmitted data. Complete line transmit/receive ("echo") testing has been accomplished. In addition, a TX-2 program has accessed the Q-32, logged in, loaded TINT, and computed with this system; the results of the computation were then routed back to the TX-2. Q-32 use of the TX-2 has been more restrictive, inasmuch as the smaller-scale APEX executive system (on the TX-2) has dedicated much of its resources to display programs.

The specification written for the SDS 940 computer network considers the various "ideal" connections between elements of a network--user, file, process, memory, and system. The element interconnections give rise to a basic system configuration and determine the capabilities required by the network. The proposed design encompasses message flow, file and memory handling and referencing, and network performance measurement. It also delves into detailed hardware and software modifications to the existing SDS 940/Berkeley time-sharing system.

Plans

Consideration has been given to building object programs on both the TX-2 and the Q-32 that accept and reformat display data for transmission, and route the displays for posting on the other machine's CRTs. Before this is possible, protocol conventions must be adopted for display data transmission, and various hardware and software modifications must be made. Work on this link will probably continue on the Q-32 until its deactivation, and then will be transferred to the System/360.

Further progress on the SDS 940 computer network is pending, awaiting industry response to the design specification currently being circulated by ARPA. In general, networking activities are expected to continue to stimulate the imagination of researchers and users alike.

Project Documentation

1. Lichtenberger, W. (Ed.) Tentative specifications for a network of time-shared computers. Advanced Research Projects Agency Document No. 40.10.130, Washington, D. C. September 9, 1966.

DATA BASE SYSTEMS

A. H. Vorhaus, Head

Developing general-purpose data management systems for nonprogrammers is the primary goal of the Data Base Systems staff. A data management system in this context is one that defines, stores, processes, and outputs data according to user requirements. The generality of the system allows the same program set to operate on many different kinds of data and different types of data problems. Providing this service for nonprogrammers gives the actual user of the data immediate access to the information in his data base without the traditional intermediaries needed in the past.

The design of data management systems has taken on new direction through the use of time-sharing and advances in computer hardware. It is now possible for the user to economically communicate directly with the computer program and to have random access to large files of data. SDC has applied the findings of its research to the development of new program systems that facilitate user interaction and take maximum advantage of the new technology.

Two indications of the effectiveness of this approach can be seen in the response of current users of TSS-LUCID (Time-Shared Language Used to Communicate Information System Design) and the degree of interest shown in thisprogram by visitors to SDC. TSS-LUCID was used by about 50 people each month during 1966. The system combines the use of an English-like query language with a data storage structure that provides random access data retrieval and operates under time-sharing. The results gained from this method are rapid response to data requests, ease in formulating data requests, and economy of operation (since actual compute time is small compared to total elapsed time). The adaptability of TSS-LUCID to variable-length names, as well as numeric data, has led to wide diversity in the types of users and data structures employed. A data base describing a patient's record at a hospital--including diseases, consultations, results, and personal information--is as feasible as a data base containing inventory information about different products--including quantity, location, cost, condition, codes, etc. This ability to adapt to various kinds of data without the need for costly reprogramming is one of the most powerful features of TSS-LUCID.

Work was initiated to incorporate text handling within TSS-LUCID. Some changes to the programs were made and a small data sample was used as a test case. Test results were satisfactory, indicating the general feasibility of adapting TSS-LUCID to the processing of textual materials. However, an attempt to use larger data samples pointed up some capacity limitations that could not be easily overcome. It appears that further work on combining free text with structured data must await development of the successor to TSS-LUCID.

The use of a list processing language for data management is also under investigation. A LISP 1.5 program that models TSS-LUCID exists; various comparisons between the two are being made. Results demonstrate that while LISP 1.5 allows greater ease in program modification, it requires more storage than TSS-LUCID. Additional work will continue using the more powerful LISP 2 language.

Another program system of considerable interest is GPDS (General Purpose Display System) for use in developing cathode-ray-tube data management displays. Interfacing with TSS-LUCID, GPDS allows the user to operate a teletype for factual data retrieval and summary information, and then to use the CRT for visual displays. However, because GPDS and TSS-LUCID have totally different designs, each employs its own language and method of operation. Whereas TSS-LUCID has limited outputs, the powerful communication language of GPDS provides diverse and flexible output display formats. GPDS, on the other hand, operates interpretively and receives no special priorities from the time-sharing system for operating the CRT display. This causes GPDS to operate very slowly except for the most trivial of applications. Also the complexity of GPDS necessitates a long learning period for potential users.

At the start of the year a major effort was made to apply GPDS to the task of calculating salary maturity curves. This proved a worthwhile undertaking in several ways. While it confirmed the original evaluation of GPDS, it also illuminated other problem areas such as sequence control, capacity limitations, and overall complexity of the language. At the same time it brought to light the desirability of the building-block concept, the acceptance of the query modifiers, and the excellence of the array handling logic.

All work on TSS-LUCID and GPDS has stopped; future efforts are being directed to the Time-Shared Data Management System. TDMS will be an integrated data management system allowing complex data structures, on-line querying, report generation, CRT displays, and update and maintenance capabilities--a using a consistent language. TDMS will operate on an IBM S/360 series of computers in a time-shared environment. It is being designed for broad user application and will retain the simplicity of operation of TSS-LUCID. Its CRT display capabilities will be restricted, but easy to use. It will benefit from the GPDS experience, but will be neither an extension of GPDS nor a reasonable facsimile.

The data base structure provided within TDMS may be extensive and complex. This places greater demands on the user to have detailed knowledge of the data base components; it also requires that the query language be powerful enough to effect meaningful data retrieval. An investigation is being directed toward easing this task through the use of natural language interrogation. The approach involves developing a question preprocessor which translates a user's flexibly structured sentence into the appropriate TDMS query requests.

Interest in TSS-LUCID and the development of TDMS has led Shell Oil Company and Atlantic-Richfield Oil Company to place research associates at SDC to participate in our data management work. A tenure of 12 to 18 months is expected, at which time they will return to their parent companies.

The experience gained from using both TSS-LUCID and GPDS--which are general-purpose, user-oriented, and deal with structured data--has influenced the design of TDMS greatly. The research performed during the past year, as well as the previous years' experience, has led to several conclusions regarding the design requirements for TDMS: (1) It is necessary to keep user interaction simple, even at a sacrifice in capability. Users are willing to accept less power as long as they can have quick and easy access to the system. (2) It is desirable to keep flexibility and user choice to a minimum. While some users may be impressed by the number of buttons and options available, most users will generally not be willing to pay the price for a system loaded with accessories. (3) Prospective users must be involved early in the system design. Designing a system without direct user interface leads to systems that soon become discarded through disuse. (4) The system must respond rapidly enough to satisfy typical users. Even though users know they are sharing the time-sharing system with others, each additional second of waiting frustrates them and leads to disenchantment with the system. An interpretive system must be capable of producing a compiled machine code version for frequently run jobs to provide the necessary speed of operation.

FIGURE 4-1. GPDS CONSOLE

Inputs can be made on-line via the RAND Tablet, lightpen, teletype, or special bank of switches on right of console.

GENERAL-PURPOSE DISPLAY SYSTEM*

S. Bowman, Project Leader
S. I. Ackley
S. H. Brown
M. H. Detambel
M. D. Draper
R. A. Lickhalter
R. H. Sterneck

Description

The General-Purpose Display System (GPDS) combines a user-oriented programming language with display equipment input to achieve general-purpose CRT display formats. Operating under time-sharing on the Q-32, GPDS provides an experienced user with the ability to experiment with various display formats, accessing data generated by TSS-LUCID.

A building-block approach was chosen for GPDS operation. The on-line user has at his disposal a set of basic processes which are used to adapt GPDS to the particular task being performed. Queries are presented to the user, and his responses are stored by the program. By calling several of these basic processes in succession, the user builds a specific display in which he is interested. He can then save this selected sequence as a new process, thus linking together a sequence of basic processes combined with the user's responses and inputs. At a later time, the user may run this saved process with a different data subset, or he may use the process in the building of a more complex display format.

Progress

In this report period GPDS continued to be demonstrated to a large number of visitors and interested persons within SDC. The demonstrations given early in the year continued to indicate that--for most users--GPDS was too slow and complex.

The language in which the user communicates with the system is powerful, but highly complex. A significant time period is required for learning the language, and even the most experienced users have difficulty remembering all its features. As a result, the user has to plot the solution to his problem in minute detail and needs programming skill. User interaction is also somewhat complicated by the wealth of input equipment available, including a RAND Tablet, light gun and CRT, teletype, auxiliary keyboard, and function button box.

Another limiting factor is the slowness of operation for running a saved process. Whereas an interpretive system may be quite adequate while the user is initially constructing a display, the system must provide rapid response when the user generates successive displays with the finished process. For a general-purpose program to be useful for any specific application, it must provide some technique for response equivalent to a special-purpose program. Such a requirement may sometimes be met by compiler methods. Despite these shortcomings, GPDS has been a valuable tool for gaining insight into man-machine interaction and has been applied to the production of meaningful work.

In February, a meeting was held at SDC with leaders in the field of graphics* to discuss GPDS and its future direction. It was suggested at this meeting that GPDS project personnel concentrate on solving a specific problem to better learn the assets and drawbacks of GPDS. The problem selected for solution was the automatic generation of salary maturity curves, using data collected from Southern California aerospace firms (see Figure 4-2).

To facilitate this task, several new basic processes were added to GPDS. These basic

*Supported by the Advanced Research Projects Agency.

*John K. Summer, The MITRE Corporation; Dr. John Egan, Headquarters, Electronic Systems Division; Douglas Ross, Massachusetts Institute of Technology; Professor Bert Herzog, University of Michigan; Jerry Feldman, Lincoln Laboratory.

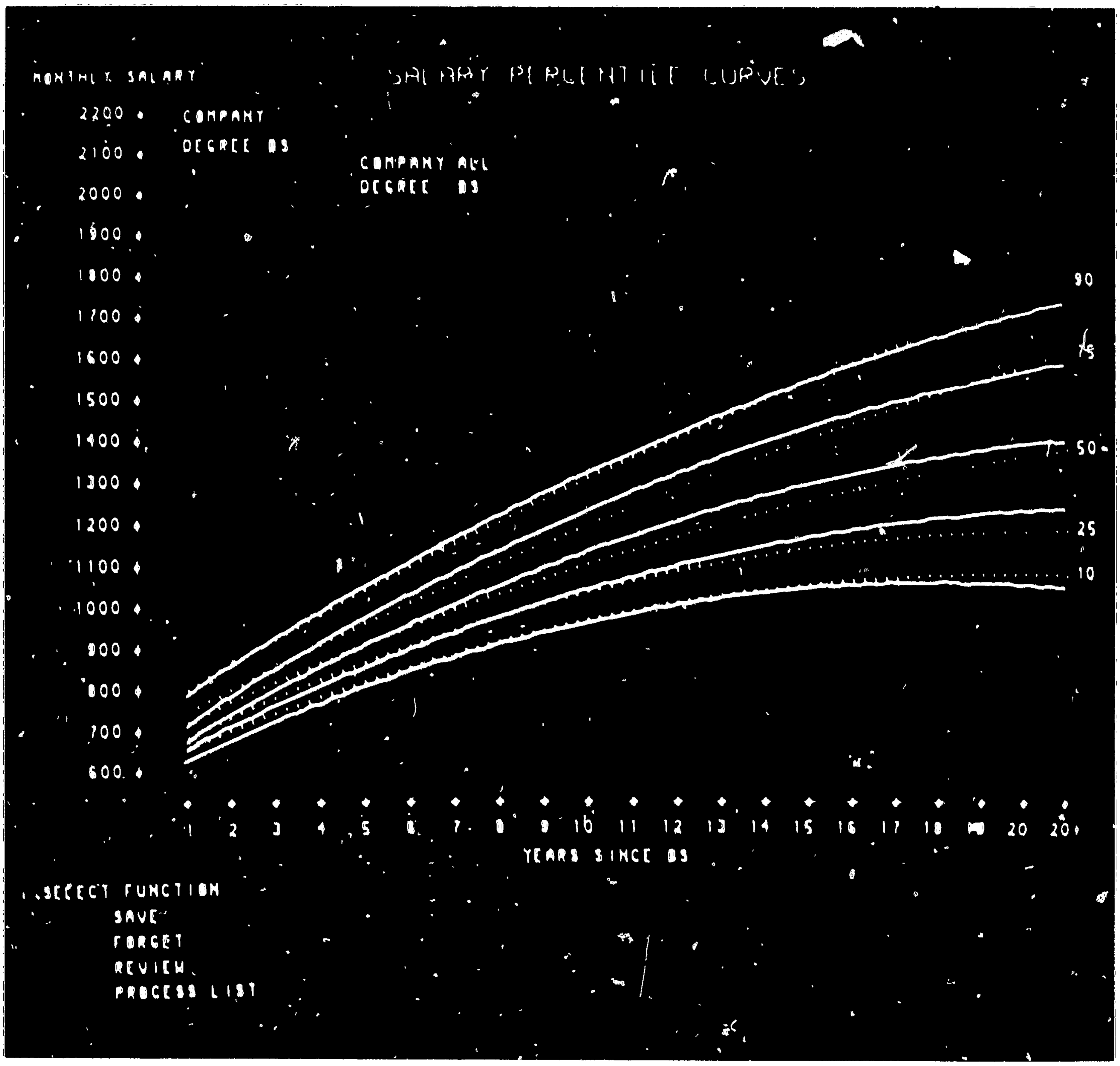


FIGURE 4-2. SALARY CURVES GENERATED BY GPDS

processes were designed to eliminate an excess of options and to present the user with a display quickly. An example of this synthesis is the new process REGRESS, which enables the user to draw a curve or line of regression by answering only two queries from the system. While the original design of GPDS provided the flexibility and contained all the building blocks necessary to construct a process having the full capabilities of REGRESS, it was only after REGRESS became a standard process that meaningful use of GPDS for graphics occurred. (This particular process was used in the analysis of time-sharing statistics by means of GPDS.)

Solving the specific problem of salary curves proved extremely valuable in the research into data management displays. It is expected that the initial display capability within TDMS will benefit from this work.

Another positive result achieved in the GPDS research was the discovery of the power and flexibility of the process-building concept. Very successful demonstrations have been given in which a process tailored especially to the spectator's interest was built on-line. The ease of tailoring the interactive queries to the user's particular needs proved valuable. The ability of GPDS to be molded into a suitable vehicle for salary curve generation demonstrates the versatility provided within the original design.

Plans

Work on GPDS has ceased. Support will be provided to maintain the program system for its present users. Currently, salary administration personnel are using GPDS for new salary curve generation. All future display research will be performed within the context of TDMS.

Project Documentation

1. Shaw, M. A. User's manual for GOR: A GDPS program for structuring TSS-LUCID produced data bases. SDC document TM-2302/000/01. February 24, 1966. 6 pp.

2. Bowman, S. and Lickhalter, R. A. Planning guide for GPDS. SDC document TM-2889. March 15, 1966. 5 pp.

3. Guillebeaux, C. A. and Grant, E. E. The general-purpose display system (GPDS) users' manual--Part I. SDC document TM-2722/001/00. March 22, 1966. 118 pp.

4. Grant, E. E. The general-purpose display system (GPDS) users' manual--Part II, Sections 1-3. SDC document TM-2722/002/00. April 25, 1966. 19 pp.

5. Grant, E. E. General-purpose display system (GPDS) users' manual--Part II, Section 4. SDC document TM-2722/003/00. May 9, 1966. 7 pp.

6. Vorhaus, A. H. General purpose display system. Datamation, 1966, 59-64.

TSS-LUCID*

R. E. Bleier, Project Leader
W. A. Cozier — J. D. Reynolds
W. C. Dennis — R. A. Lickhalter
P. A. DeSimone — M. A. Shaw

Description

TSS-LUCID (Time-Shared Language Used to Communicate Information System Design) is a time-shared data management system operating on the Q-32. It is general-purpose in design and capable of processing many types of data. Output is available as fact retrieval on the teletype and a minimal form of a line-printed report. The system allows the user to describe the names of the data base elements and the data base structure in an easy-to-use nonprogramming language. The data base is created when the user submits his data input values to be loaded. After the data are successfully loaded, they are so organized that the user can easily query and update them on-line.

The chief virtues of TSS-LUCID are its simplicity of user operation, application to variable-length names (as well as numerics), easy-to-learn and logically structured query language,

*Supported by the Advanced Research Projects Agency.

and rapid retrieval. On the negative side, the data base structure allowed within TSS-LUCID is essentially a single level with provision for multiple values for elements; the query language does not provide arithmetic capability; teletype outputs provide only minimal formatting capabilities.

The experience gained from using TSS-LUCID has been applied to the design of TDMS (see p.4-9).

Progress

TSS-LUCID achieved checkout status in the early part of the report period. The system then began to be extensively demonstrated to visitors and SDC personnel. User interest increased, and soon users with differing data problems from within SDC, government agencies, and private companies began using it, both at SDC and remotely, to perform their data management functions. A sample list of organizational users of TSS-LUCID includes:

. Lackland AFB for personnel research.

. USAF Headquarters at the Pentagon for USAF Research Project Management.

. SDC Salary Administration for the Salary Information Retrieval System.

. SDC Technology Directorate for TSS accounting information.

. SDC administration for project information on corporate planning.

In supporting these users and responding to their needs, various improvements have been made in the system. During this period, the staff accomplished the following:

. Published user's manuals for all components in the system.

. Developed a program that allows a user to organize a TSS-LUCID data base on any series of elements in the data base. It also alphabetizes all name-type element values in the data base.

. Developed a program that will merge identically described data bases.

. Developed a program that will output a symbolic file in TSS-LUCID input format from a TSS-LUCID data base that has been updated. In addition, this program will provide printed output on the line printer or teletype in blocked format.

. Modified the fact retrieval program to retrieve on entry numbers, to order the output on either the ascending or descending order of a numeric element's values, to retrieve on two new conditional relations (EXISTS and FAILS), and to reduce the amount of user-required inputs through the commands REPEAT and SAME.

. Modified the load program to provide more pertinent error detection support and to allow for on-line input on data values.

Plans

The work on TSS-LUCID is complete. Support of the users of the system will continue.

Project Documentation

1. DeSimone, P. A. The TSS-LUCID reactive translator (LUTRAN) user's manual. SDC document TM-2844/000/00. January 26, 1966. 10 pp.
2. Dennis, W. C. ORNZ user's manual. SDC document TM-2826/000/00. January 27, 1966. 4 pp.
3. Cozier, W. A. and Dennis, W. C. QUUP user's manual. SDC document TM-2711/000/01. February 28, 1966. 21 pp.
4. Reynolds, J. D. TSS-LUCID LOADER user's manual. SDC document TM-2936/000/00. May 13, 1966. 12 pp.
5. Dennis, W. C. User's manual for MERG. SDC document TM-2880/000/02. June 6, 1966. 4 pp.
6. Dennis, W. C. User's manual for SYMOUT. SDC document TM-2920/000/01. October 4, 1966. 6 pp.

TIME-SHARED DATA MANAGEMENT SYSTEM*

R. E. Bleier, Project Leader
P. R. Bartram
K. M. Brandon
S. H. Brown
V. L. Cohen
W. A. Cozier
W. C. Dennis
P. A. DeSimone
C. A. Kribs
R. A. Lickhalter
J. D. Reynolds
R. B. Reynolds
S. Shapiro
M. A. Shaw
U. Michel (Atlantic-Richfield)
W. D. Williams (Shell Oil Company)

Description

The Time-Shared Data Management System (TDMS) is a general-purpose data management system providing nonprogrammer users the necessary tools to handle large volumes of structured data. Designed for operation on the IBM S/360 series of computers, TDMS represents a significant gain in capability over TSS-LUCID, particularly in the areas of user-oriented language, improvements in data structure (i.e., nested repeating groups of data), and the inclusion of new functions performing data base element manipulation and report generation.

TDMS provides the user with a set of integrated programs that he can use to solve a wide variety of data management problems.

Use of TDMS is expected to reduce initial costs of creating the software required for new data processing applications, as well as reducing the time required to implement new data handling systems.

System flow within TDMS is shown in Figure 4-3. The major components included in TDMS are listed below.

. DEFINE permits the creation of new data base descriptions and alteration of existing data base descriptions. The description includes, for each component of the data base, the name, date or numeric definition, and legality specifications. Capacity has been increased to 1023 components over the 128 elements that TSS-LUCID can handle.

. LOAD accepts values either from tape, disc or interactive console. All types of numeric and nonnumeric data are accepted and legality-checked. Errors can be corrected on-line. Inputs can be segmented to permit loading subsets of the total data base.

. QUERY interprets the interactive language required to specify the data retrieval instructions. QUERY provides the ability to request single values, multiple values, or entire entries, printed either on-line or off-line, in a standard tabular format. Full arithmetic capabilities, plus operators such as sum, minimum, maximum, count, average and standard deviation, are provided.

. UPDATE permits changing of existing data values and adding or deleting single data values or entire entries. The updating will occur dynamically and the user will have immediate use of the new information.

. COMPOSE allows the user to design a report format and specify the data components to be printed, employing a user-oriented language. This program is highly interactive and includes arithmetic, ordering, formatting, and tutorial capabilities. It operates in an update mode so that previous report descriptions can be easily modified.

. PRODUCE produces reports on either the on-line interactive console or line printer, as directed. The primary inputs are the output of COMPOSE and the designated data base.

. DISPLAY provides graphical presentation of data on the CRT. A limited set of standard display formats will be allowed including bar charts (both vertical and horizontal), scatter plots, regression curves, pie charts, and connected line segments. The program will automatically establish the relationships

*Supported in part by the Advanced Research Projects Agency.

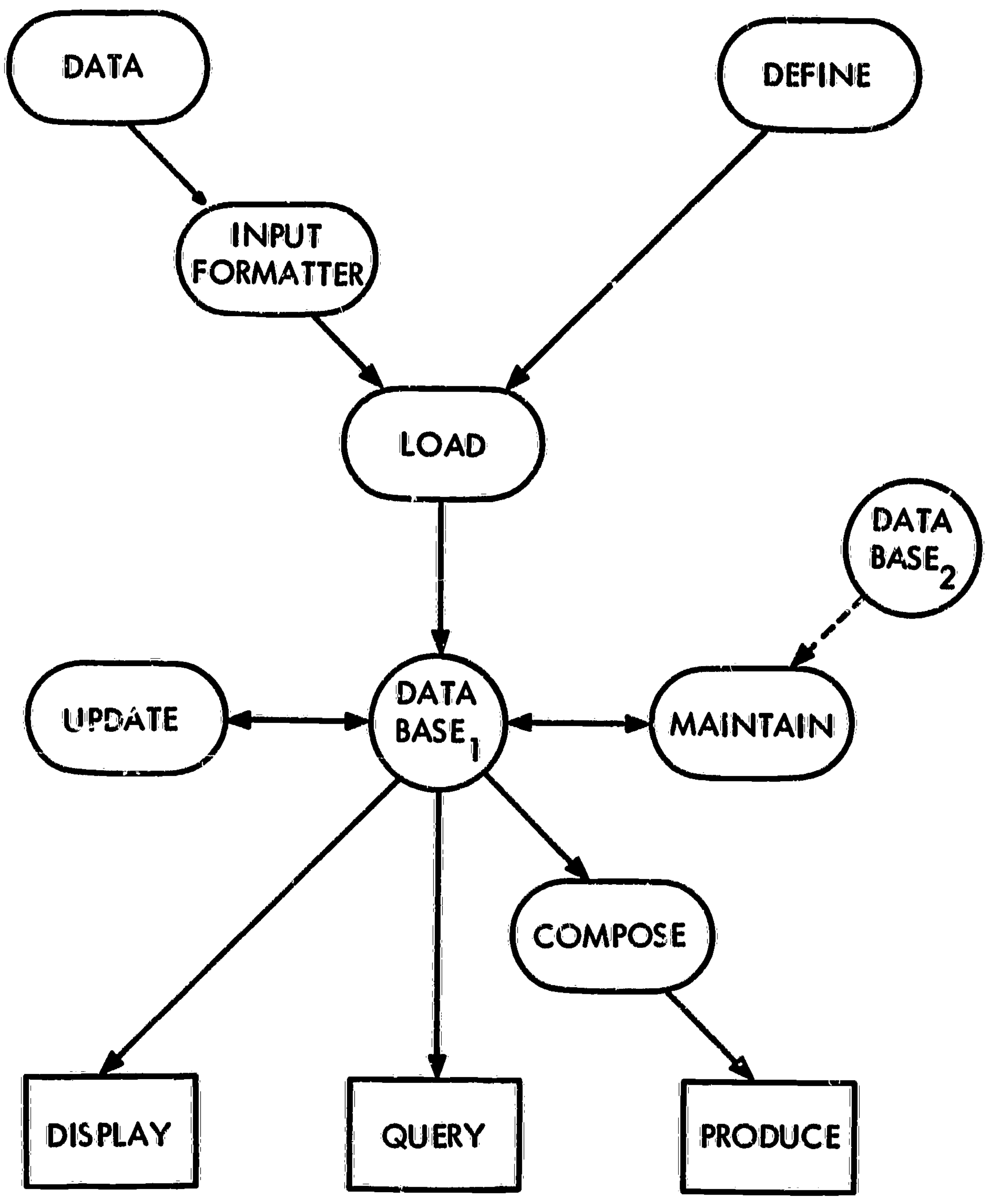


FIGURE 4-3. TDMS FLOW DIAGRAM

between the values to be plotted and the CRT surface, axis labeling and graduation. In all cases, the user will be able to override the program's selection.

. MAINTAIN provides for merging, subsetting, extracting, ordering, restructuring, and updating data bases. The program accepts one or two different data bases, an on-line description of the desired output data base, rules for selection of data, and the transformations that are required. Instructions can be saved, modified, and reused.

. TDMS INPUT FORMATTER provides a comprehensive capability to convert existing data values to the numbered-fields/data-set format required by TDMS. The inputs to this process are a description of the existing input symbolic tape and a description of the transformations desired. The input formatter then transforms the input data to the new format as required by LOAD.

. TSS-LUCID TO TDMS DATA BASE CONVERTER is being written for the Q-32 to provide conversion of existing TSS-LUCID d .a bases to TDMS format. The primary aims of this task are to facilitate checkout as parts of the TDMS system become available and to provide continuity for TSS-LUCID users.

Progress

TDMS design, coding, and checkout are under way. TDMS language development has progressed across the entire system with very careful attention being paid to consistency, simplicity, and power. Additional language was developed to handle the data retrieval scope designations created by the inclusion of nested repeating groups. The intent of the language is to maintain the attractive simplicity of TSS-LUCID but to provide the necessary increase in power required by the more complex data structures allowed.

A central table design was created to efficiently utilize the 360 computer and provide the necessary items for program communication.

Coding work has progressed with substantial portions of the central input/output, retrieval package, and TSS-LUCID-to-TDMS converter completed. Currently, the 360/65 is providing usable results for program checkout.

Plans

Implementation of the TDMS design will be achieved for both the IBM 360/65 and IBM 360/50. Research will continue to provide additional capabilities within TDMS.

Project Documentation

1. Franks, E. W. A data management system for time-shared file-processing using a cross-index file and self-defining entries. SDC document SP-2248. April 21, 1966. 17 pp.

2. Cohen, V. L. The time-shared data management system (TDMS) for the IBM 360/65. SDC document TM-2745/000/02. October 3, 1966. 4 pp.

METHODS FOR HANDLING ENGLISH TEXT WITHIN TDMS*

J. Farell
E. W. Franks

Description

A joint project was initiated by the Advanced Systems Division and the Research & Technology Division to investigate the possibility of incorporating text processing within TDMS. Such a capability would be of value to those interested in processing documents that contain both structured and unstructured (free-text) data. One document of this type is the Department of Defense Form 1498, "Research and Technology Resume." This form--which was used in the present study--was designed for use by various government agencies to record information concerning research performed under government contracts.

Entries on Form 1498 concerning the date, number code, security classification, etc., are

*Supported in part by the Advanced Research Projects Agency.

considered to be structured text entries, since their structure is predetermined and limited. For example, the security classification entries are limited to the following: Unclassified, Confidential, Secret, and Top Secret. Free-text entries, on the other hand, may consist of any sequence of words describing a project, agency, results of research, etc. For example, the title of a project could be "Nuclear Magnetic Resonance Studies of the Effect of Pressure on Glass Structure." Such strings of words are viewed by a processing program as free text.

Progress

Initial work was performed on the development of a program that would derive simple statistical techniques for analyzing textual information. This work was discontinued, however, in favor of a more promising approach. Since TSS-LUCID has some of the capabilities proposed for inclusion in TDMS, it was chosen as the experimental vehicle for this study. Changes were made in the TSS-LUCID load program to eliminate the character-size restrictions; changes in the query and translation programs were also made to handle text-defined elements.

Data from a sample of ten DD 1498 forms were loaded. Using the query program within TSS-LUCID, data were successfully retrieved from this data base. An attempt was then made to load a larger data base (containing the information on 294 forms). With this increase in the number of text literals, however, the computation requirements for the load program became too great, and the data base could not be loaded.

Although the experiment demonstrated that there are no basic incompatibilities between fixed-length formats and text-processing techniques, it showed that TSS-LUCID is unsuitable for useful text-handling work. It did, in addition, provide a valuable guide to the preparation of the text-handling specifications for TDMS. These are presently being documented; they are also based in part on techniques that have evolved from experience with SDC's Synthex I system (see p. 5-7).

Plans

Provision has been made in the design of TDMS to permit the later incorporation of a text-handling capability. However, due to present time and manpower limitations, this project may be considerably delayed.

FACT RETRIEVAL FROM TDMS USING NATURAL LANGUAGE*

C. H. Kellogg

Description

Since large-scale data base systems may contain thousands of data elements, the typical user should not be required to learn or remember the precise element names for achieving data retrieval. The principal goal of this project is to develop a program that will assist the user in making retrieval requests by accepting questions of varying generality in a subset of English and by translating these questions into a form acceptable to TDMS.

The overall procedure being followed to achieve this goal may be considered a multistage interactive decision process in which the computer provides varying amounts of information at six levels of processing. These six levels are:

1. Dictionary lookup. Each word in a question is associated with appropriate grammar codes and a series of lexical readings that represent the several meanings of the word. Words not previously defined to the system are displayed to the user who then must either define them or rephrase his question.

2. Syntax analysis. A phrase structure grammar for the English subset is used to recognize and

*Supported in part by the Advanced Research Projects Agency.

parse the question into its constituents. The user is apprised when and how the grammatical patterns of the English subset have been exceeded; when this happens, he must rephrase his question.

3. Semantic interpretation. Lexical readings are selected and combined according to semantic rules similar to the projection rules proposed by Katz, Fodor, and Postal at MIT. An undefined meaning association is indicative of a question that is not relevant to the data base, or a faulty dictionary entry, or both.

4. Query construction. Information from the preceding three levels is used in conjunction with a generative phrase structure grammar of the TDMS query language in order to produce the one or several queries implied in the question.

5. Query presentation. The queries are presented to the user for verification that they represent the information request that he has in mind. If so, he can ask the computer to find the answers to these queries; if not, he can formulate a more general or specific question within the English subset.

6. Answer presentation. Based on the answers he receives, the user may decide to formulate more questions.

Progress

A question-to-query translation program has been constructed for the TSS-LUCID query language. This program performs translation according to a set of internally stored rules and a dictionary that relates the various meanings of English words to their equivalents in the query language.

The translation program has been applied to numerous questions posed in English. User feedback provided by Levels 1 through 4 (above) has been invaluable in "debugging" the rules and dictionary components of the translator. The program currently accepts at least one English question equivalent for most query expression types that may be posed to the experimental data base. Some simple paraphrase-handling ability is evident, e.g., the questions "What is the number of people in Arizona?", "What is the population of the state of Arizona?", and "How many people are there in Arizona?", all receive equivalent query interpretations.

The translator has been programmed in the META/LISP programming language developed at SDC. This language is being used at present to implement a data description program that will interface with TSS-LUCID.

Plans

The project has already shed some interesting light on the relationships between subsets of a natural language and artificial query languages. This knowledge will be used in adapting the translator to efficiently interface with TDMS. To achieve a larger degree of data base independence, the data description program must be completed so that dictionaries may be easily created and updated on-line.

Project Documentation

1. Kellogg, C. H. An approach to the on-line interrogation of structured files of facts using natural language. SDC document SP-2431. April 29, 1966. 86 pp.

EVALUATION OF LIST PROCESSING FOR DATA MANAGEMENT*

R. Bosak
H. L. Howell
B. L. Jones

Description

Since many of the problems encountered in data management involve rearranging a complex file structure, an investigation is being made of the desirability of applying a list processing language to the problems of data management. LISP was chosen as the programming language, since it is the most advanced list processing language

*Supported by the Advanced Research Projects Agency.

available today. TSS-LUCID and TDMS were selected as the data management systems to model.

In 1965, a program (known as CLEAR) was written, modeling essentially the input language and functions of LUCID. This work was done in LISP 1.5 and proved to be suitable only for very small data bases since no secondary storage was used.

Experience gained from CLEAR led--early in 1966--to work on a new program, written in the more powerful LISP 2 language. It was also decided to model TDMS with its more complex data structure.

Progress

Design of the new program for LISP 2 was completed in August 1966. However, several features of LISP 2 that were needed to implement the program were not yet completed. In particular, the restriction on the amount of core storage available for the user's program and data had not been lifted and the design of the top-level ALGOL-like source language--one of the primary advantages of LISP 2 over LISP 1.5--was still subject to change. The new program was therefore temporarily shelved, and work resumed on CLEAR.

CLEAR had been written to process only small, core-resident data bases which were loaded initially via teletype. Modifications were made to allow for disc or tape storage of the data base, and for off-line capability in the define and load functions. CLEAR was also revised to reflect the capabilities of TSS-LUCID.

A three-thousand-entry data base containing TSS operating statistics was loaded and manipulated by CLEAR, with several statistics recorded during these activities. These pointed to areas of the program that could be improved, and to operations in LISP that are particularly time-consuming. These findings led to several iterations of program improvements.

Plans

Work will continue on improving CLEAR; a report of the results of this work is being prepared. Further work using TDMS and LISP 2 will be based on this report.

Project Documentation

1. Bosak, R. CLEAR, an experimental data base management system. SDC document TM-2673. October 11, 1965. 31 pp.

LANGUAGE PROCESSING AND RETRIEVAL

R. F. Simmons, Head
H. Borko, Associate Head

Efforts of the Language Processing and Retrieval staff encompass three main areas of research and technology. These are document retrieval and classification systems; evaluation studies as applied to information systems; and the study of linguistic, semantic, and logical methods for analyzing and processing natural language text.

From the first of these areas, the display-oriented document retrieval system, BOLD, has been developed to a high degree of effectiveness in both its speed of operation and its convenience to users. Expanded experience with a large document collection and an active set of users can be expected to improve the present system. SURF, a personal file retrieval system, has reached the point of providing a convenient service to users. ALCAPP is a newly developed system for automatically classifying very large numbers of documents represented by lists of index terms. During the year, the team of Doyle and Blankenship managed to break through two barriers to effective automatic classification. The first of these was a space problem--how to deal with a matrix of thousands by thousands in size. The second was a cost problem--if classification requires comparison of each document with every other, cost climbs exponentially with the size of the comparison matrix. Solutions to both these problems were devised by an iterative approach to segmenting the comparison space. The effectiveness of the solution was recognized in the selection by the American Documentation Institute (ADI) of the ALCAPP report as a prize-winning paper.

The evaluation of various kinds of document representations was also honored by a prize from ADI, awarded to the principal researcher, R. Katter. This study devised a method for evaluating the effectiveness of titles, abstracts, sets of index terms, and condensations as means of representing the content of documents. The companion study, concerned with methods of measuring the factors that influence judgments of relevance, still continues. So far, a number of factors including user orientation, knowledge level of judges, specificity of document and query, and others have been discovered and partially quantified.

In the language processing area, research continues on stylistic analysis, anaphoric and discourse analysis, and semantic and cognitive approaches to understanding text and answering English questions. Three new projects were inaugurated this year under the sponsorship of the Advanced Research Projects Agency. One is a lexicographic study that will convert Webster's Seventh New Collegiate Dictionary to machine-readable form and will analyze the definitions to discover semantic field relationships among the meanings of English words. This study provides basic data for all other language research. The second study is the development of an on-line system to aid a linguist in testing the consistency and effectiveness of any portion of a transformational grammar he may devise. In view of the ever-increasing complexity of grammars required for language processing, such a system is an essential tool for language researchers. The third study will develop criteria concerning cost, learnability, and error proneness to aid in selecting subsets of natural English for use in question-answering systems. This is an important link from language research to data base systems construction.

In the Synthex research on understanding natural language text and answering questions, a significant synthesis of linguistic, semantic, and logical approaches to question answering can be seen. First, an effective, interactive syntactic-analysis system incorporating pronoun-antecedent logic from the anaphoric and discourse analysis project has been developed. Second, a clear statement of the roles of syntactic, semantic, and logical processing of language has been embodied in the design of Protosynthex III, which is conceived as the first fairly complete approach to a natural language processor.

Taking a large view of this staff's activities, it is possible to observe that work on document retrieval, automatic classification of documents, and evaluation studies has moved in each case very close to the point of practical applications. In contrast, the language processing research has on the whole moved toward more basic studies of how information is encoded into natural languages. Such studies as those concerned with stylistic analysis, machine-aided translation, and automatic extracting achieved logical stopping points from our point of view, either by having used existing knowledge to its present limits or by having been superseded by projects more basic to our language processing interests.

For the future, it can be predicted that the staff's interest in document retrieval, classification, and the evaluation of information systems will show a definite bent in the direction of applications to practical systems for handling documents or libraries for the Corporation and its customers. The language processing effort can be expected to move slowly toward such a position after some years more of basic research in linguistics, and semantic and logical analysis of natural languages.

LANGUAGE PROCESSING

Our program of research in language processing aims toward the eventual goals of providing capabilities for (1) using subsets of natural English as a query or command language for computers, and (2) enabling computers to read, understand and generate English text. These goals are recognized to be difficult and distant ones that necessitate the development of basic syntactic, semantic and lexicographic knowledge and tools.

At the syntactic level, the transformation tester project is a tool system which augments a linguist's capability to produce useful and internally consistent grammars. Other syntactic subprojects are found in the anaphoric and discourse analysis research and in the PLPII syntactic analyzer.

Semantic research is emphasized in a project of synonymy and semantic classification, in the Synthex research on question answering and linguistic inference, and in work on discourse analysis.

At the base of all these studies is the recognition that a great deal of linguistic and semantic information has been accumulated by lexicographers in standard dictionaries. Making a dictionary machine-readable and preparing tools for applying the information to other tasks is yet another major project of this staff.

Two studies of a more applied nature, stylistic analysis and the study of query subsets, use linguistic knowledge to provide useful tools on the one hand for the comparison of text styles and on the other for guiding the selection of English subsets for use in data base query systems. These studies are initial links in the chain from language research to useful technology.

Lexicographic Studies*

J. C. Olney
P. Ziff, Consultant at SDC (University of Wisconsin)
} Principal Investigators
D. L. Dwiggins

Description

The overall aim of this project is to enlist the aid of computers in describing the vocabulary structures of natural languages. It has long been recognized that word senses reciprocally delimit each other within a language. Thus, in carrying out a semantic analysis of a given word (or, more properly, a morpheme) in some language, it is important to take account of all the other words in that language that are significantly related semantically to the given word. However, to prepare an exhaustive list of these other words by hand is an almost hopeless task, at least for any of the major languages, because of the sheer size of their vocabularies. In principle, no such list could be considered exhaustive unless every word in the language had been considered for inclusion or rejection with respect to the given word. In addition, there are problems in deciding exactly what should count as a significant semantic relation between a pair of words.

We are immediately concerned with the vocabulary structure of English as manifested in Webster's Seventh New Collegiate Dictionary, henceforth referred to as "W7." We intend to develop a program system that can be used in either an automatic or an interactive, semiautomatic mode to specify the semantic field(s) of any randomly chosen word by processing a Hollerith transcription of W7. For each sense of the word a semantic field will be specified by locating the sense within a network of other word senses having specified relationships to the given sense (and to each other). These relationships will comprise

*Supported by the Advanced Research Projects Agency.

(approximate) synonymy, antonymy, and any of various field relationships (e.g., the "have a" relationship holding within each of many sets of terms referring to the body parts of animals). The search for other word senses so related to the given word sense will involve taking account of the W7 sense descriptions in which the given word occurs (unless it is a function word like "the" or "of") and those in which one or more words occurring in the description of the given word sense also occur. Because the routines for establishing links between word senses will often link through the wrong sense of a word occurring in a sense description under analysis, the semiautomatic mode of operation of the program system will probably be the more useful one, at least initially.

In addition to facilitating the formal semantic analysis of English, our program system may well prove invaluable to lexicographers for the preparation of new dictionaries. An application of the system to the analysis of discourse structure is described on page 5-5.

Progress

A set of encoding rules for transcribing the lexical portion of W7 (pp. 1-1041) onto magnetic tape has been specified, and a contract for the transcription has been negotiated in which 99.5 percent accuracy in the keyboarding of characters is guaranteed. This means that as many as 65,000 characters may be transcribed incorrectly. However, by taking advantage of the rigid format of the dictionary entries, we expect to be able to detect more than 90 percent of these errors under program control. Moreover, most of the routines that will detect transcription errors would have to be written even if the transcription were known to be error-free, as for example, the entry parser, which recognizes and separates the various parts of each entry, such as the pronunciation, the etymology, the sense descriptions, etc.

To evaluate and improve the performance of our projected program system in specifying the semantic fields of particular word senses, it is necessary to develop criteria for judging whether or not a given pair of word senses stand in a significant semantic relationship. On approach to this problem that we are pursuing is to examine the various productive and semiproductive patterns of sense development exhibited both among senses defined for the same word and among those defined for morphologically related words. One such pattern is exemplified in the relationship between the sense of "flush" defined in W7 as "to take wing suddenly" and the sense defined as "to cause (a bird) to flush." Our working hypothesis is that most of what one could reasonably consider to be significant semantic relationships will be represented among the (semi-)productive patterns of sense development. Many of these patterns are easy to find, being implicit in the senses described for the various affixes listed as main entries in W7.

A complementary approach we are pursuing is to provide conceptual analyses of the meanings of nonfunction words that are used very frequently in W7 definitions, starting with those chiefly used in defining derived senses (e.g., "cause" in the second definition given above for "flush"). We will use these analyses to develop precise terminology for describing the semantic relationships represented by the patterns discovered. Finally, for each pair of word senses chosen as an exemplar of a pattern, we have specified the syntactic features appropriate to the uses of the word(s) in those senses. In this way we hope to isolate that element (if any) of the sense shift which corresponds to a shift in syntactic function.

Plans

As soon as W7 has been transcribed, we shall process it on the Q-32 computer with a modified version of the VAPS Indexer (a word-by-word indexer constructed for Synthex) to obtain an index that lists for each word the sense descriptions in which it occurs. With this index and a routine for stripping off inflectional suffixes, we could already start tracing semantic fields. However, a hand simulation of this processing indicated that, unless the morphemic components of the words were taken into account, the scope of the search for semantically related words would be too narrow, and that syntactic information about the sense descriptions would be needed to avoid gross errors in tracing fields. We plan therefore to develop routines for carrying out a rather detailed morphemic analysis of words occurring in W7 sense descriptions and a syntactic analysis sufficient to determine at least their parts of speech in most cases. In analyzing the words morphemically extensive use will be made of the syllabic and etymological information provided in W7. The availability of detailed morphemic analyses and the stylistic rigidities of W7 definitions will greatly simplify the task of analyzing the sense descriptions syntactically, but undoubtedly we will sometimes fail to resolve part-of-speech ambiguities.

Project Documentation

1. Ziff, P. More on understanding understanding utterances. SDC document SP-2504. June 8, 1966. 16 pp.

Anaphoric and Discourse Analysis*

J. C. Olney, Principal Investigator
D. L. Londe

Description

An expression is anaphoric if it replaces another expression occurring in the same text. Generally, third-person pronouns and possessive adjectives, such as "he" and "his," are anaphoric expressions, as are many other kinds of pronouns. However, in English scientific and technical writing the most prevalent type of anaphoric expression is exemplified by the following sequence: "Several experiments with college students were performed. The results were encouraging." "The results" is anaphoric; it stands for an expression that does not actually occur in the text but can be constructed from one that does, namely, "several experiments with college students."

Anaphoric expressions play a crucial role in promoting information transfer across sentence boundaries and between parts of complex sentences. In particular, they serve to abbreviate preceding passages, enabling writers and speakers to add to what they have just said without repeating all of it. In English scientific and technical articles about 90 percent of the sentences contain at least one anaphoric expression.

A machine capability for handling anaphoric expressions is needed for many applications of language data processing, especially those requiring analysis of the specific information content of natural text, e.g., fact retrieval and autoabstracting. Machine handling of anaphoric expressions can be divided into three phases: (1) recognizing anaphoric expressions; (2) finding the appropriate antecedent for each anaphoric expression, that is, either the expression that it replaces or the one from which the replaced expression can be constructed; (3) indicating how the replaced expression can be integrated into the sentence in which the anaphoric expression occurs, so as to eliminate the anaphora.

Progress

So far rules have been formulated and programmed for the first two phases only. These rules form three separate packages (boxes 3, 4, and 5 in Figure 5-1). Together with two language processing

*Supported in part by the Public Health Service.

programs developed by other researchers (boxes 1 and 2) they constitute a program system that yields as output not only an anaphoric analysis of the input text but also an analysis of relations of discourse equivalence that hold among expressions occurring in the text (see (5) below). A brief description of each component of this system follows.

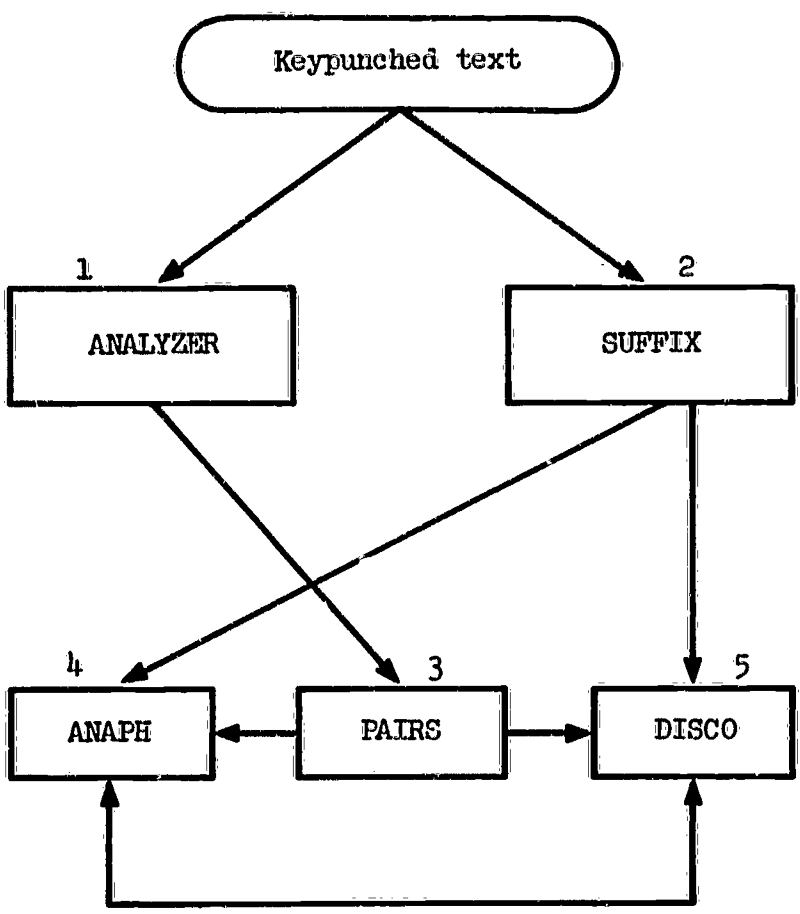


FIGURE 5-1. FLOW DIAGRAM OF A PROGRAM SYSTEM FOR OBTAINING ANAPHORIC AND DISCOURSE ANALYSES OF ENGLISH TEXT

1. ANALYZER is a large syntactic analysis program for English developed by Kuno and Oettinger of Harvard.

2. SUFFIX is a program developed by Sedelow and Ruggles that groups together word forms that are spelled the same except for their suffixes (see p. 5-17). SUFFIX will soon be superseded by a more powerful morphemic analysis routine to be developed on the Lexicographic studies (see p. 5-3). Ultimately the entire program system being developed on that project for tracing semantic fields will be added in box 2, to provide semantic information about all the words in the text under analysis as inputs to ANAPH and DISCO.

3. PAIRS processes syntactic analyses output by ANALYZER to make fully explicit the various kinds of structural relationships indicated in the analyses, in all cases where this can be done deterministically. The running time of PAIRS, a SNOBOL III program, has been reduced two-thirds by reprogramming, but it is still rather slow, averaging about 40 seconds per analysis.

4. ANAPH uses the output of PAIRS and SUFFIX, together with special dictionary information relating to pronoun replacement, to identify potential antecedents of certain types of anaphoric expressions. For each pair consisting of an anaphoric expression and one of its potential antecedents, ANAPH obtains an evaluation score by checking whether the pair satisfies any of several relevant attributes. For the relatively small amount of text on which ANAPH has been run so far, its accuracy in recognizing anaphoric expressions has been about 90 percent, but only about 50 percent in preferring the true antecedents of these expressions.

5. DISCO uses the same inputs as ANAPH and yields a discourse analysis of the text. The immediate goal of discourse analysis is to achieve a grouping of the expressions in a connected discourse that will approximately correspond to a grouping of the entities they denote or express, such that those entities closely related in the conceptual framework of the discourse are grouped together. This implies that a successful discourse analysis will include, as special cases, the grouping together of each anaphoric expression with its appropriate antecedent. The grouping together of expressions as "discourse equivalent" is based exclusively on certain kinds of morphemic and syntactic parallels that the expressions exhibit.

Almost all of the routines used by DISCO to establish equivalences between single words occurring within the same paragraph have been checked out; remaining to be checked out are the routines that establish phrase equivalences and evaluate both types of equivalence. A recent run of the checked-out routines on a 750-word scientific text yielded 279 equivalences, of which more than 85 percent were intuitively acceptable.

Rules have been formulated but not yet programmed to extend ANAPH and DISCO in various ways, e.g., to seek and evaluate potential antecedents for additional types of anaphoric expressions, to establish equivalences between expressions occurring in comparative constructions, and to extend equivalences across paragraph boundaries.

Plans

We have so far confined our attention to the microstructure of text, as constituted by relations of discourse equivalence (including anaphoric substitution) between single words and short phrases. Building on recent work by Christensen of USC and Becker of the University of Michigan, we shall attempt to formulate rules for tracing the macrostructure of text, as constituted by relationships among clauses and sentences within and across paragraph boundaries. Macrostructure analyses of texts should prove mutually reinforcing with microstructure analyses. In particular, certain anaphoric expressions, e.g., 'however' when used as a sentential adverb, cannot be linked to an appropriate preceding text segment without information about interclause and intersentence relationships in the text in which they occur.

We also plan to experiment with the use of deep-structure syntactic analyses as an alternative to the essentially surface-structure analyses currently obtained from ANALYZER and PAIRS. The deep-structure analyses will have to be provided manually since no deep-structure analyzer capable of processing scientific and technical texts has been developed as yet.

Project Documentation

1. Olney, J. C. and Londe, D. L. A research plan for investigating English discourse structure, with particular attention to anaphoric relationships. SDC document TM(L)-3256. November 22, 1966. 17 pp.

Synthex: The Computer Synthesis of Language Behavior*

R. F. Simmons, Principal Investigator
J. F. Burger H. Manelowitz
R. E. Long

Consultant: M. Garvin, University of California, Los Angeles

Description

Synthex research is aimed toward developing an understanding of those semantic and cognitive processes that are required for synthesizing complex language behavior on computers. A long-term goal of the project is to produce such useful language processors as question-answering and essay-writing programs.

Various experimental programs make up the context of this research. Protosynthex I uses an indexing logic and a statistical treatment of semantic and syntactic weights for words and word combinations to retrieve textual material relevant to a question.

PLPII (deriving from the McConlogue Pattern Learning Parser) abstracts structural information from a dependency-analyzed text to build automatically a grammar that generalizes to similar text. Protosynthex II integrates lines of syntactic and semantic research from previous years into a system that accepts a wide range of English structures, and reduces them first to syntactic kernels and then, via a semantic analysis, into formal language kernels that are acceptable as input to a meaning structure of objects connected by relations. This structure

*Supported in part by the Advanced Research Projects Agency.

is well enough understood to allow several mathematical and logical inference procedures to be applied to question answering.

Progress

Protosynthex I. This text retrieval system, including a root form logic, the maximal indexing principle, and an algorithm for maintaining statistical control of word ordering, has been translated to SDC's S/360 Time-Sharing System. Final checkout as an effective text retrieval system awaits further development of the S/360 software. This effort is expected to be completed in the early part of the coming year, at which time Protosynthex I will become a shelf product for use in demonstrations and selected text retrieval applications.

PLPII. The second Pattern Learning Parser is an outgrowth of our experience with the McConlogue Parser. Testing the latter system on a wide range of English sentences revealed great strengths in its ability to predict word-classes for unknown words, provided they occurred in familiar contexts. Its greatest weakness was discovered to be an inadequacy in its treatment of phrases and clauses. Although it could learn to parse a sentence of indefinite complexity, it could not generalize its recognition of nested phrase and clause structures beyond very simple limits.

The analysis led naturally to the design of PLPII. PLPII accepts an ordinary English sentence such as the following:

The man for whom the bell tolls is dead.

If no data exist in the dictionary of grammar, the user is required to indicate word-class codes and dependency relations as in the following strings:

```
Art N Prep Rpron ART N V  V ADJ .
N   V   V   *PREP N  V*N  *  *V .
```

The abbreviations are the usual notation for article, noun, verb, preposition, relative pronoun, and adjective. An asterisk attached to a symbol means that the governor occurs earlier in the sentence; an asterisk by itself indicates the head verb of the sentence. The first string is simply an assignment of word-class symbols to each word in the sentence. The second string is the indication of what part of speech in this sentence is the governor for each word immediately above.

From this information, the system is able to build syntactic frames for each word that it experiences. For example, from the above sentence, for the words "bell" and "tolls" the system builds the following frames:

```
BELL:   ART N V V
TOLLS:  N V V  V
```

For each word, the context frame shows the surrounding word-classes and the word-class of its governor. As a result of experience with each word in several sentences, the system develops several context frames for each.

From this information, from additional general phrase structure rules of syntactic combination, and with the use of an ordered pushdown stack, PLPII is able to develop a grammar sufficient to analyze a very large subset of natural English.

The analysis is furnished in two forms: as a dependency structure and as an immediate constituent tree. Multiple analyses are permitted, and interaction with the user allows for selection of a single analysis or correction of mistakes made by the system. In a sentence containing unknown words, the system can correctly assign word-classes to one or two unknown words in succession. While not thoroughly tested, PLPII appears to be highly satisfactory in its ability to accumulate a grammar sufficient to analyze a larger range of English structures than we have ever before managed.

A certain type of transformational logic is required to obtain immediate constituent (IC) analyses from many kinds of sentences. These rules actually reorder the English string to bring immediate constituents into adjacency with each other.

A most recent addition to the system was a program embodying rules developed by Olney and Londe for finding antecedents to common pronouns and possessive adjectives (see p. 5-5).

The resulting IC analysis, with antecedents substituted for pronouns, is used as a basis for producing a set of nested kernels that correspond approximately to the deep structure of the sentence. Figure 5-2 shows the dependency analysis, the IC tree, and the resulting kernels for an example sentence.

Protosynthex II & III. Protosynthex II was a question-answering system based on a specialized conceptual dictionary. It included a grammar and a syntactic analysis procedure sufficient to account for a narrow range of English sentences. It successfully answered a limited range of English questions and offered an extremely limited capability to generate childish essays in response to a "describe" command.

Protosynthex II was useful in showing the need for a definite phase of semantic analysis and for a conceptual structure that would allow fairly sophisticated inference procedures. Experiments with it resulted in enough information to permit the design of Protosynthex III.

Protosynthex III is designed as the first breadboard model of a complete language processing system. Six components are seen to comprise the system:

1. Syntactic analysis of text or question into a set of nested English kernels representing the relational structure of the statement.

2. Semantic analysis and transformation of English kernels into a formal language of objects and relations.

3. Construction of a cognitive structure (or semantic net) representing meanings in a context of general knowledge of the world.

4. Question answering through the use of deductive and inductive logic applied to the cognitive structure.

5. Semantic synthesis from the cognitive structure into English kernel sentences.

6. Syntactic synthesis of kernel sentences into more complex English forms.

The two most critical problems in the construction of this system are the design of the cognitive structure and the development of means for accomplishing a semantic analysis and transformation of English kernel sentences into that structure. The cognitive structure makes a clear distinction between words of English and the representation of their meanings of various contexts. The structure is composed of objects and relationships each of which can map in a many-many fashion onto English words. The relationships may be simple and well understood such as "HASPART," "SUBSET," "SUPERSET," etc., where properties of the relation such as reflexivity, transitivity and symmetry are known and useful for inference. A relation may be simple but its properties may be incompletely known as in such examples as "SIZE," "SHAPE," "QUALITY," etc. Finally, a relation may be complex as "RUN," "HIT," "REMAIN," "SEE," etc. Theoretically a complex relation should be resolvable into an ordered set of simple relations. If this conjecture is true, a great deal of understanding must still be gained to prove and apply it.

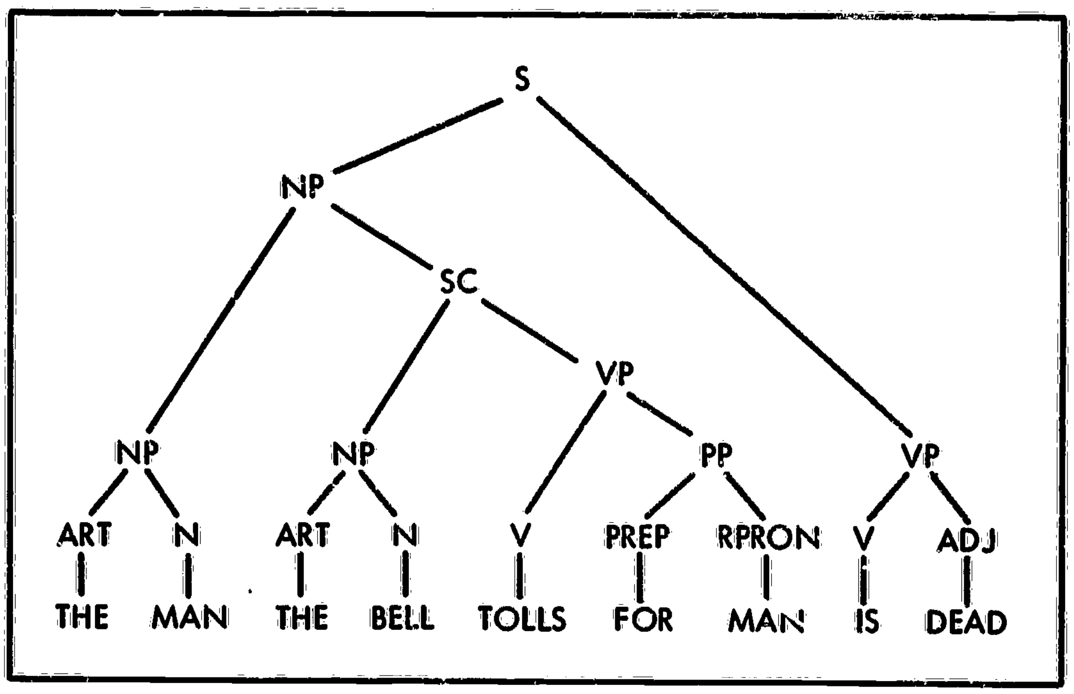
In using a cognitive model of this type, answering a question requires a relational match of the objects and relations of the question to the data of the model. A simpler structural match would require only that the exact structure of the question be discovered in the cognitive model. The relational match requires that the objects of the question map onto each other through acceptable relational paths to form an answer. Thus, if the question is asked, "Did the play please John?" and a structure in the model includes the idea that "John Smith enjoyed a dramatic performance," the question-answering system must discover that John Smith is an instance of John, that a dramatic performance is equivalent to a play,

THE MAN FOR WHOM THE BELL TOLLS IS DEAD.

Input Sentence

((1 THE ART MAN 2)
(2 MAN N IS 8)
(3 FOR PREP TOLLS 7)
(4 WHOM RPRON FOR 3)
(5 THE ART BELL 6)
(6 BELL N TOLLS 7)
(7 TOLLS V MAN 2)
(8 IS V * 0)
(9 DEAD ADJ IS 8))

Dependency Analysis for the Sentence



Display Scope Presentation of IC Analysis for the Sentence

(((MAN EQ ((BELL (TOLLS FOR MAN) ****) IS DEF)) IS DEF) IS DEAD)

(a)

1 (2 IS DEAD)
2 (3 IS DEF)
3 (MAN EQ 4)
4 (5 IS DEF)
5 (BELL 6 ****)
6 (TOLLS FOR MAN)

(b)

Kernel Structure for the Sentence in (a) Nested Form and (b) Expanded Form

FIGURE 5-2. PLPII ANALYSIS OF A SENTENCE and that X enjoyed Y is an equivalent inverse relation of Y pleased X.

Semantic analysis and transformation of English kernels into the formal language representing the model is seen to be a specialized form of question answering. For each element X of an English kernel, the question is asked, "What are the supersets of X?" More than one superset is usually found for each term of the kernel, and several possible interpretations usually result. For example the kernel "Pitcher struck umpire" results in the following expansions:

1. man hit man
2. man caused-work-stoppage man
3. man rich-find man
4. container hit man
5. container caused-work-stoppage man
6. container rich-find man

Each member of the expanded set becomes a question, and, if discovered to be true with respect to the cognitive structure, is considered to be a possible interpretation. Since the kernels of a sentence are nested, a kernel that may prove possibly true in isolation is often discovered to be impossible in context. For example, in the context, "The angry pitcher struck the umpire with a baseball bat," only the interpretations of pitcher as a man and strike as a hit are possible.

So far in Protosynthex III, the syntactic component, PLPII, is fairly complete. Programs that produce the cognitive structure and answer questions from it have been produced but will probably require many changes in the future. Program design is in progress for the semantic analysis component. No work has begun on either the semantic synthesis or syntactic synthesis components. The system as it so far exists is written in LISP 1.5 on the Q-32 TSS system.

Plans

Protosynthex I will be finally checked out on the IBM S/360 and made available as a useful product. No further work is planned on that system. Major effort will be devoted to the development of Protosynthex III as an experimental natural language processing system. A paper entitled "A Theory of Verbal Understanding" is in preparation.

Project Documentation

1. Simmons, R. F. Automated language processing. In C. A. Cuadra (Ed.) Annual review of information science and technology. New York: Interscience, 1966.
2. Simmons, R. F. Storage and retrieval of aspects of meaning in directed graph structures. Communications of the ACM, 1966, 9(3), 211-215. (Also available as SDC document SP-1975/001/02.)
3. Simmons, R. F., Burger, J. F., and Long, R. E. An approach toward answering English questions from text. AFIPS Conference Proceedings, 1966, 29, 357-363. (Also available as SDC document short-form SP-2445.)
4. Burger, J. F., Long, R. E., and Simmons, R. F. An interactive system for computing dependencies, phrase structures and kernels. SDC document short-form SP-2454. April 28, 1966. 28 pp.
5. Simmons, R. F. Natural language processing. Datamation, 1966, 12(6), 61-72.

Transformational Grammar Tester*

D. L. Londe, Principal Investigator
W. J. Schoene

Description

This project is constructing a system of computer programs that will assist linguists in building and validating transformational generative grammars. The system is designed for operation on a time-shared computer and seeks to make full use of the advantages of man-machine interaction in a situation that allows the user virtually immediate access to the computer.

Chomsky defines a generative grammar as one that "attempts to characterize in the most neutral possible terms the knowledge of the language that provides the basis for actual use of language by a speaker-hearer."** It is "a system of rules

*Supported by the Advanced Research Projects Agency.
**N. Chomsky. Aspects of the Theory of Syntax, 1965.

that in some explicit and well defined way assigns structural descriptions to sentences."*

The syntactic component of such a grammar specifies the well-formed strings of formatives (minimal syntactically functioning elements) in the language and assigns structure to them. Transformational grammars are built on the concept of the separation of two types of structure. Accordingly, the syntactic component of a transformational grammar is made up of two sets of rules: phrase structure rules, which generate a deep structure; and transformations, which determine the ultimate surface structure of a sentence. The deep structure is operated on by the semantic component of the grammar to determine meaning. The surface structure is interpreted by the phonological component.

The emphasis on explicitness is a distinct advantage of the generative grammars. However, it imposes a great burden on the linguist. For example, determining the applicable rules in the derivation of a particular sentence is an extremely tedious and time-consuming task, difficult to perform with accuracy. And, as the grammar becomes large (as the linguist attempts to account for more phenomena in the language he is describing), it becomes more difficult to discover all of the interrelations of the rules.

In constructing a grammar with the transformational grammar tester (TGT) developed under this project, the linguist will be asking the same questions and employing many of the same procedures that he would have used were the tester not available. With the tester, those functions that are amenable to programming will be available for operation by the linguist using a teletype and display scope, on which devices he will also receive computer output.

The most important tasks to be performed by the tester center around the ability to execute grammar rules. The linguist will be able to use the computer to determine the applicability of transformations and to execute them and have their results displayed quickly and accurately. Many ancillary functions will be provided for specification and manipulation of rules and test structures. Combinations of these functions will greatly aid the linguist in the very important job of determining the implications of changes in the rule set. For example, the linguist can save entire derivations of sentences that he considers correct. He may then insert a new rule, or change or delete and existing one, and then have the computer apply the rule set to the base structures of the saved sentences. The computer will then display any changes in their derivations.

Progress

The following general capabilities are currently available as part of the first stage of TGT, which operates on the Q-32 Time-Sharing System:

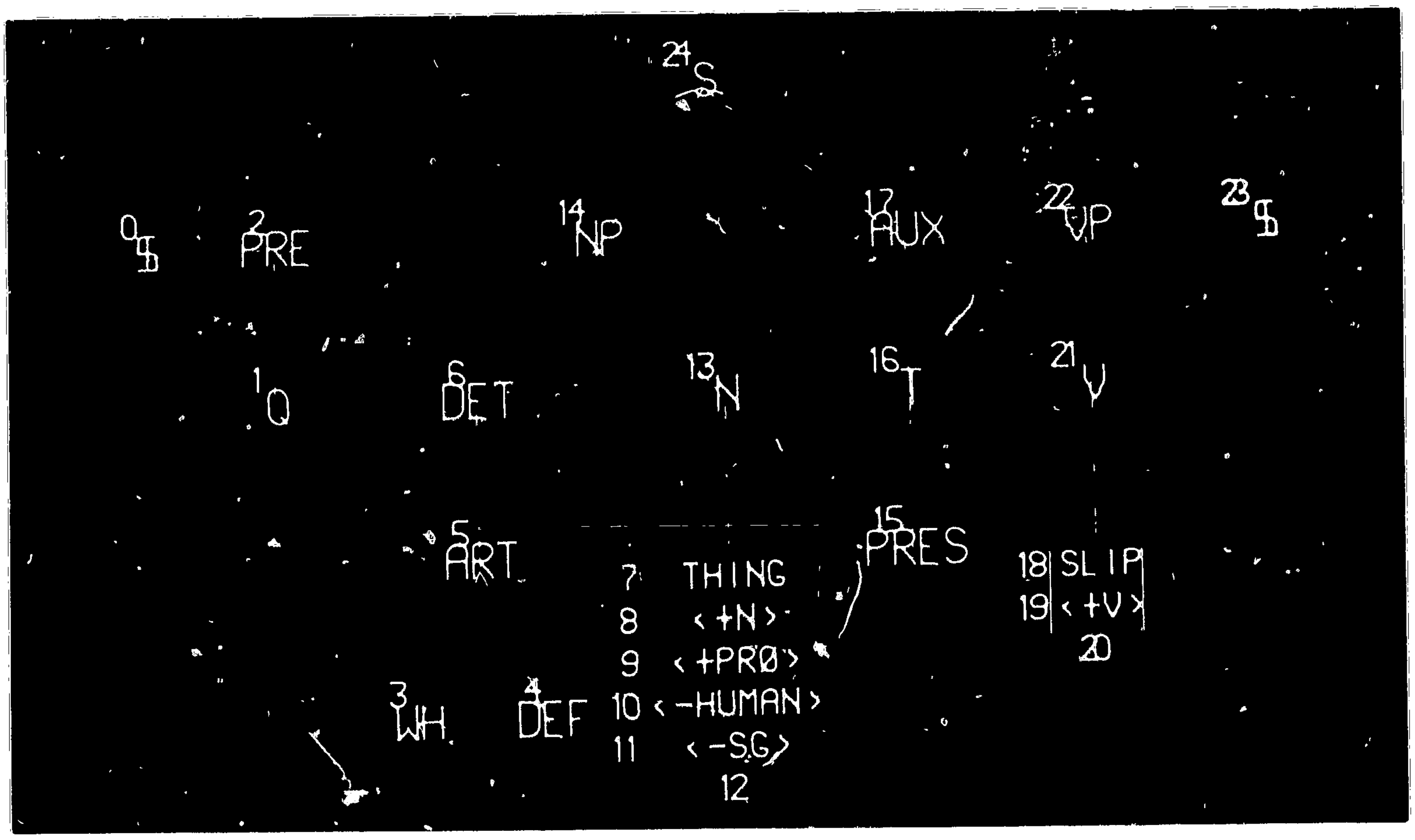
1. Create and display tree structures (Figure 5-3).
2. Modify tree structures (e.g., adjunction, erasure).
3. Name and save trees and subtrees.
4. Determine the applicability of individual transformational rules to specified tree structures.
5. Input new grammar rules and change existing ones.

The rules on which the system is currently being tested are those of the IBM Core Grammar (developed by Dr. Peter Rosenbaum of IBM) as are the conventions for rule interpretation.

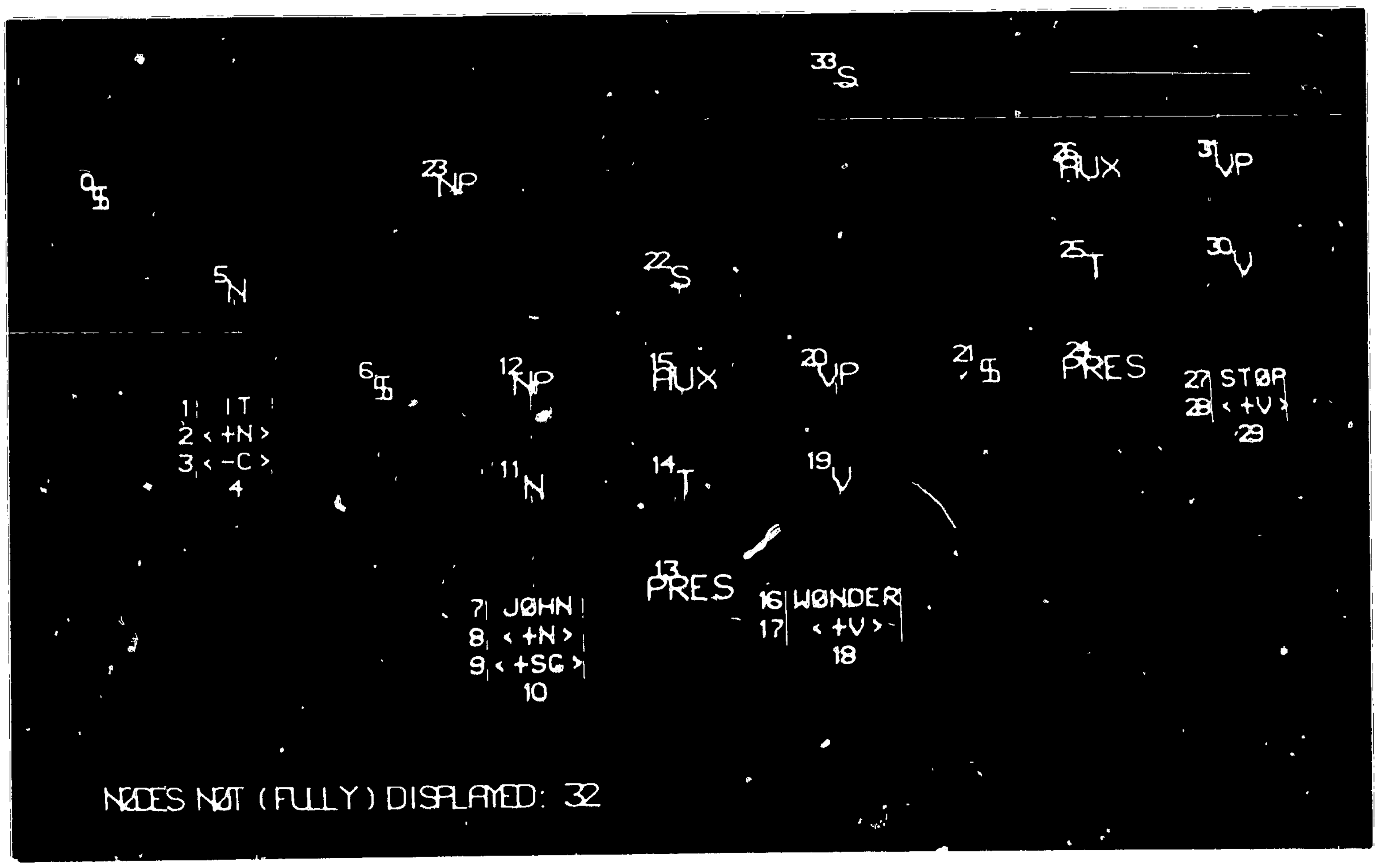
Plans

Our immediate plans include: improvement of the existing system, particularly by the discovery of fast-fail rules for transformation interpretation (rules that will quickly determine the inapplicability of certain transformations to certain structures); the programming of algorithms that will enable the user to apply the entire set of transformations to an initially defined structure, with capability to handle optional rules; and the

*N. Chomsky. Aspects of the Theory of Syntax, 1965.



A typical tree display generated by the TGT program.
The tree represents the sentence "Which things slip?"



A more complex tree representing the sentence "John stops wondering." Note that the linguist is informed which parts of the tree do not fit on the scope using the current spacing parameters.

FIGURE 5-3. TREE STRUCTURES GENERATED BY TGT extension of the above to allow the user to save derivations and compare the output of new rule sets with all or specified subsets of the saved structures.

We will soon modify the tester for use by the Air Force-UCLA Syntax project, a project to integrate all of the transformationally oriented work done thus far on English syntax. This entails changing the conventions for rule interpretation. We hope to construct the algorithms in a way that will allow even high-level changes to be made with minimum difficulty. We shall also be concerned with determining and providing the computer support needed for the construction of a grammar by a group of linguists. General areas of study will be the problems of intercommunication (the sort of problem that occurs when many users of the same data base are allowed to modify it) and of retrieving data from the grammar.

Synonymy and Semantic Classification

Karen Needham, Consultant at SDC
(Cambridge Language Research Unit, England)

Description

This research is concerned with the role of a semantic classification in natural-language discourse analysis. It is based on the hypothesis that some classification of words by their meanings is required if text is to be semantically analyzed or interpreted. This hypothesis thus involves theoretical studies of the way in which such a classification could be used for semantic analysis, and of the kind of classification that might be required for this purpose; and it involves practical work on constructing suitable classifications and testing them. The problems that arise in any attempt to tackle these extremely difficult questions are naturally considerable, but there is equally no doubt that automated language processing will not get very far unless this attempt is made.

Progress

Some progress has been made in understanding two problems of primary interest. These are: (1) what are the different ways in which classificatory information about a word may be used to determine the semantic structure of a text; and (2) what kinds of information have to be given in the dictionary entry for a word if the right answers are to be obtained when it is so used.

The use of a semantic classification in discourse analysis has been primarily advocated as a means for selecting the correct sense of an ambiguous word, the argument being that we can do this by referring to the conceptual character of the surrounding text. The justification for this argument is that we are more likely to find the information we need to make the selection if we look for any word expressing the appropriate concept, rather than for some specific word. However, this selection procedure depends on some knowledge of the kinds of things that may be said in terms of standard conceptual combinations, or "message forms"; and successful discourse analysis must therefore depend on knowing what these are, and on how the specific message forms characterizing a particular text are identified. It is suggested that this conceptual organization of a text is closely associated with its semantic structure, and that recognizing the semantic structure of a text is as essential as recognizing its syntactic structure. It seems indeed that we can distinguish three levels in the semantic structure of text in general. These range from immediate relationships between words, through intersentential connections, to the themes or topics of whole paragraphs or sections. Permitted conceptual combinations are, moreover, involved in all of these, since particular word groups, say of an adjective like "green" and a noun like "jug," depend on them, as do the intersentential connections determining the thread of an argument. Such connections may be provided by the occurrence of a near-synonym in one sentence for an expression in another, or the subsuming of a whole piece of discourse under a theme saying what the text is basically about. The semantic structure of a text is obviously connected with its syntactic structure, but cannot be correlated with it in a straightforward way, since it also depends on our knowledge of the relations between ideas, and of the concepts associated with particular words. Yet it is apparent that some identification of the semantic structure of a piece of discourse is required for a correct analysis of particular words or word formations in it.

This being the case, it is evident that any classification of words for discourse analysis must take into account the conceptual combinations under which a word may be subsumed. But the difficulty is that the reasons why a word may be so classified do not seem to have much to do with the word's meaning as we might attempt to define it in a fairly rigorous way. Thus, the basic problem of constructing a classification is to obtain groups of words whose membership is based not on strict definition, but on the actual occurrences or behavior of words in text, for example. Some work has therefore been done on suitable methods for grouping words, using information about their occurrences or co-occurrences in text, or about their behavior in standard dictionary entries, since these can be taken as summaries of large amounts of text material.

Plans

Further investigation of these problems is expected to continue, both at Cambridge Language Research Unit and at SDC, with special emphasis on types of classification and procedures for obtaining them.

Query Subset Studies*

R. L. Weis, Principal Investigator
R. V. Katter
C. H. Kellogg
R. F. Simmons

Description

The aim of this study is to discover some principles that can guide the selection of subsets of natural English for query languages (query subsets) in future sophisticated data base question-answering systems. The purpose behind the use of natural-English-appearing query subsets is to simplify user training and reduce errors by capitalizing on the linguistic competence of the user. Query subsets may vary in size of lexicon, size and complexity of grammar, number of meanings per word, and what might be called the "freedom of expression" or "paraphrase" within the subset. It is highly unlikely, however, that this list includes all the important dimensions. Consequently, in addition to studying the contribution of these obvious features of language, we are seeking some underlying structure that is psychologically fundamental and that may explain the effects of the obvious variables.

Freedom of expression is a peculiarly important and subtle variable. For instance, it is perfectly feasible to construct a query language in which all the permissible forms are identical to natural English sentences and yet have the rigidity of form and syntax of a formal programming language. Such a rigid subset would completely defeat the purpose of a natural-English-appearing subset. That purpose is to capitalize on the built-in linguistic knowledge of the system user to increase the learnability of a query subset and decrease the errors of use.

The problem, then, is to find customary or preferred patterns of English paraphrasing of natural language users, and to order these in

*Supported by the Advanced Research Projects Agency.

terms of their likelihood (i.e., relative frequency) of occurrence. The system designer can then select from this ordered set only those high-likelihood substitutable forms that are within the size, cost and computing restraints of his system.

The rationale of the study is that frequency of substitution of sentence types is a linear combination of variables, such as syntactic features. Therefore, a matrix composed of the mutual substitution frequencies (confusions) of a set of sentence types can be analyzed to recover measures of the contribution of these variables. The experimental labor involved in gathering such a matrix is prohibitive, however, except for very small sets of sentences. It has been shown that a confusion matrix can be accurately predicted from the psychologically perceived similarity relation between sentences. Therefore, judgment of similarity will be used in place of confusion scores. The relation of perceived similarity will be analyzed to recover underlying variables that govern such behaviors as learning acceptable patterns of response, and making errors in using query subsets. It is hypothesized that the property of similarity will interact with the paraphrase habits and preferences of a user (response biases) to predict the characteristic responses of that user.

The research design includes the following major steps (not necessarily in this order):

1. Obtaining a set of natural English questions relative to a given data base to determine the pattern of response biases in unrestrained questioning of a data base.

2. Producing a model grammar, G_a, that will generate and parse a large variety of English questions. The grammar will most likely take the form of a large context-free grammar with perhaps some limited transformational capacity. The purpose of G_a is to have available an ostensive definition of (a) each of the query subsets to be defined and (b) the syntactic description (parsing) of any question obtained in step 1.

3. Using the grammar G_a to generate a number of sets of semantically and lexically related questions to be used in judging of psychological similarity. These sets will cover the same variety of questions as obtained in step 1. The judgments will be an analyzed for patterns of similarity.

4. Deriving several query-subset languages by selecting subsets of G_a that will accept: (a) the highest likelihood paraphrases (as predicted from the response bias and similarity judgment data); (b) the lowest likelihood set; (c) a set of intermediate likelihood paraphrases, with the additional constraint that the acceptance set include only the syntactically simplest of the likely paraphrases. These languages will be tested for learnability and error proneness in an experimental situation simulating a data base system.

Progress

The design and preparations for an experiment to investigate response biases are near completion. The data base to be used will be derived from a census data base, modified to permit a richer variety of questions. The data base will be presented to the subjects in the form of a very large matrix, with individuals (i.e., persons, cities, counties, etc.) as rows, and attributes (i.e., population, education, etc.) as columns. The actual number or type of attributes will not be defined to the subjects except as these may be alluded to in the information requirements. These information requirements, which serve as the stimuli for questioning, will be situational descriptions in the form of a problem. Subjects will produce a set of English questions, under instructions to obtain all the information called for in the requirement.

Plans

Work on the first phase of the experiment will continue. The model grammar will be constructed. The other experimental phases await results of the response bias information and the formulation of the model grammar.

Stylistic Analysis*

S. Y. Sedelow, Consultant at SDC (University of North Carolina) Principal Investigator
T. L. Ruggles

Description

Stylistic analysis is the recognition of patterns formed in the process of linguistic encoding of information. This project is aimed toward developing algorithms that will enable a computer to perform stylistic analysis--i.e., to make useful discriminations, such as distinguishing one author from another, or one type of natural language text from another. In order to make such discriminations, the salient features in the discourse will have to be discovered. Therefore, such analysis has value not only for detection of change or difference, but also for recognition of the characteristics of a given text, protocol, etc., and would be of use to intelligence officers, psychologists, sociologists, or to any project dealing with both the content and structure of language. Computational stylistic analysis would also lead to algorithms of use in the production of readable (i.e., idiomatic to natural language usage) automatic abstracts and machine translations.

Progress

During the past year, work on the stylistic analysis project has had three emphases: (1) expansion of the programming system; (2) translation and some redesigning of the programming system from the form used for the Philco 2000 to one that is accessible to the IBM S/360 and takes maximum advantage of its facilities; (3) extensive analyses of output.

The VIA (Verbally Indexed Associations) programs now keep more detailed records of word association (or linkage) patterns, which, in turn, lead to more comprehensive output and provide a concordance option to facilitate inspection of the context of major idea-bearing words. MAPTEXT, the program that produces an abstract representation of the text, now includes several statistical summaries of distribution patterns of words designated through VIA; in addition, MAPTEXT provides a graph that condenses the more detailed main MAPTEXT representation.

For the S/360, the programs are being translated initially from a FORTRAN/TAC combination into PL/I. The decision to shift from FORTRAN was determined by the fact that PL/I incorporates many of the features once predicted for a FORTRAN VI, which never appeared. In particular, PL/I greatly facilitates the manipulation of alphabetic data, which is difficult in FORTRAN; nor is the Hollerith constant a problem in PL/I, as it is in FORTRAN. Our plan is to translate VIA and MAPTEXT, incorporating new features of design, into PL/I first. Subsequently, sorts and searches that occupy major time segments in the programs' operation will be broken down into machine language. The first stage of VIA, which produces a sort, index and frequency count, is currently in process of translation. The program has been redesigned to provide for a much more flexible indexing procedure to cope with an extensive range of natural language inputs.

Analyses of output have confirmed estimates of VIA's power for content analysis as well as indicating possible implications for computer analyses of syntax and semantics. For example, in sections analyzed from translations of Soviet Military Strategy, by Sokolovsky, words such as "state," which have both syntactic and semantic ambiguity when taken out of context, exhibited both syntactic and semantic rigidity. Although "state" could be used both as a noun and a verb

*Supported in part by the Office of Naval Research, U. S. Navy.

in the sections studied from Sokolovsky, it almost always filled a nominal function. Even when restricted to the nominal syntactic function, "state" could have several meanings, including condition and government. But, in fact, "state" was used in a semantically rigid way, almost always standing for government. If such rigidities are common in certain types of verbal data, it is likely that economies in the analysis of the data can be effected. When a semantic and syntactic classification is once achieved, it should be possible to reduce greatly the number of later classification tests on identical words.

Plans

The translation of all present program modules of VIA and MAPTEXT into a form accessible to the IBM S/360 will be completed. Work directed toward automating the one manual search connected with VIA will be continued. This work entails experimenting with thesauri, thesauri combinations, thesauri-context combinations (some experiments in this area were implemented this year), as well as extended work on syntax and semantics.

So far as semantics is concerned, we want to begin using VIA's linked-list output as input to a program that will produce analyses of semantic networks; we are also interested in the Needhams' procedures and the Doyle-Blankenship approaches to cluster analysis. In the syntactic area, we are most interested in the probabilistic approach exemplified by the Simmons-McConlogue and WISSYN parsers.

For MAPTEXT, we hope to experiment with Markov chain analysis with a view toward increasing the range of MAPTEXT's statistical performance.

Project Documentation

1. Sedelow, S. Y. Updating of THESAUR program. SDC document TM-1908/009/00. December 17, 1965. 50 pp.
2. Sedelow, S. Y. Stylistic analysis: Report of the second year of research. SDC document TM-1908/200/00. March 1, 1966. 137 pp.
3. Sedelow, S. Y. A preface to computational stylistics. In Jacob Leed (Ed.), The Computer & Literary Style. Kent, Ohio: Kent State University Press, 1966, 1-13.

DOCUMENT RETRIEVAL

Automatic Classification*

D. A. Blankenship
L. B. Doyle
F. B. Tierney

Description

This project is directed toward the development of systems for automatic classification and for the generation of labeled hierarchies, both based on the use of word-content similarity of documents. It is believed that automatic classification will be an extremely useful part of any information retrieval system.

In the past, most effort has been directed toward the purely technical problem of finding a workable classification algorithm for libraries of one million to ten million documents. Such an algorithm, in order to be practical, must use computer time in direct proportion to the number of items processed, rather than in proportion to the square or cube of the number of items. The ALCAPP system developed by this project includes algorithms that satisfy this condition, and attention is now being directed toward improving and generalizing the system and testing the adequacy and usefulness of the generated classification.

The process of building up hierarchies occurs in three steps. First, lists of content words (the basic representation of the documents) are derived. This selection process may be done manually, or automatically from machine-readable text. Next, the collection of word lists is partitioned into many smaller categories by an iterative cluster-finding program. In the last step, the one- to

*Supported in part by Rome Air Development Center.

two-hundred documents in each of the resulting clusters are hierarchically grouped using any of a number of available programs. If desirable, it is also possible to group the clusters themselves in a hierarchy, thus forming a complete "tree" representing the entire library.

The cluster-finding technique begins by partitioning the document word lists into some number of groups, the exact number being chosen by the experimenter. The initial partitioning may be either completely arbitrary, or based on some hypothesis about what subjects are represented in the data. A frequency count is then made of all the words in each of the groups. Each word within a given group is assigned a weight based on its rank frequency; the most frequent word gets a weight of, say, 63, the next most frequent, 62, and so on for each frequency rank.

The resulting lists of weighted words are called profiles, and serve as the basis for a new partitioning of the word lists. A "score" is determined for the relation of each word list to each profile by obtaining, for the words that co-occur on a list and on a profile, the sum of their rank-frequency weights on the profile. The list is assigned to the highest-scoring profile. This new partitioning is used to form new profiles. The entire process of profile formation and repartitioning on the basis of scores is iterated until no list changes assignment from one cycle to the next.

The hierarchical grouping program (HGP) being used at present starts with weighted or unweighted word lists for some number, N, of objects (which could be either documents or document clusters). A similarity coefficient is computed for each pair of lists, based on some function of the number of words the lists have in common. The resulting N x N matrix of coefficients is scanned, and the word lists of the two objects having the greatest similarity are combined. New similarity coefficients are computed for the N - 1 objects that remain. The whole process of computing similarities and combining word lists is repeated until only one object remains.

HGP contains a labeling routine that tags each compound object formed with a number of words drawn from its word list. The label words are selected according to their tendency to occur frequently in the object, but infrequently in other objects. Thus the selected words are both discriminating and representative.

Progress

Both of the programs described above (i.e., ALCAPP and HGP) were written this year, and are now operational as one complete program package. The system is written in JTS and operates on the Q-32 time-shared computer.

The conception and design of the iterative cluster-finding technique, accomplished early this year, represents a major technological breakthrough. Extensive experimentation was done to determine that the iteration process does terminate--that the condition can be met that no list change cluster assignment from one cycle to the next. Once the fact of termination had been empirically demonstrated from a number of initial partitionings, it became possible to examine the quality of the generated clusters as a function of the topical density of the lists being classified.

"Topical density" is a means of expressing how closely related a set of documents are, on the average. For example, a set of 1000 documents, all on the subject of information retrieval, is much more topically dense than, say, 1000 documents consisting of 10 subgroups of 100 documents each, the documents of each subgroup being variously drawn from one field, such as physics, philosophy, modern art, religion, etc. In general, it was found that at low to moderately high densities, virtually complete separation of the subgroups is obtained.

At very high densities, represented in our experiments by a corpus of 419 document word-lists pertaining to computer applications, it is much more difficult to find nonjudgmental criteria for determining the accuracy of classification. It appears, however, that the generated clusters are perfectly acceptable on an intuitive basis; furthermore, the clusters generated from different initial starting points are quite consistent with one another, indicating good classification stability in a situation where classification accuracy is at best debatable.

Plans

All the programs presently running on the Q-32 Time-Sharing System will be rewritten for SDC's S/360 Time-Sharing System. At the same time, major design changes will be implemented in the cluster-finding program. The new program will be able to classify between twenty and forty thousand documents, and will be "open ended," in the sense that only minor changes will be required to allow much larger document sets to be classified. The present program can handle at most one thousand documents. In addition, the new program will have extended parameter controls that should allow a more complete automation of the cluster-finding process; at present the experimenter must make decisions about certain program parameter settings at each iteration of the cycle.

A series of experiments is planned to determine the value of automatically generated classification in an information retrieval environment. In particular, the experiments should show the relative effectiveness of coordinate and associative indexing retrieval techniques. It is conjectured that some combination of the two approaches will prove most useful in a real-user situation.

Project Documentation

1. Doyle, L. B. Breaking the cost barrier in automatic classification. SDC document SP-2516. July 1, 1966. 62 pp.

2. Doyle, L. B. Perpetual user studies, a prerequisite for management of information on a national scale. Datamation, 1966, 12(10), 28-30. (Also available as SDC document SP-2105.)

3. Doyle, L. B., and Blankenship, D. A. Technical advances in automatic classification. Proceedings, 29th annual meeting, American Documentation Institute. New York: John Wiley & Sons, 1966.

BOLD: Bibliographic On-Line Display

H. Borko, Principal Investigator
H. P. Burnaugh

Description

BOLD was designed to be a highly automated and versatile document storage and retrieval system. The objective was not simply to automate, but through automation, to achieve a high degree of operational efficiency and user satisfaction.

As computers are used for more complex operations, and as the number of scientific documents and their users increase rapidly, the idea of an automated library becomes increasingly attractive. Retrieval time becomes critical, data collections large, and search requirements unspecific. Moreover, users are often located far from document centers and remote access to the library's information becomes an essential element. Hence, a system that uses the speed and capacity of a powerful computer system, coupled with the use of communication that is found in a conventional library, would seem to be the answer to the problems of increasing documentation and inaccessibility of information. The feasibility of such an approach is demonstrated in BOLD.

In order to use the BOLD system, one logs in at an inquiry station consisting of a teletypewriter and a CRT display scope (Figure 5-4). The system can be operated without the scope although the displays do add flexibility.) After the desired data base and program tapes are loaded and the system is operating, an initial tutoring display appears on the scope. This display defines the ten light-pen actions, and their teletype equivalents, which are available to the operator, and

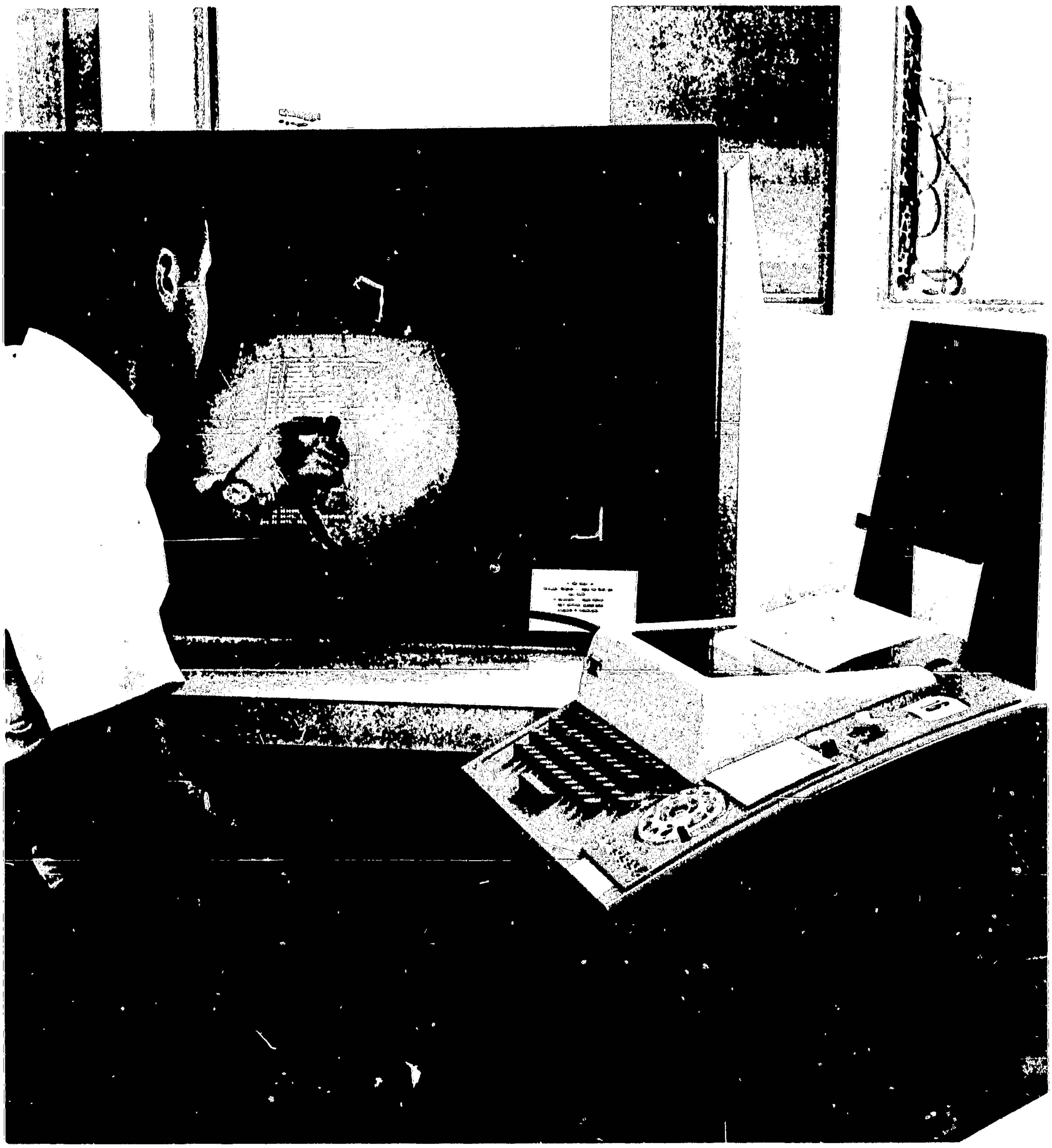


FIGURE 5-4. BOLD INQUIRY STATION

provides all the essential instruction for using the system. The user does not have to be a programmer or receive intensive training before he can retrieve information.

BOLD provides the user with two retrieval strategies for locating the information he seeks. In the BROWSE mode, the user is shown a listing of the major categories into which the documents have been organized (Figure 5-5). He can request a further breakdown of the divisions and subdivisions within any category. The new display will also inform him of the number of documents contained therein. At any time he may elect to browse through various attributes of a category, e.g., author, title, index terms, or abstract. In the last-mentioned case the first abstract in the category will be displayed (Figure 5-6). When he has finished reading this abstract, he can, by means of the Continue action, request the next one, and so on, browsing through all the abstracts in the category.

The SEARCH mode of operation is commonly used when one is seeking information on a specific topic. The user interrogates the internally stored dictionary to find which words could be used as index terms for his retrieval purposes. For example, he may begin by asking whether SPACESHIPS is an index term. The system responds that, in addition to SPACESHIPS, there are a number of similar terms that are also usable index words. These related terms are found by dividing the query word in half and locating all index terms that start with the same combination of letters. Invariably some of the words the user tries are not index terms, but as a result of this dialog, he finds enough that are.

With the information obtained, the user can now formulate a search request. He does so by selecting a number of terms and combining them by the Boolean connectors <u>and</u>, <u>or</u>, and <u>not</u>. The system will search the entire data base and display the identification number of all documents that meet this criterion (Figure 5-7).

Before requesting copies of the documents that have been indexed by the selected terms, the user may request and have displayed the titles, authors, contract numbers, abstracts, or other information that may help him to determine the relevance of the retrieved documents or to prepare a bibliography (Figure 5-8). If a permanent record is desired, it may be obtained on-line by transferring the display to the teletypewriter or to magnetic tape for off-line printing.

<u>Progress</u>

During the past year, we have completed the programming of BOLD and have been improving the displays and interactive aspects in order to make it more responsive and easier to use. The programs have also been modified so that the system can be used without the scope facilities, thus making the system available to remote users.

<u>Plans</u>

Plans call for increasing the capabilities of the system by adding many new features such as automatic classification, automatic indexing, and selective dissemination. It will be reprogrammed for operation on the IBM S/360.

BOLD will also be used as a research tool for studying and comparing the effectiveness of various subsystem configurations. Specifically, it will be used by the staff to study: (1) the distribution of index terms as related to their retrieval effectiveness; (2) the retrieval effectiveness of manually versus automatically derived index terms; (3) the effect of delayed system response versus an interactive response on information search strategy.

<u>Project Documentation</u>

1. Borko, H. and Burnaugh, H. P. Interactive displays for document retrieval. <u>Information Display</u>, 1966, <u>3</u> (5), 47-90. (Also available as SDC document SP-2557.)

2. Burnaugh, H. P. Data base generator for the BOLD system. SDC document TM-2306/001/02. January 10, 1966.

3. Burnaugh, H. P. The BOLD (bibliographic on-line display) system. SDC document SP-2338/000/01. April 6, 1966. 23 pp.

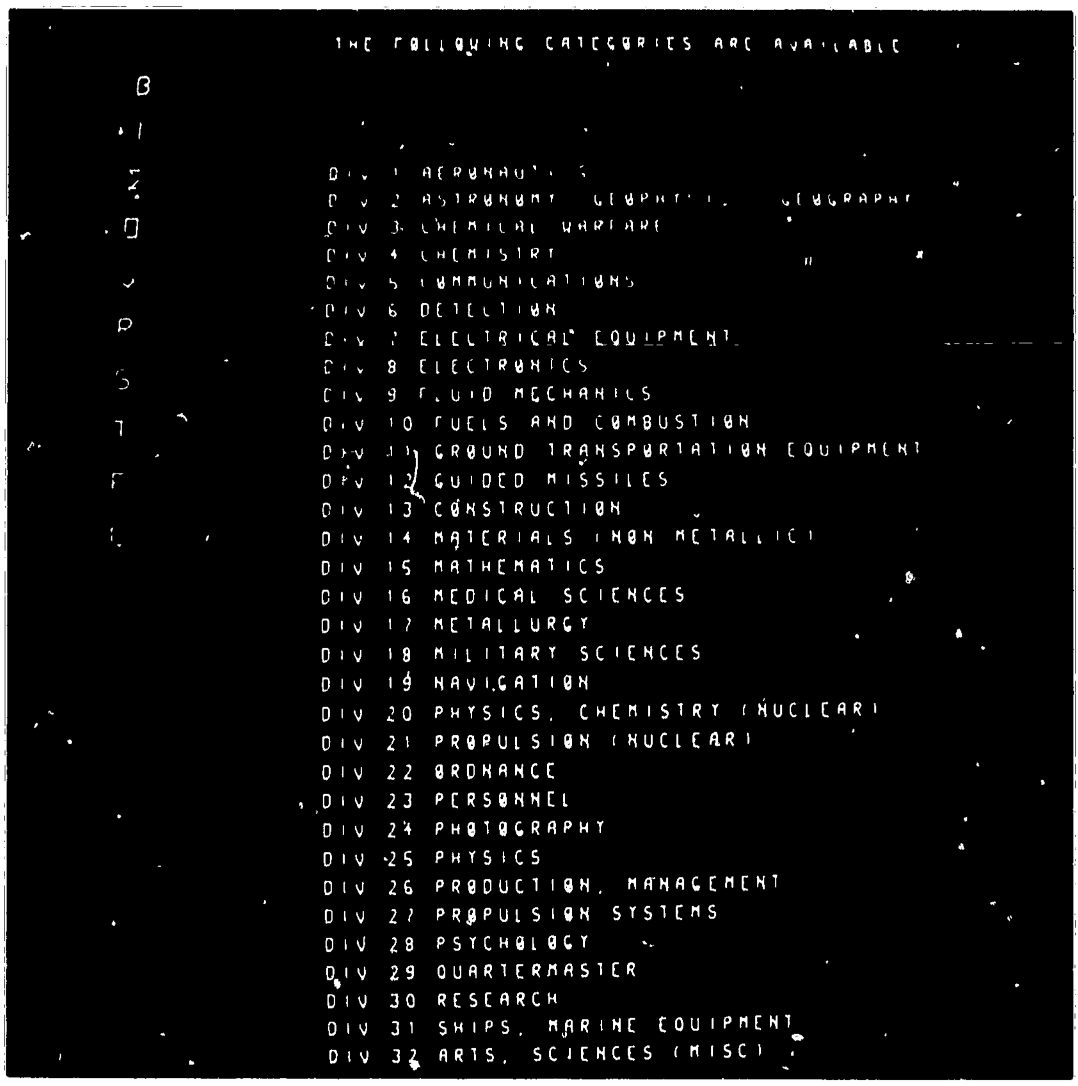


FIGURE 5-5. DISPLAY OF CLASSIFICATION CATEGORIES FOR A COLLECTION STORED IN BOLD

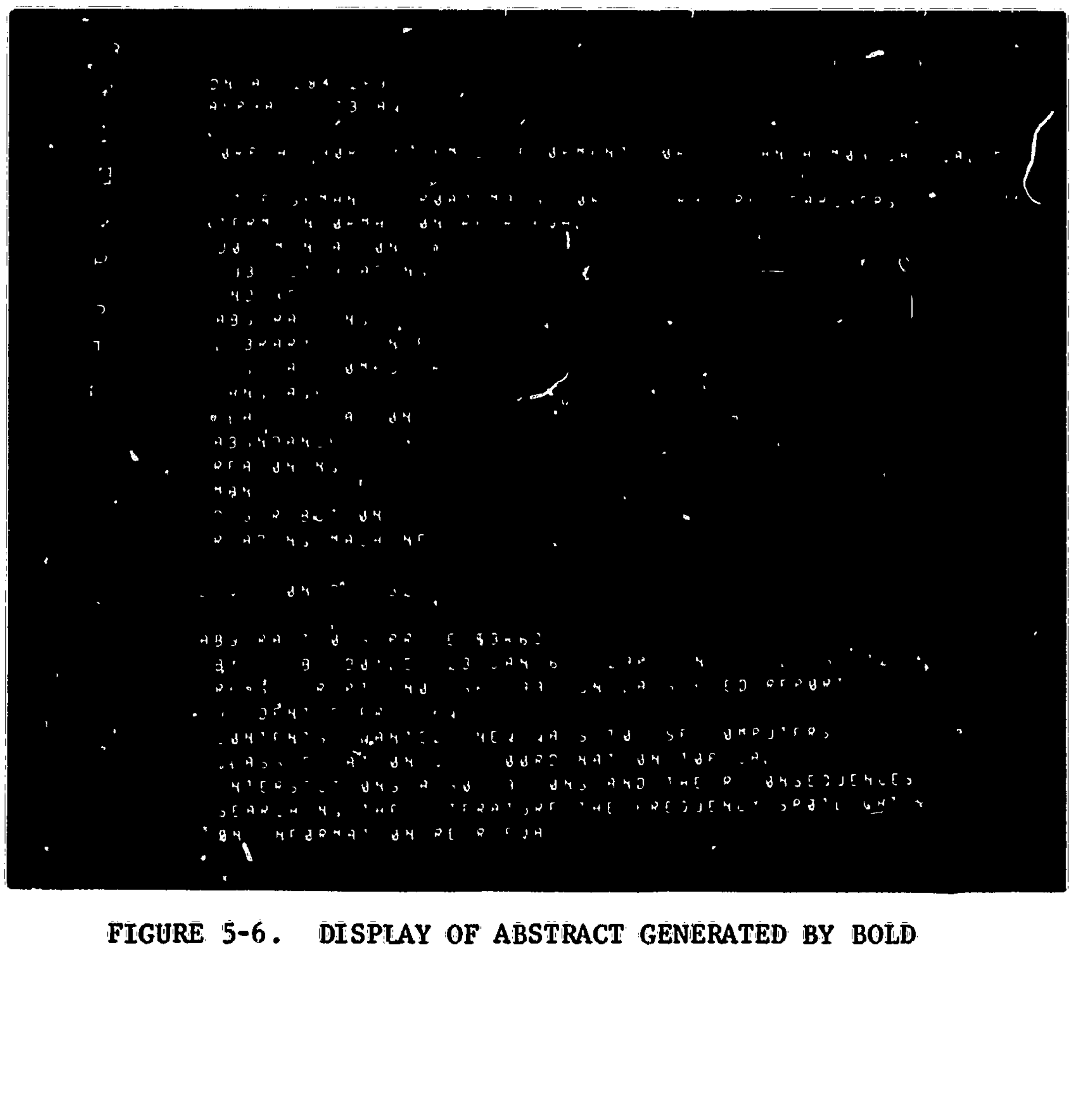


FIGURE 5-6. DISPLAY OF ABSTRACT GENERATED BY BOLD

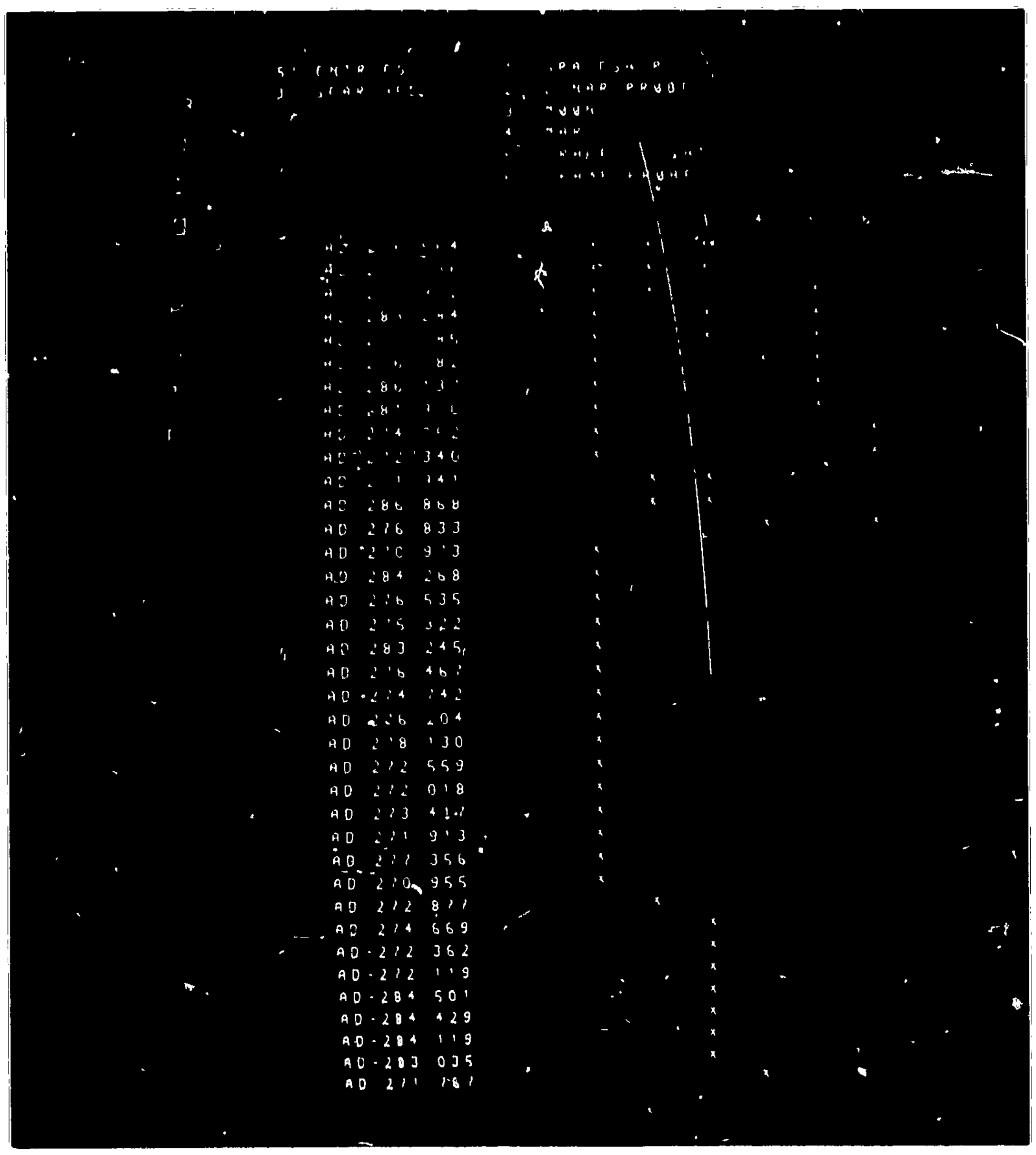


FIGURE 5-7. SEARCH MATRIX OF RETRIEVED DOCUMENTS

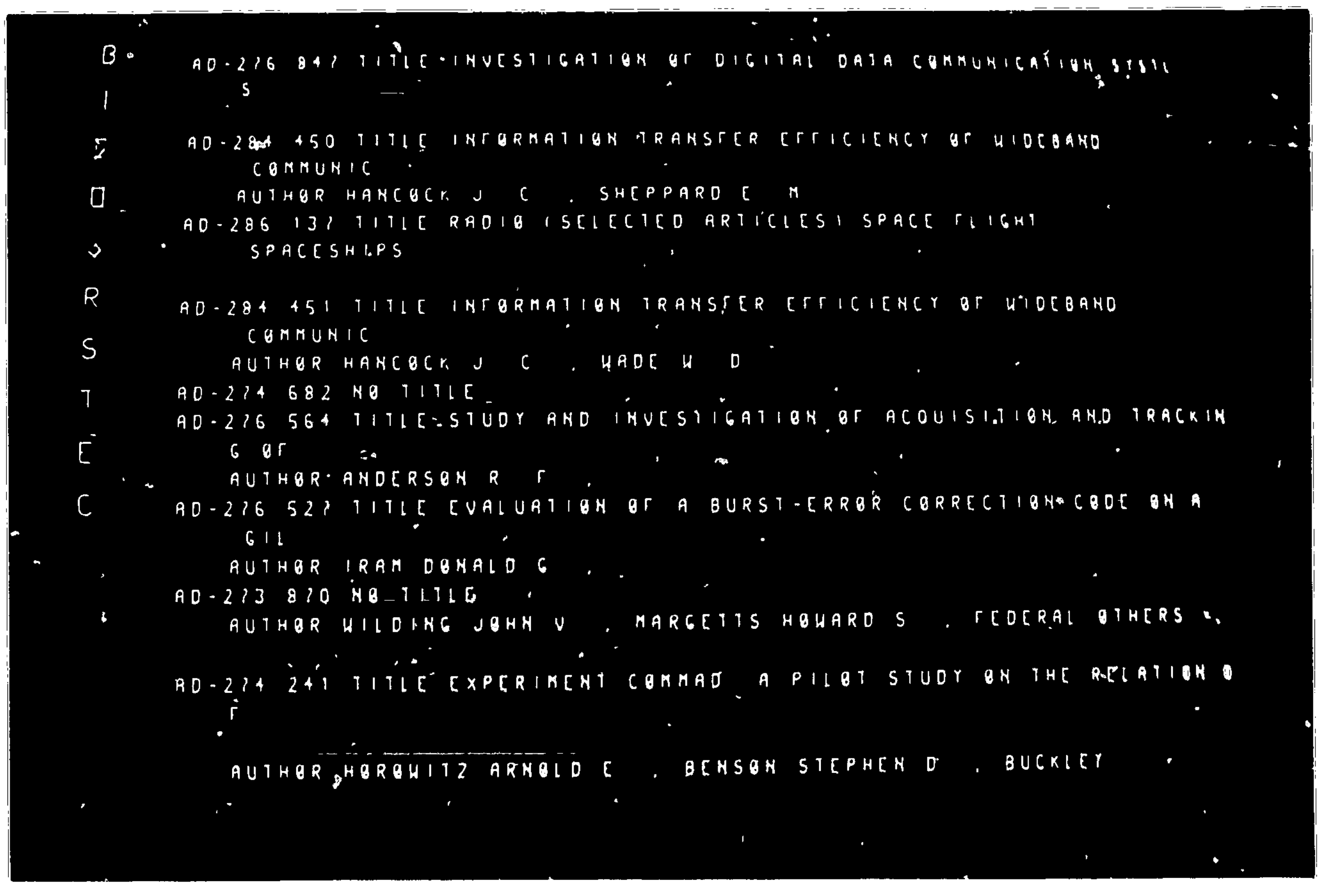


FIGURE 5-8. BIBLIOGRAPHY GENERATED BY BOLD

4. Burnaugh, H. P. The BOLD user's manual for retrieval. SDC document TM-2306/004/00. September 6, 1966. 27 pp.

5. Borko, H. Utilization of on-line interactive displays. SDC document SP-2575. August 22, 1966. 32 pp.

SURF: EDP-based Support of User Records and Files

E. M. Wallace

Description

SURF is a computer-based service that assists individuals and groups in organizing, maintaining, and finding the contents of their personal and office files. Users of the service index documents entering their files, fill out input coding sheets and submit them to the service, and regularly receive consolidated index listings. The service is implemented through SDC's MADAM language and system for the IBM 1401. The system has been designed to be adaptable to a variety of unique and differing requirements responsive to individual needs and habits of work. It is currently serving SDC personnel on a company-wide basis and is available to SDC customers.

In the course of giving service, the system builds machine-readable files reflecting individual needs, perspectives, vocabulary, indexing practices, and manner of organizing information. These files are being used to enhance mutual awareness of holdings, and to improve identification of individual information requirements for feedback to centralized information and document-handling services.

Progress

The previous year's work had established that the system was capable of responding effectively to a broad variety of very different requirements and that the service was valued by its users. During the spring and summer of 1966, the programs were documented, a user's guide was published, and a study of past user indexing practice and behavior was initiated. Alternative format and processing options were programmed, checked out, and operated to serve particular user requirements. Documentation was initiated of these latter routines and the corresponding user guide supplements. During the fall the service was offered to all SDC personnel.

One of the preliminary results of the study of user practices was that most users found that they learned to build better indexes by having to use the products of their earlier indexing decisions. They were able to diagnose poor practice and improve their own grasp of what is required in nomenclature and perspective. In addition, they testified that one of the chief virtues of the service is that it does provide a user with a product that completely reflects his outlook and manner of organizing information.

Plans

As the number and variety of SURF users grow, it is planned to further extend the study of user practice and behavior, and to develop practical means of providing regular feedback to centralized services for their modification and improvement. It is also planned to explore the feasibility of identifying people with common interests through disclosure of commonality of content in their respective files. The future capability and responsiveness of the service should be greatly enhanced through operation in a time-sharing mode.

Project Documentation

1. Wallace, E. M. SURF: Support of user records and files - description and operation. SDC document TM-2912. April 29, 1966. 33 pp.

2. Wallace, E. M. A user's guide to SURF: Support of user records and files. SDC document TM-2913. June 24, 1966. 27 pp.

3. Wallace, E. M. User requirements, personal indexes, and computer support. Proceedings, 29th annual meeting, American Documentation Institute. New York: John Wiley & Sons, 1966. (Also available as SDC document SP-2535.)

Automatic Extracting and Abstracting

H. P. Edmundson
D. L. Dwiggins

Description

Automatic extracting uses a program system to select sentences from machine-readable text and so form an extract which has proved in many cases useful as an adjunct to retrieval systems. Automatic abstracting, on the other hand, would be expected to select important topics from the text and generate sentences to form a content-representative abstract.

Progress

Automatic extracting programs were tuned to a fairly high level of effectiveness for use on the IBM 7094. Studies of methods for recognizing topics of paragraphs and text were made. The result of these studies was an indication that extensive syntactic and semantic analysis would be desirable both for recognizing topics and for generating appropriate abstract sentences. At this point the project was terminated in favor of more basic research on syntactic and semantic analysis and generation methods.

EVALUATION STUDIES

Document Representation Techniques*

R. V. Katter, Principal Investigator
E. H. Holmes
R. L. Weis

Description

This study comprises a series of experimental investigations of the sensitivity of a new method for testing the representativeness of different types of document representations (such as index terms, classifications, and abstracts). Methods for testing representativeness are important because successful document retrieval depends on the extent to which representations accurately represent documents to the searcher.

One reason it is desirable to study the process of judging representativeness itself is that the making of such judgments is a core activity upon which several important document-retrieval-system functions depend. For example, a whole series of judgments of representativeness are made whenever a system of representations is either devised or revised, or whenever a representation is assigned to a particular document, i.e., whenever a document is titled, indexed, classified, or abstracted.

Another reason for studying this judgment process is that its unreliability is a major stumbling block in comparing and evaluating various systems of representations. Many individuals' judgments of representativeness are not stable across even short periods of time, and a high level of agreement between different individuals is rarely achieved. Learning more about the sources of low reliability of representativeness judgments therefore appears to be a prime requisite for further significant progress in the field of document representation and searching.

The objective of this project is to devise and test different methods for measuring the representativeness judgment, and to use such measurements in systematic experiments to identify sources of variability entering into this judgment. Hopefully, knowledge about such determinants will lead to methods for controlling them.

In considering various methods of measuring representativeness judgments, the point of view taken in this project is that the methods should be adaptable as small-sample "test-boring" devices that can be applied to limited topical areas of documents and their representations. More specifically, to be maximally useful for research and development, the methods should have the following characteristics:

*Supported in part by the National Science Foundation.

Directness: The method should record the judgments as they are made directly from documents and from document representations, rather than having the record of such judgments mediated through a chain of system processes that introduce extraneous variables.

Response Separation: The method should allow the separation of the processes involved in judging documents and the processes involved in judging representations of those documents.

Uniform Applicability: The method should be equally applicable to judgments made on any kind of representation, rather than favoring one type.

Freedom from Content Guidelines: The method of eliciting and recording judgments should not be geared to particular kinds of content, nor inhibit the recognition of others.

Quantitativeness: The method should allow the judgments to be expressed and analyzed in quantitative terms.

Sensitivity: The method should be able to register the effects of variables known to modify complex judgment processes of the kind involved in judging document contents.

A method, known as the Pattern Congruity Method (PCM), was developed that demonstrably meets the first five of the above criteria. This method is an adaptation of the well-known psychological method of multidimensional scaling analysis of paired-comparison proximity judgments. The experiments conducted over the past year in this project were aimed at testing the PCM on the sixth criterion listed above--that of sensitivity.

To apply the PCM, a test set of perhaps a dozen documents is selected from within some delimited topical area, and a panel of appropriate judges who are familiar with and interested in the topical area are identified. The documents are processed by the two or more different representation techniques that are to be compared (e.g., index assignments are made for the documents, abstracts are made for the documents, etc.). The result is a set of representations for each type of representation being compared.

A separate sample of judges is used to judge the document test set and each of the representation test sets. An identical judgmental procedure is used for documents and for each type of representation: each judge is presented with all possible pairs of objects (documents, representations) in his set, a pair at a time. For each pair he rates the overall content similarity between the two members of the pair. The ratings for judges of each type of object (documents, representations) are averaged. For each set of objects the result is a pattern of averaged ratings of content similarity between every object and every other one.

A measure of pattern similarity or congruity can be calculated between the pattern of averages obtained from document judgments and the pattern obtained from judgments of each type of representation. The type of representation producing the pattern with the highest congruity to the pattern of document judgments is taken as the most representative. The various judgmental data can also be analyzed by multidimensional scaling programs, which produce a portrayal of the judgmental dimensions that underlie the rating data. This furnishes the ability to study the manner in which different kinds of representations portray different kinds and numbers of dimensions of document content.

Progress

Experiments were conducted to test the sensitivity of the PCM for registering the effects of four manipulated variables: degree of topical homogeneity of the document test set, level of knowledge of the judges for the topical content being judged, orientation of the judges regarding the use to which they intended to put the documents they read for the experiment, and degree of condensation of the different types of representations constructed for the documents. This series of experiments used journal articles from the field of psychology, and employed graduate students in psychology as judges.

The results showed the PCM to be appropriately sensitive to all four of the manipulated variables. Higher degrees of interjudge agreement within test sets were associated with: higher topical homogeneity; higher levels of judges' knowledge; judges' use-orientations, which stressed comparative judgments; and higher degrees of condensations of the representation. Higher degrees of congruity between document judgments and representation judgments were associated with the first three variables in the same fashion as just depicted above. However, the fourth variable (level of condensation) showed a reversed result: higher document-to-representation congruity was associated with less condensation of the representation. The conclusion was that the PCM is very promising as a method that can be developed for investigating and comparing document representation techniques.

Plans

Results of the present studies suggested some fresh hypotheses regarding the usefulness of the PCM for analyzing the conceptual patterns underlying the use of coordinate indexing terms to describe documents and information requests. These hypotheses are being investigated in the context of two operational coordinate indexing systems for technical information.

Project Documentation

1. Katter, R. V., Holmes, E. H., and Weis, R. L. Progress report: Experimental investigations of a method for analyzing document representation. SDC document TM-3090. August 9, 1966. 113 pp.
2. Katter, R. V. Validating a test of document representations. Proceedings, 29th annual meeting, American Documentation Institute. New York: John Wiley & Sons, 1966.
3. Weis, R. L. and Katter, R. V. Multidimensional scaling of documents and surrogates. SDC document SP-2713. January 23, 1967. 29 pp.

Empirical Study of Relevance Assessment*

C. A. Cuadra, Principal Investigator
E. H. Holmes
R. V. Katter
F. Neeland
E. M. Wallace
R. L. Weis

Description

This project involves a series of experimental studies aimed at discovering and analyzing the variables that determine or offset judgments of the relevance of documents to statements of information requirements. Knowledge about such variables is needed to improve methodologies for the evaluation of document storage and retrieval systems.

The proliferation of document retrieval systems and of approaches to document retrieval has brought a corresponding increase in efforts to evaluate the outputs of the various systems and techniques. Each such approach to evaluation, no matter how complicated or apparently mathematically sophisticated, has rested finally on some simple judgmental measure of the "relevance" or "pertinence" of individual documents. This judged attribute of document relevance has been conceived of in many ways, e.g., as a relation to a stated information requirement, as a relation to a hypothetical "state of need" of a document user, as a relation to a classified field of knowledge, and several others.

Investigators have differed sharply on the basic concept of relevance. Even when consensus has been reached on a definition, different individuals using the same definition have shown marked variation in rating the same items. Yet, although relevance judgments show such unreliability, there is no substitute for them in system evaluation, and it therefore becomes important to try to gain a better understanding of relevance judgments in order to improve their reliability and usefulness. The challenge is to gain this understanding quickly and efficiently.

*Supported by the National Science Foundation.

The objective of this project is to bring methods from the field of experimental psychology to bear on the problem of low reliability in relevance judgments. The mode of attack is to experimentally manipulate variables that are suspected of having a strong effect on relevance judgments and that have a wide range of different values in different individuals and across different relevance judgment situations. The hope is first to find such variables and to confirm their power experimentally, and then to learn how to control their effects in the making of relevance judgments. The specific goals of the present study are to survey the field of possible variables, to identify a set that seems especially amenable to laboratory study, and to identify experimentally some of the more powerful variables in this set.

Progress

The initial steps in the research program were to survey important literature, review and analyze the relevance problem and related methodological issues, and develop a first-round listing of possible variables for study. An initial list of more than 50 such variables was compiled, and variables from the list were selected for preliminary experimentation.

A number of separate experiments have been conducted thus far. The subjects have been (1) professionals and students in library science/information technology, and (2) undergraduate and graduate students in psychology. The experiments have used materials appropriate to each group. All of the experiments attempt to create judgment situations in which the influence of particular variables or sets of variables can be detected and measured.

A brief description of each experiment is given below, together with a summary statement of the findings.

. Implicit Use Orientation. The purpose of this study was to determine whether the particular attitude taken by the judge about the intended use of documents demonstrably affects the relevance attributed to them. The experiment explored 14 different use orientations, instilled by means of special instructions. It was found that all 14 of the special instructions produced different relevance judgments than were made under a standard "controlled" condition. It is clear that one may markedly increase or decrease relevance scores simply by means of differential instructions to the judges about intended use of the documents.

. Knowledgeability of Judges. The purpose of this experiment was to determine how much the agreement among judges on relevance ratings is affected by their level of training. Four levels of knowledgeability were defined, in terms of number of hours of academic credit and score on a specially devised test. Results show that there is not a straight-line relationship between level of knowledge and interpersonal reliability. While a high level of knowledge was related to more reliable judgments than a low level of knowledge, an intermediate level of knowledge produced even higher reliability.

. Forms of Relevance Judgments. Three forms of relevance judgments appear distinguishable: degree, probability, and pattern. The purpose of this experiment was to determine whether the form of the judgments required might have some impact on the results. The results confirmed this hypothesis and suggested that findings from evaluation studies that have used (or inadvertently permitted the use of) different judgment forms are not directly comparable.

. Restrictions on the Shape of the Distribution. The purpose of this study was to determine whether restrictions on the shape of the distribution of judgments, e.g., by requiring ranking, introduce significant distortion into relevance judgments. Distributions of unrestricted ratings often tend to pile up relevance values near the bottom or the top end of the scale, depending on the sample of documents and information requirements. Of necessity, ranking and other forms of distribution restriction distort these values into more regular shapes and introduce other anomalous effects.

. Information Specificity of Stimulus Materials. The purpose of this experiment was to determine how the specificity of the information contained in the requirement statement and the documents affects the relevance attributed to documents. Documents previously rated as being the "softer" (less information-specific) produced higher relevance rating scores, as did the less information-specific requirements statements. These results suggest that less information specificity in the judgmental materials produced lower certainty of judgments, and that the judge's strategy of judgments was to err on the side of leniency, i.e., to assign a higher relevance score.

. Sensitivity of Scale Forms to Range and Anchor Effects. Category scales, like ranking, may introduce distortions in judgments. The purpose of this experiment was to see whether a magnitude-ratio type scale might provide better sensitivity than a category scale. The results suggest that it does indeed have more sensitivity but that the resulting judgments may be less stable.

. Stylistic Valuative Structure of the Articles Judged. It is recognized that judgments made about articles may be contaminated by judge's "extraneous" attitudes and values regarding the style and manner of exposition (as separate from the content). The purpose of this experiment was to determine first whether a general valuative structure of judgments of articles could be found, and second, whether some of the dimensions of this structure can be shown to contaminate typical judgments of relevance. Factor analysis provided three readily interpretable factors, which are now being analyzed against relevance judgments made on the same documentation.

. Study of Query Techniques. Attention has been given to means for eliciting additional information about the judging process. Questionnaires, individual interviews, and "group debriefings" have been used, in addition to observations. Each of these has contributed in significant, though different, ways to the understanding of the judging behavior.

In addition to the required formal project documentation, an article has been submitted for publication. Extensive communication has also been maintained with the related relevance project at Western Reserve University and with other interested specialists.

Plans

During the early part of 1967, the project will complete several additional experiments presently in progress, examine additional variables, and prepare a comprehensive report for the project as a whole.

Additional plans are to (1) achieve a better conceptualization of document-judging behavior and (2) develop a set of guidelines for system evaluation that take appropriate account of the sources of variability discovered in the course of the project.

Project Documentation

1. Cuadra, C. A. and Katter, R. V. Semiannual progress report, empirical study of relevance assessment program. SDC document TM-2790. January 1, 1966. 7 pp.

2. Cuadra, C. A. and Katter, R. V. Experimental studies of relevance judgments: second progress report. SDC document TM-3068. July 1, 1966. 53 pp.

COMPLETED STUDIES

The following studies conducted by the Language Processing and Retrieval staff were completed prior to 1966 and are not described in this report.

Synthex

1. Klein, S. and Simmons, R. F. A computational approach to grammatical coding of English words. Journal of the ACM, 1963, 10(3), 334-347. (Also available as SDC document SP-701.)
2. Klein, S. and Simmons, R. F. Syntactic dependence and the computer generation of coherent discourse. Mechanical Translation, 1963, 7(2), 50-61. (Also available as SDC document TM-758.)
3. McConlogue, K. L. and Simmons, R. F. Analyzing English syntax with a pattern-learning parser. Communications of the ACM, 1965, 8(11), 687-698. (Also available as SDC document SP-1950.)
4. Quillian, R. Word concepts: A theory and simulation of some basic semantic capabilities. SDC document SP-2199. September 15, 1965. 39 pp.
5. Simmons, R. F. Storage and retrieval of aspects of meaning in directed graph structures. SDC document SP-1975/001/02. September 1965. 20 pp.
6. Simmons, R. F. and Londe, D. Namer: A pattern recognition system for generating sentences about relations between line drawings. SDC document TM-1798. June 5, 1964. 41 pp.

Language Communication Reliability

1. Katter, R. V. Language structure and interpersonal commonality. Journal of Verbal Learning and Verbal Behavior, 1964, 3, 132-138.
2. Katter, R. V. Research bases of language data processing system design. SDC document TM-1199. June 17, 1963. 52 pp.

Automated Content Analysis

1. Ford, J. D., Jr. Computer analysis of text for the measurement of social perception during human relations training. SDC document SP-1373/001/00. January 9, 1964. 11 pp.
2. Ford, J. D., Jr. Automated content analysis. SDC document TM-904. February 18, 1963. 12 pp.

Automatic Abstracting

1. Borko, H. and Chatman, S. Criteria for acceptable abstracts: A survey of abstracting instructions. American Documentation, 1963, 14(2), 149-160.

Text Analysis

1. Doyle, L. B. The microstatistics of text. Information Storage and Retrieval, 1963, 1(4), 189-214. (Also available as SDC document SP-1083.)

Term Diagram Studies

1. Ford, J. D., Jr. and Holmes, E. H. Evaluation of report summaries: Quantitative results and discussion. SDC document TM-662/001/00, May 23, 1962. 15 pp.
2. Ford, J. D., Jr. and Holmes, E. H. The effects of grammatical structure upon verbal encoding habits. SDC document SP-967/000/01. June 21, 1963. 14 pp.

Automatic Generation of Association Maps

1. Doyle, L. B., Conley, R. E., Meyer, W. E. and Karas, D. E. Programmed generation of association maps from collections of keyword lists: Final report. SDC document TM-2083. September 17, 1964. 50 pp.
2. Doyle, L. B. Indexing and abstracting by association. American Documentation, 1962, 13(4), 378-390. (Also available as SDC document SP-718/001/00.)

Semantic Analysis

1. Klein, S. Dynamic simulation of historical change in language using Monte Carlo techniques. SDC document SP-1908. December 30, 1964. 19 pp.
2. Klein, S. Automatic paraphrasing in essay format. Mechanical Translation, 8, Issues 3&4 combined, August-December, 1965. (Also available as SDC document SP-1602/001/00.)
3. Olney, J. C. Some patterns observed in the contextual specialization of word senses. Information Storage and Retrieval, 1964, 2, 79-101. (Also available as SDC document TM-1393.)
4. White, J. H. The methodology of sememic analysis with special application to the English preposition. Mechanical Translation, 1964, 8(1), 15-31. (Also available as SDC document SP-1339.)

Automated Document Classification

1. Borko, H. The construction of an empirically based mathematically derived classification system. AFIPS Proceedings of the 1962 Spring Joint Computer Conference. (Also available as SDC document SP-585.)
2. Borko, H. and Bernick, M. D. Automatic document classification. Journal of the ACM, 1963, 10(2), 151-162. (Also available as SDC document TM-771.)
3. Borko, H. and Bernick, M. D. Automatic document classification, Part II, Additional experiments. Journal of the ACM, 1964, 11(2), 138-151. (Also available as SDC document TM-771/001/00.)

4. Borko, H. Measuring the reliability of subject classification by men and machines. American Documentation, 1964, 15(4), 268-273.

5. Doyle, L. B. Re-expression in standardized code to improve the automatic classifiability of text items. SDC document TM-2213. February 25, 1965. 32 pp.

Stylistic Analysis

1. Sedelow, S. Y. Stylistic analysis: Report on the first year of research. SDC document TM-1908/100/00. March 1, 1965. 105 pp.

2. Sedelow, S. Y. Form recognition in literature. IFIP Proceedings, 2, 1965.

BEHAVIORAL GAMING AND SIMULATION RESEARCH*

G. H. Shure, Head

Laboratory gaming and simulation continue to be used as research and training tools of great value in many fields of social and organizational behavior. Applications range over such diverse areas as economics, education, business management, city planning, international relations, and strategic studies of political-military crises.

Discriminating researchers are aware, however, of the many fundamental problems that limit the valid applications of simulation studies. Laboratory simulation inevitably creates a problem of interpretation, i.e., of inferring correspondence between the simulation model studied in the laboratory and the "real" world. This is particularly true where the world created (simulated) in the laboratory attempts to simulate complex aspects of social reality. For instance, if political-military gaming is ever to be of more than heuristic value, so as to afford a basis for decision making, the methodology of simulation must itself be subject to direct scrutiny.

The problem expresses itself in such basic questions as these: (1) Which parts of a given model should be deterministic, probabilistic, or man-decision operated? (2) How can more elementary models be expanded to incorporate more subjects, strategies, and properties and yet yield to observation, control, and analysis? (3) How can the experimenter determine to what extent human decisions made in the environment of the model are like those made in the real world? (4) What are the effects of participants' motivations, value systems, personalities, and perceptions in a continually changing laboratory situation? Traditionally, gaming for research and training has ignored many of these questions and dealt with others only superficially. One major reason has been the problem of data handling: first, the difficulty of recording data in the precise detail and large volume required; and second, the problem of usefully analyzing such an enormous and complex body of data.

While these basic questions are not easily answered, the use of a large-scale computer-based laboratory, such as that developed in the Behavioral Gaming and Simulation program, permits significant inroads to be made into the problem of validity, which remains relatively intractable under attacks of a smaller scale.

For the simpler experiments the multistation computer-based laboratory (see also p. 9-7) permits a large number of replications, and experimental variations are easily introduced by simple program changes. Perhaps more important, the laboratory enables the construction of more detailed and complex simulation models, which, when run, can be recorded and analyzed more directly at the micro as well as macro dimensions of behavior. Use of such detailed behavior results in a better reconstruction of the perceptions, attitudes, responses, and interactions of the participants.

*Formerly called Decision Processes Research, the name of the program was changed to reflect more accurately the character of the developing and future research program.

Where replication is not a feasible alternative to establish the reliability and validity of results, this approach of embedding the experiment in a larger matrix of observations affords another way of bridging the simulation/real-world correspondence problems. To record all these phenomena, and to afford experimenter control over them, requires a capability for on-line recording and evaluation that only a computer provides. Furthermore, the data recorded under these conditions, if they are to be optimally examined and analyzed, require not only computer processing but also newly developed capabilities designed to detect order and pattern in such complex events.

The current studies of the group address these questions at various related levels, ranging from models of individual learning in simple environments, through behavior and perceptions of subjects in mixed-motive two-person games, to behavior of 20 or more individuals interacting within a complicated social organization.

The project on Human Data Processing has continued its study of how an individual's choice is affected by different kinds of environments and reinforcement arrangements (i.e., different ways of informing him about the accuracy of his predictions). Experiments requiring a subject to predict which of two events would occur next, showed that seemingly trivial differences in the way the subject was informed about the correctness of his choice significantly affects his performance. The study also showed that a subject was more responsive to feedback based on prior success than on prior failure, e.g., in making his choice, he was more likely to "stay with a winner" than to "shift from a loser," even when the more adaptive strategy was to shift.

The project on Decision Making and Leader Selection, concerned with the relationship between an individual's success in making correct decisions and the likelihood of his being elected leader of a small group, has been completed, and a report has been drafted. The experiment showed that the mathematical model's predictions of the voting and leader selection fitted the data well after voting behavior stabilized. Certain revisions for the model were indicated. These have been made, and the model has been extended to cover four- and five-man groups.

In the continuing research on Bargaining and Negotiation Behavior, an unusually heavy schedule of laboratory runs was completed before the removal of the Philco-based laboratory, and much analysis remains for the months ahead. Additional experimental variations of the Territories Game and the Pacifist Game have been conducted to explore--more fully--bargaining strategies, the use of threat, and the effects of bargainers' personality characteristics. Among findings to date are that high situational conflict leads to less achievement of what is jointly possible than low conflict situations. Results also suggest that subjects obtain useful ideas by being asked via probes about what they are doing. However, at high levels of conflict, subjects are unable or unwilling to use this information. Thus, the probe, developed as a methodological technique, emerges as a potentially significant tool in the bargaining process itself. As in earlier studies it was found that adversaries used threat little when it was available to both, suggesting self-imposed restraint produced by the prospect of retaliation. This restraint is markedly reduced when the capability is unilateral. Three new paths of experimental research were started in the past year. First, some 350 subjects were run through the Prisoner's Dilemma Game, in experiments to explore a number of detailed questions about negotiation behavior in a simple context. Second, a new Bartering Game was designed and run with approximately 100 subjects to study bargaining trust under different risk conditions. Third, SDC participated in a coordinated program with nine other social psychology research centers located in various parts of the world. Each used an SDC/UCLA-designed game in conducting its own series of experiments. (This is believed to be the first time that such a program of simultaneous, directly comparable, cross-cultural experiments has been conducted on such a scale.) Researchers met in Santa Monica on November 17 to compare results and to plan for the continuing year.

Work has also continued on Leviathan, a computer-based model of large social organizations, in which some 24 human decision makers of various levels interact with each other and with machine-simulated robots. Analysis has continued on data gathered in earlier experiments. The model is being modified to increase its flexibility and data gathering efficiency, in preparation for its adaptation to the SDC S/360 Time-Sharing System. During the summer of 1966 Leviathan was used experimentally as a training tool in a management workshop conducted at USC for Air Force officers (graduate students of the AF Institute of Technology).

Research has continued toward increasing the researcher's ability to deal with the large volume of detailed data generated by the computer-based behavioral experiments. The power and versatility of the TRACE program for the on-line manipulation and analysis of data was further developed (TRACE II), and a user's handbook was prepared. A related project, IDEA, is attempting to develop an on-line program that joins the powers of human and computer induction, in the analysis of socio-psychological data, through the use of heuristic decision trees, which summarize the structure of the multivariate data base. In another project, four experiments were conducted to test the effectiveness of computer aids to human problem solving.

Two small programs were developed to aid in data analysis and interpretation: VARIANC, foı computing theory-predicted statistics for behavior in extended series of binary events (e.g., stimuli, responses); and BINAL, for analyzing events in such series in relation to one another.

HUMAN DATA PROCESSING BEHAVIOR

O. H. Heuckeroth,* Principal Investigator
R. Weichel
M. L. Mochson

Description

A major goal of the psychologist is understanding the relationship between the characteristics of the environment and the way a person responds to these.

Three important types of environments have been identified. (1) In a noncontingent environment, one event does not depend upon any other that has already occurred; e.g., the likelihood of a coin coming up heads on the second throw is independent of the outcome of a previous toss. (2) In an event-contingent environment, one event is contingent upon a previous event; e.g., a landslide is more likely to follow rain. (3) In a response-contingent environment, an event depends upon the individual's preceding response; e.g., injury often follows fighting.

Our work has been primarily concerned with two problems: First, how is behavior affected by the way in which people are informed about the appropriateness of their responses; i.e., how does performance differ with one method of providing feedback as opposed to another method? Second, how do people perform against different kinds of environments; e.g., those in which event and/or response contingencies do or do not exist?

For research on both these questions, subjects were required to predict on each of a large number of trials which of two events, A or B, would occur. One reason for using this experimental situation was that the dependency of one event on prior events (and responses) could be kept under strict experimental control, and prediction behavior as a function of each type of environment could be reliably assessed. Perhaps just as important is the fact that this paradigm lends itself to mathematical modeling.

*Replaced B. R. Wolin on March 31, 1966.

Progress

Contingent and Noncontingent Feedback Arrangements. When a person is to perform some task, it is desirable to know how to maximize his rate and final level of performance. One factor that has received little consideration is the seemingly trivial (but theoretically distinct) differences in the way that people are informed about the correctness of their behavior. For example, after a person has completed some task we could tell him whether he was correct or incorrect by making an evaluative response of "RIGHT" or "WRONG," we could present him with the correct answer and allow him to infer whether he was right or wrong, or we could provide him with no feedback at all. These three modes of feedback can also be identical for both correct and incorrect responses (noncontingent), or the mode of providing information could be made contingent upon the correctness of the response. As an example of the latter, we could tell a person "RIGHT" when he is right and give him the correct answer when wrong; or we could say nothing when he is right and give him the correct answer when he is wrong. Thus, if we consider providing evaluative feedback by saying "RIGHT" or "WRONG" as one mode of providing information, presenting the event (correct answer) as another, and providing no feedback as a third, then there are six contingent and two (nontrivial) noncontingent feedback arrangements that can be considered (see Figure 6-1 below). No feedback

		Feedback when the response is correct		
		EVALUATIVE "RIGHT"	CORRECT EVENT	NONE
Feedback when the response is incorrect	EVALUATIVE "WRONG"	a		
	CORRECT EVENT		b	c
	NONE	d	e	

FIGURE 6-1. THREE POSSIBLE MODES OF INFORMING PEOPLE ABOUT THE CORRECTNESS OF THEIR RESPONSES

for both correct and incorrect responses (shaded cell in Figure 6-1) would convey no information and therefore no predictable change in performance.

Cells on the main diagonal (a, b, and shaded cells) represent noncontingent feedback arrangements. The remaining cells are contingent.

Using the binary choice situation, subjects were run in the five lettered feedback arrangements of Figure 6-1. The findings indicate that when feedback is provided for only the correct or the incorrect response, learning is more effective when that feedback is provided for the correct response (d and e) than for the incorrect response (c). Where noncontingent feedback is provided for both correct and incorrect responses, the evaluative mode is more efficient than is event feedback (a vs b). No feedback for the incorrect response (d and e) tends to yield performance comparable to that in the optimal evaluative feedback condition (a). The differentiation of feedback mode for correct and incorrect responses appears to offer a promising new distinction for shaping the learning process. Detailed analysis of these results is now being completed.

Complex Contingent Reinforcement. It is generally assumed that people will repeat a response if it is correct and change it if it is wrong. However, this strategy ("win-stay--lose-shift") will be sound only if the environment is structured so that repeating previously correct responses tends to be correct again, and shifting from incorrect responses tends to result in correct responses. But it is possible to conceive of an environment where such a strategy would not be adaptive, and where the adaptive strategy would be to shift when correct and stay when incorrect. To investigate people's ability to adopt predictive strategies that are consistent with the reinforcing properties of the environment, subjects were run in a binary prediction experiment under six experimental conditions. The results indicate that where the "win-stay--lose-shift" strategy was most adaptive, people were more likely to "stay with a winner" than to "shift away from a loser," even when the most adaptive performance would be to shift away. Where the reverse, "win-shift--lose-stay," strategy was most adaptive, people were more likely to shift when correct than to "stick with a loser," even when performance would be more adaptive if they did stick. It appears, therefore, that subjects are more responsive to contingencies based upon prior success than on prior failure, even when the failure is objectively more important. One would expect that if these results were due to subjects not fully understanding the meaning of the contingencies, adding a history of previous events and/or predictions would make the contingencies clearly evident. The effect of providing a history, however, did not materially change the results. These findings imply that if outcomes are to serve as instructions (in the form of contingent relationships) for future behavior, performance will probably be better if those outcomes are successes.

Plans

Work in the immediate future will involve a shift in emphasis from examining decision making behavior as a function of the structure of the environment to making a detailed study of the effects of motivational variables. Our initial work will involve examining the effects of monetary payoff on several aspects of decision making behavior, e.g., the time required for each decision and the confidence with which that decision is expressed. The data will also be used to develop a mathematical model that will describe each of these aspects of decision making behavior.

Project Documentation

1. Wolin, B. R. and Weichel, R. Binary prediction behavior when the generating source resembles a language. SDC document SP-1953. March 31, 1966. 21 pp.

DECISION MAKING AND LEADER SELECTION*

O. Heuckeroth** } Principal
A. Binder, Consultant at SDC } Investigators
(University of California, Irvine)
J. K. Oslund
S. J. Terebinski

Student Associate: R. L. Degerman, Johns Hopkins University

Description

This research investigates the relationship between success in decision making and the likelihood of being voted leader (group decision maker) of a small group. The study analyzes small-group behavior in terms of concepts and procedures derived from statistical learning theory. The goals are to develop a model that will predict various aspects of group performance, such as leadership selection, voting behavior, and learning trends, and ultimately to develop and apply such a model to larger groups and more complex situations.

Experiments were conducted in the SDC Research & Technology Laboratory under control of the Philco 2000 computer. The experiments involved a team of three individuals playing a game of alternate leadership selection and decision making. Each trial began with a majority selection of a leader by the three members (self-voting was not permitted). The leader then made a decision for the group. On the next trial, members again voted for a leader after being informed of the correctness of the leader's decision on the previous trial. Subjects were indirectly motivated to vote for the member who seemed the most successful decision maker as indicated by updated team scores of correct decisions that were presented to the members.

The particular decision made by the leader on each trial involved predicting which of two events would occur next in an event sequence like that used in probability learning; actually, the correctness of a leader's decision was a function of a reinforcement schedule under control of the experimenter. By varying these schedules, differences in voting behavior could be studied. The initial set of experiments was run with 600 trials--later sets were reduced to 480, as it was found that the last 120 did not contribute additional knowledge about the course of learning.

Progress

In general the model predictions of voting and leader selection fit the data for asymptotic trials quite well, except for a condition in which monetary incentive was provided. In all cases, however, learning during the initial trials, i.e., before voting behavior had stabilized, was not well represented by the model. Also, the member who was reinforced most often when he was leader voted more often for the next-most reinforced member than was predicted by the model. For example, in the condition 975 (where the numbers mean that one member, when leader, was reinforced with a probability of .9 for his decisions, another with p = .7, and the third with p = .5), the .9 member voted for the .7 member on 78 percent of the asymptotic trials, whereas the predicted value was 56 percent. It seemed clear that a motivational question was involved here, for when a 931 condition was rerun using a monetary incentive, the .9 member was selected as leader almost 100 percent of the time.

Data analysis clearly indicated the need for a revised model and showed the kind of revision required. The main alteration involved the axiom structure of the model--which made assumptions about the probability of vote shift as a function of who was leader and his correctness on the previous trial. This axiom structure was revised to take into account the direction of voting shifts, i.e., whether a member shifted his vote for leader from the high member (compared to himself) or to the low member; and, further, a learning process was introduced into the parameters

*Supported in part by the National Science Foundation.

**Replaced B. R. Wolin on March 31, 1966.

to describe the probabilities of vote shifting. Specifically, the probabilities were assumed to change throughout the course of learning, as each member acquired more information about the decision making ability of other members. Such a revised model has been elaborated, but only partly tested. The model has also been extended to cover four- and five-man groups.

A final report has been written presenting experimental findings and other work accomplished.

Plans

A final theoretical paper is in preparation, which will discuss the mathematical approach used, the modifications made, and the relation of the findings to statistical decision theory.

The next step would be to test the revised model more fully on empirical data, and then attempt to extend it to more complex situations. A study plan has been drafted for applying the model to an actual situation in which a group is faced with evaluating various sources (radar stations) which are transmitting the same information but with different degrees of reliability.

Project Documentation

1. Binder, A. and Terebinski, S. J. Decision making and leader selection--final report. SDC document TM-3030. June 24, 1966. 24 pp.
2. Terebinski, S. J. The effects of financial reward upon voting preferences in small groups. SDC document SP-2503. June 7, 1966. 13 pp.

BARGAINING AND NEGOTIATION BEHAVIOR*

G. H. Shure, Principal Investigator
R. J. Meeker
W. H. Moore, Jr.
F. V. Daniels (NIH Postdoctoral Fellow)

Consultants: H. H. Kelley, M. Seeman, University of California, Los Angeles

Description

This research program is designed to lead to a set of empirically derived propositions on bargaining and conflict resolution behavior and processes that occur in situations where the parties involved have both common and conflicting interests, that is, where they are mutually dependent in the pursuit of otherwise antagonistic goals. Of particular concern are those issues and strategies in bargaining that appear to be relevant, actually or potentially, in political-military confrontations, particularly limited conflicts.

Within this framework, this project continues the investigation of the effects of three factors on bargaining processes and outcomes: (1) variations in threat availability and conditions affecting threat employment, (2) variations in cooperative or conciliatory moves and orientations, and (3) variations in bargainers' personality characteristics and attitudes.

The research employs two-person, nonzero-sum bargaining games in which the computer is used as an experimental tool for on-line analysis, umpiring, control, and recording of subject behavior, particularly the dynamic interaction process that takes place between the bargainers. A computer program pairs one player against another or against a simulated player. Each subject sends messages (moves, bids, threats, offers) to his paired opponent via button insertions, which are displayed on the receiver's display scope.

*Supported in part by the Advanced Research Projects Agency.

Computer programs present game situations to the subjects, assist in umpiring legal moves, provide displays of game-relevant information, and record all moves, messages, and times. Between game messages and moves, or between trials, the computer, through displayed questions, probes the subject for the bases of his actions, his current intentions and expectations, and his perception of his opponent, or through subtle or disguised items indirectly elicits the subject's attitudes about the bargaining situation.

Progress

Work undertaken previously was continued and extended during 1966. Because of the anticipated time gap between the removal of the Philco-based laboratory and the availability of the SDC IBM 360-based laboratory, an unusually heavy schedule of laboratory runs was completed. Two new games (Prisoner's Dilemma and Bartering Game) were programmed and run; additional experimental runs with three previously described games (Territories Game, Pacifist Game, and Communications Game) and one series of runs with a manual game (Transnational Game) were also completed.

Territories Game [3]. This game, adapted from Shelling's Map Game for on-line computer use, affords an analogue to territorial disputes and permits tacit signaling of intentions and information through a rich vocabulary of game actions. The game has been designed to avoid research difficulties associated with other similarly complex games. It embodies continuity of bargaining history, permits meaningful degrees of threat, and provides no obvious basis for a "fair-share" resolution--the bargaining values are not given, but must be articulated in the course of bargaining moves. Experiments have employed two variations: degree of territorial conflict between players (high, medium, or low interference on outcomes), and variations in potency of threat and reprisal moves (nonavailable, unilaterally available, or bilaterally available).

Experimental variations also either omitted or included computer probes and questions between moves. The former conditions permit an assessment of the effects of the probes procedure itself; the latter permit a more refined analysis of the effects of threat availability.

Comparisons of outcomes for the various combinations of experimental conditions yield the following results: (1) In general, high conflict leads to less realization of maximum available joint outcomes than low conflict. (2) These differences in outcome are not as large where threat is not available, or in bilateral threat conditions where probes are used. (3) For experimental conditions without probes, the availability of bilateral threat actions leads to poorer joint outcomes than those from comparable conditions without available threat capabilities (57 percent versus 80 percent of the maximum possible joint outcome). However, where probes are used in the bilateral threat conditions, low- and medium-conflict pairs do as well as those with no threat available. This finding contrasts sharply with that obtained for high-conflict pairs. Here, if the probe has any effect, it decreases the joint outcome obtained (45 percent versus 33 percent of the maximum possible joint outcome).

These results suggest that subjects gain useful ideas by being asked about what they are doing, but that the extent to which they are able to use this information is related to the degree of objective conflict they confront. If conflict is high, subjects appear unable or unwilling to use the information; if it is lower, they appear better able to use it. For instance, the presence of probes is associated with an almost complete elimination of threat employment in the two lower-conflict conditions. Thus the probe condition, initially introduced as a methodological refinement, has emerged as a potentially significant aspect of the bargaining process with substantive implications for negotiation under threat conditions.

An analysis of game moves indicates that in conditions with bilateral availability of threat, relatively little threat was used--only 1/8 of the bargainers overtly threatened their adversaries. An obvious interpretation of these results is that relatively little use of threat, under these conditions, reflects a self-imposed restraint based on the prospect of retaliation. This interpretation was tested--and confirmed--when the capability for retaliation was removed by the unilateral threat variation under all three levels of conflict: Seventy percent of the bargainers threatened adversaries who were known to be incapable of retaliation (as compared with about 12½ percent otherwise). Another result from the unilateral threat condition was that the bargainer possessing the threat capability dominated the outcomes.

Another focus of our data analysis concerned the factors of information exchange--identifying methods used for signaling between parties, assessing relative correctness of resulting perceptions, and measuring the relationship of successful information exchange to the bargaining outcome. Attempts by subjects to signal the importance and degree of resolve through initial territorial claims led to more favorable outcomes.

A final report on this series of studies is in preparation.

Prisoner's Dilemma Game. A number of aspects of cooperative or competitive behavior can be studied more precisely in simpler games. One particular variant of a nonzero-sum game, the Prisoner's Dilemma, has been the focus of a rapidly growing literature (Gallo and McClintock, Rapoport and Orwant). The general form of this game is represented by the following payoff matrix:

		Choices for Player II	
		B_1	B_2
Choices for Player I	A_1	X_1, X_1	X_2, X_3
	A_2	X_3, X_2	X_4, X_4

and is described by the following set of rules:

(1) $2X_1 > X_2 + X_3 > 2X_4$

(2) $X_3 > X_1$

(3) $X_3 > X_2$

(4) $X_4 > X_2$

If each subject selects the strategy best for himself (A_2 and B_2), each realizes the outcomes in cell A_2B_2. This outcome is less, however, than either would realize if the players were able to tacitly agree to select the A_1 and B_1 strategies. Thus each player has compelling reasons to compete and at the same time has a genuine stake in cooperating with the other. The ability of subjects to reach the tacit agreement has been studied under variations in payoff matrix, communication opportunities, incentive, personality of players, length of play, etc.

One fairly consistent pattern of results reported by investigators is that the cooperative responses run well under 50 percent and decrease over a series of trials. Furthermore, the subject's rate of cooperative responses appears uninfluenced by the opposing simulated strategy, whether cooperative or competitive. Many subjects, who initially respond cooperatively to the simulated cooperative strategy later switch to competitive ones. The subjective perceptions of subjects in these situations have been left to speculation. For instance, is the switch to competitiveness based on an attempt to exploit? Or is there some question as to how a real subject perceives a player who is, say, "83 percent" cooperative? How are the random defections of the simulated cooperator perceived? These and other puzzling results can be clarified by obtaining the subject's perceptions and expectations of the other player in the context of play. More important, the subjective significance of behavior elicited in a vast array of mathematically generated, simple game situations permits us to compare mathematical formulations and empirical patterns with the data obtained by the probe technique in these situations.

The flexibility of the computer-administered game also permits some direct mathematical tests of assumptions found in some models. For instance, the behavior of a player may be predicted on the frequency with which he has encountered the A or B choices in a series of preceding trials. The "intentions" of the other players are not explicitly considered as factors influencing a player's behavior. If this assumption is warranted, players should respond identically to the same pattern of game actions whether these are emitted by a single opponent or by different ones. The latter condition, which is almost impossible to run meaningfully outside a computer-based experimental setting, can be convincingly and easily realized in our laboratory (see condition 3 below).

The Prisoner's Dilemma Game has been installed in the laboratory and approximately 350 subjects have been run under eight sets of conditions: (1) Live subject vs live subject. (2) Live subject vs simulated subject. Here the simulated player is either predominantly cooperative or competitive, in a ratio of 80/20 or 20/80 respectively. (3) Round robin among live players. In this, each player switches to a new, unspecified adversary on each successive trial. (4) Round robin with simulated players. This is the same as 2, except that the subjects are given the orientation of 3--i.e., they are told they will be playing against other subjects in the round-robin mode, but they are in fact playing against the predominantly cooperative or competitive ratio of choices. (5) Live players vs cooperative "shaper." Here one member of each pair (unknown to the other) is given the task of getting the other to cooperate by whatever tactics he chooses. (6) Live player vs cooperative shaper with payoff advantage. In this, the payoff values are manipulated so that the shaper is in the advantageous position. This condition will be compared with 5 above to assess the importance of a power advantage in eliciting cooperation. (7) Live player vs simulated tit-for-tat shaper. Here, the simulated strategy matches the previous response of the subject, as a "firm" or "conditional" cooperator--firm in that defection from cooperation will not go unpunished, conditional in that if the live subject is cooperative, the simulated subject will cooperate. (8) Live player vs tit-for-tat shaper with payoff advantage. This condition combines the strategies of 6 and 7 above.

Conditions 1 and 2 are yielding results comparable to those of similar experiments conducted elsewhere; the relevance of interactive history and data from other conditions are leading to clarification of some of the factors that inhibit successful joint outcomes.

A series of papers is being prepared.

Bartering Game. This game [2] was designed to study the development of bargaining trust under different conditions of situationally defined risk and jeopardy. The study explores the importance of procedural factors in bargaining situations that guard against, or fail to guard against, the danger of unexpected loss for either or both parties.

Two experimental variations are of primary interest:

a. Mode of contract determination: In a jointly determined contract condition, both participants must agree on a proposal before a trade agreement is reached; in the unilaterally determined contract condition, every offer is open to exploitation by the other party.

b. Level of proposal cost: High and low costs create conditions of greater or lesser risk associated with bargaining actions.

These two manipulations were expected to interact, serving to create variations in perceived jeopardy. Thus, in the unilaterally determined contract condition, since proposals are not conditionally protected by contract agreement procedures, self-defeating patterns of bargaining exchanges were anticipated where proposal costs are high.

The four combinations of the two experimental variations were run with 12 pairs of subjects in each condition. In accord with predictions, where subjects were required jointly to reach a binding agreement before either could act in a way affecting the other, increased incentives and associated increased proposal costs led to the highest outcomes. Where subjects could act independently, without requiring agreement from the other bargainer to a proposed action, increased incentive and proposal costs led to poorest outcomes. Thus, increased incentive for successful negotiations may make it more rather than less difficult to initiate successful offers and agreements if the party initiating the proposal must place himself in a state of even temporary vulnerability to exploitation (as in a decision to disarm unilaterally).

The results from these studies should help to clarify the form of influence of a pervasive, structural feature of many bargaining situations, and to clarify some inconsistent effects of incentive variations reported in experimental studies. Subjective factors and perceptions and features of the interaction process that mitigate or extenuate the overall results are being analyzed.

Transnational Game. As a result of participation in a 12-man international conference, Dr. Shure, and Dr. Harold H. Kelley of UCLA, undertook the major responsibility for developing a manually administered bargaining game [4] that was used in nine centers in Europe and America. The task was designed to evaluate factors influencing the decision to enter into a bargaining contract or to "go it alone." The effects of relative power of the two bargainers and incentive variations were evaluated in a 2x2 experimental design. Eighty subjects were run in each completed design.

The game affords opportunities both for the collection of hard data on bargaining and for the observation and recording of the free verbal interaction within or between the bargaining parties. The extensive replication of the experiment, both nationally and abroad, makes this social psychological study a unique effort in interlaboratory collaboration, and particularly for the evaluation of cross-national comparisons.

The data collected from the various laboratories were collated and analyzed at SDC using TRACE (see p. 6-15) and other computer programs in preparation for an international conference held in Santa Monica during November 1966 (see also p. A-4).

Pacifist Game. In an earlier series of experiments a game was designed to clarify relationships between nonviolent strategies and their effects on bargaining. In these runs, the computer was programmed to simulate the role of a cooperative (pacifist) bargainer. A description of the game, problem areas explored, and a sample of the findings obtained were reported [5].

In this experimental series, the unaided moral appeal was not generally found to be an effective bargaining tactic. Although communication with the adversary aids the ethically motivated bargainer, his appeal has limited effectiveness, and clarification of his motives and interactions is as likely to lead to further exploitation as to the improvement of his outcomes.

A replication of the earlier experimental design with this game was made to evaluate the effect of the presence of an "observer" on the responsiveness of subjects to a simulated pacifist appeal. Forty-eight subjects were studied under a combination of experimental conditions. Preliminary analysis indicates that the presence of the observer did not materially improve the effectiveness of the moral appeal made by the pacifist. Further analyses of these findings should clarify why exploiting subjects are uninfluenced by such social exposure.

Communications Game. Hypotheses concerned with threat and psychological factors in conflict escalation were tested in three earlier series of experiments using the Communications Game. A review of related research, methodological issues, and detailed findings has now been summarized in a monograph-length presentation [6] which has been submitted for publication as a book.

Plans

We plan to continue our program of research in the new 360-based laboratory along the lines of our earlier studies. Technical improvements in game procedures are being designed to take advantage of improved console and computer capabilities. The detailed formulation of the conflict situations that will be represented in the new games will be specified in the process of analyzing and developing theoretical models for the extensive series of completed studies. In addition, plans are being developed to extend the techniques of the simpler games to a relatively complex political-economic game.

Project Documentation

1. Daniels, F. V. Communication, incentive, and interaction process in interpersonal exchange and negotiation. SDC document SP-2523. July 7, 1966. 40 pp. (To appear in Journal of Experimental Social Psychology.)
2. Daniels, F. V., Meeker, R. J. and Shure, G. H. The bartering game. SDC document TM-3082. August 2, 1966. 38 pp.
3. Meeker, R. J. and Shure, G. H. Self-administered instructions for the territories game. SDC document TM-2442. June 1, 1965. 70 pp.
4. Shure, G. H. and Kelley, H. H. Transnational bargaining game. SDC document SP-2554.
5. Shure, G. H., Meeker, R. J. and Hansford, E. A. The effectiveness of pacifist strategies in bargaining games. Journal of Conflict Resolution, 1965, 9, 106-117.
6. Shure, G. H., Meeker, R. J., Moore, W. H., Jr., and Kelley, H. H. Computer studies of bargaining behavior: The role of threat in bargaining. SDC document SP-2196. February 9, 1966. 172 pp.

LEVIATHAN*

B. K. Rome, S. C. Rome } Principal Investigators
M. L. Almquist
R. E. Krouss
S. J. Terebinski

Description

Leviathan research investigates the structure and government of large social organizations by means of computer-based laboratory simulations. The computer represents essential features of bureaucratic organizations and their working forces, and live subjects enact the roles of executives in a pyramid of command. Twenty to 30 subjects meet for 8 to 12 hours each week over three-month periods; they provide hundreds of computer-simulated robots with decision rules. The robots in the computer carry out the policies of the executives, and the computer furnishes each live decision maker with knowledge of technological accomplishments according to his level of command, functional specialization, and territorial authority.

Large organizations are viewed as hierarchically structured: They consist of multilayered, interinvolved, dynamically developing complexes of social transactions. Through successions of these transactions, an organization grows, develops, and continually restructures itself. Our theoretical interest is to identify and to trace stages of growth in such transactional behaviors,

*Supported in part by the Air Force Office of Scientific Research, Office of Aerospace Research.

particularly those associated with high organizational performance. Experimental objectives are to determine interrelationships between (1) executive policy formation, decision making, and control, and (2) system performance of the simulated organizations, within (3) experimentally controlled organizational environments. As an experiment progresses, the experimenter monitor and assess the subjects' executive transactions, track their performance accomplishments, and responsively modify the subjects' organizational environment.

Progress

What began a year and a half ago as a solution to a data processing problem has led to a consequence of basic theoretical significance: the formulation of legislative processes of large organizations in abstract, general terms.

The initial data processing problem was one of logistics. Because of the limits of the computer employed until now, massive quantities of data were recorded in traditional ways--by sound-tape recordings, direct visual observation through one-way glass, and handwritten messages. Analysis of these data showed that the subjects' more important transactions were taking place outside the computer, in their extraformal agreements and disagreements. With the impending availability of the extensive S/360-65 computing facility for on-line, real-time research, it became possible to bring a large part of these extraformal transactions into the computer.

Meanwhile, further analysis of the experimental data revealed that, by means of their extraformal agreements, the subjects were accommodating themselves to the structures of the formal organization imposed in the experiments. Actually, they were deliberately transforming precise, unambiguous formal structures, existing in tables in the computer, into ambiguous, multisignificant expressions. This was being done in three distinct areas: authority, responsibility, and the power to effect changes in their operations. These transformations, furthermore, proved to be a large part of the self-organizing process by which the subjects attempted to accomplish the common tasks set them.

It became clear that we should no longer attempt to impose organization on the subjects in the areas in which they had seized control, but rather should let the subjects themselves structure their organization.

The question arose: How should the subjects declare their self-organization to the computer? If they were simply to place a mass of freely structured declarations into the computer, the logistic problem would not really be solved. No adequate programs exist for analyzing freely structured social response, although many are being attempted.

In previous research, we had developed trees of formal organization in which subsystems are distinguished by the functional specialization of the live decision makers within the pyramid of command. We realized that these trees could be so elaborated that separate structures of authority, of responsibility, and of social power could be designed to exist in the computer. These structures could then be altered piecemeal by suitable interventions taken over the computer by the subjects. Each intervention by a subject would produce an orderly organizational consequence in the computer. Transgressions of authority, failures of responsibility, and usurpation of power could also be monitored by the computer.

By means of new programs to be made operational in 1967, the subjects will be able literally to build and to rebuild their bureaucratic organization in response to experimentally controlled exigencies. Response spaces, constrained as desired by the experimenters, will exist in the computer. Within these spaces the subjects will have complete freedom to select their interventions.

FIGURE 6-2. SUBJECTS BEING ORIENTED DURING LEVIATHAN EXPERIMENTS

Each intervention will declare an intent to the computer, and will elicit an overt response from it. Each intervention that produces an overt change in the evolving organization in the computer will thereafter affect the evolving social world in the laboratory.

Computerizing social transactions provides the basis for their conceptual coding. Because the actions take place over a digital computer, they are classifiable, countable, and transformable by rule. Further significance of this formalization is that the operations that can be realized in the laboratory can be postulated of natural bureaucracies. Natural organizations can also be viewed as progressions of transactions like those incorporated in our computer programs. Hence, hierarchical processes that develop and are observed in idealized form in the laboratory environment can serve as prototypes of actual processes in natural large organizations.

A comprehensive book-length report on the Leviathan methodology is about half finished. It will include: (1) a full presentation of our conceptual approach, (2) the translation of the approach into an abstract, general simulation vehicle, (3) a detailed description of the simulation vehicle, together with a systematic unfolding of design principles, (4) a complete account and rationale of the specifications that convert the abstract general vehicle into an operating system, an instrument for experimental control, (5) a full account of the dialogic technique for conducting large-group live/robot experiments, (6) an analysis in depth of the 1963 experiments, (7) a statement of the considerations that went into the redesign of the simulation vehicle in 1964, (8) an analysis of the accomplishments of the 1964 subjects, (9) and finally a description of the developments that led to the design of the next-generation simulation vehicle and the 1967 experiments.

During the summer of 1966, the Leviathan staff provided support for a management training workshop conducted by USC with approximately 30 Air Force Institute of Technology graduate-level officer trainees under the direction of M. G. Holmen and J. V. Zuckerman. This was a pilot study for the adaptation and development of the Leviathan methodology to executive training.

This was the first time that the Leviathan simulation of an organization was manned by other than graduate college students. The consensus of the observers and Air Force participants was that Leviathan was a promising management training vehicle, but that its potentiality for such use required modification and developmental work. A forthcoming document by Holmen and Zuckerman will describe the exploratory study in some detail.

Plans

Experiments will be conducted in 1967 with new programs and the new computer system. In these experiments, administrative authority, responsibility, and power will initially be exercised at the higher levels of command in a simulated organization, and operational responsibilities will be allocated to the lowest level of live command. The object of study will be the stages of development in this stratification of governing activities. These experiments will form part of the continuing Leviathan program: to discriminate patterns of effective social growth in large organizations that successfully cope with uncertain futures in crisis environments.

Project Documentation

1. Rome, B. K. and Rome, S. C. Information processing potentials in large-scale operations. SDC document TM-1128/100/00. July 18, 1966. 40 pp.

2. Rome, B. K. and Rome, S. C. Humanistic research in large social organizations. In J. F. T. Bugental (Ed.) The challenge of humanistic pyschology today. New York: McGraw-Hill, in press.

3. Rome, B. K. and Rome, S. C. Leviathan: An experimental study of large organizations with the aid of computers. In R. V. Bowers (Ed.) Studies on behavior in organizations. Athens, Georgia: University of Georgia Press, 1966, Chap. 11.

4. Rome, B. K. and Rome, S. C. Automated learning process (ALP). Ibid., Chap. 12.

5. Herold, V. P. Bibliography of project Leviathan documents. SDC document TM-983/000/02. March 31, 1966. 9 pp.

COMPUTER METHODOLOGY FOR DATA ANALYSIS

TRACE: Time-Shared Routines for Analysis, Classification, and Evaluation*

R. J. Meeker
W. H. Moore
G. H. Shure

Description

In an inductively oriented experiment, where the goals of hypothesis generation may be of greater importance than those of hypothesis testing, the experimenter will be inclined to collect any data that hold promise of amplifying or clarifying his understanding. As a result, the computer is sometimes both a curse and a blessing to a researcher, for the increased amount of information that can be generated in a computerized study, particularly an on-line experiment, is often outweighed by the burden of time and effort required to organize, collate, and assess it. The data from these experiments will frequently be highly redundant; and in addition, much of the data may be irrelevant. If these data are also hierarchical, sequence-ordered, and of variable length (frequently characteristic of experimental data), the problems of data management may become overwhelming.

The TRACE program was developed as an outgrowth of these data analysis difficulties as encountered in the context of studies in bargaining and negotiation. TRACE is intended to assist in classifying, grouping, and summarizing experimental data so as to identify summary and configurational indices that satisfy the criteria of reliability, specificity, validity, and relevance. TRACE is designed to assist the investigator in exploring relationships that may obtain among complex sets of data from a number of different and newly suggested points of view until he is satisfied that he has derived optimally defined operational indices; it also permits rapid checking of hypotheses about patterns and relationships for particular subsets of the data.

The desired data manipulations--recombining, regrouping and recalculating--are conducted in a time-sharing mode of computer operation and make use of computer-connected teletypes and CRT displays. TRACE is written in JTS for the AN/FSQ-32 computer; the on-line capability of the program permits immediate feedback to the user about the relative utility of derived indices and permits adoption or modification of these for further analyses. TRACE permits an effective interplay between the investigator's conjectural and judgmental skills and the computer's capacity for rapid and accurate data processing.

Progress

Model II of TRACE is now operational and is composed of seven major subprograms, which permit the user to perform the following operations:

- Input raw data from cards or tape. Dictionary description of data permits retention of sequential and nested hierarchical order in data tables. The program performs internal checks on completeness of data input. The storage and processing of logic permits the user to add subsequently to his original data base.
- Derive new measures based on raw data or previously derived measures. Derived variables

*Supported in part by the Advanced Research Projects Agency.

can be based on simple linear partition, on a variety of combinations of logical relations, or on a variety of mathematical operations or transformations. Sequentially ordered data can be dynamically indexed (i.e., end points of the search can be conditionally specified). The search can be restricted to selected subsets of the data based on specified variable values. Highly similar specifications can be parameterized to reduce redundancies of user interaction. Processing logic places no logical limitation on size of data base or derived variables.

. Construct a tabulation or cross-tabulation of any derived measures. The number and width (or included values for nominal variables) of distribution intervals can be specified. Interval limits based on standard specification (equal width, equal numbers, all distinct values) are automatically determined by the program; otherwise, interval limits are supplied by user input.

. Display on teletype and/or CRT display any previously constructed tabulation. Displayed tabulations include marginal distributions, simple statistical indices of distributional properties, and preliminary tests of the relationship between cross-tabulated measures. Particular subsets of data can be identified and/or tagged by user reference to their position in the univariate or bivariate tabular displays (this permits the user to test hypotheses about particular cases that fail to fit the general pattern of results).

. Edit the data base. The user can list values under any variable, change any value in the data base (to correct input error, to adjust coding discrepancies, etc.), rename any variable or table, delete variables, tables, or data sets.

. Output on tape or cards any variable in the data base. This permits the user to use the derived measures as input to other programs, especially batch processing statistics.

. Use statistical analysis. This offers the user a set of two sample tests, a data base generator for input to IDEA (see below), or the option of operating IDEA as a TRACE subprogram.

Plans

Design objectives and principles for TRACE III for the IBM 360 computer are formulated. They include the following major innovations: (1) a multidimensional derivation philosophy, (2) a self-instructional capability, and (3) a tutorial capability. A translation capability from natural language to TRACE III formal language is also being evaluated. Coding of these capabilities will be initiated shortly.

Project Documentation

1. Meeker, R. J. User interaction with the TRACE system. SDC document TM-2621/002/00. October 19, 1966. 18 pp.
2. Shure, G. H. TRACE II--Timeshared routines for analysis, classification and evaluation. SDC document TM-2621/001/00. October 12, 1966. 8 pp.

IDEA: Inductive Data Exploration and Analysis

M. S. Rogers, Principal Investigator
G. H. Shure

Student Associate: L. I. Press, University of California, Los Angeles

Description

This project uses a novel technique for attempting to discover the structure of a multivariate data base, and to summarize it in a decision tree format.

Classical induction algorithms have been successfully employed to reveal structure. When the structure becomes complex, man's experience, knowledge of the field from which the data come, intuition, and pattern recognition skills may enable him to succeed where mechanical techniques fail. To overcome the limitations and restrictions of pure machine induction, IDEA allows the investigator to collaborate with a heuristic library in the "growing" of decision trees.

The data base consists of the values of m variables which are observed (measured) for a set of n entities (e.g., individuals, interviewees, experimental subjects, etc.). For a given analysis, one of the variables is considered dependent. The group of m observations associated with a particular entity defines a point in a space of m dimensions. A decision tree partitions that space into exhaustive, mutually exclusive regions. This effect is similar to that of many pattern recognition systems,[1] cluster-seeking systems,[2] concept-learning programs,[3] and discriminant analysis.

Figures 6-3 and 6-4 illustrate decision trees for an abstract data set and for a data set of hypothetical questionnaire responses.

This type of decision tree is a generalization of those used by Hunt _et al_., and by Morgan and Sonquist.[4] It is a useful form in which to summarize data since it can represent complex relationships between variables that are measured on any scale (nominal, ordinal, interval). In addition, interactions between variables or involving different relationships within one variable are easily seen; and, since it focuses on one variable at a time, a decision tree is easily understood and interpreted. The tree may be used to estimate the value of the dependent variable for unknown entities. It may also be used to guide further research to test the hypothesis that it reliably portrays the structure of the population from which the entities in the data base were drawn.

The processes for deriving decision trees by means of IDEA have three distinguishing features:

1. There is extensive provision for interaction between the investigator and the analysis programs.
2. Heuristic techniques make such processes computationally feasible.
3. IDEA can be applied to data of interval, ordinal, and nominal scales via the ability to select appropriate heuristics from a modular library.

Since the number of decision trees that could result from a practical data base is large or infinite, the routines that search for structure are of necessity heuristic. They seek out a subset of the feasible partitions at a particular node. These "promising" partitions are then evaluated using heuristically chosen criteria for assessing the ability of a partition to explain the variation in the dependent variable. The heuristics must decide which independent variable to use at a node and how to discriminate on that variable or they may decide that the path should be terminated. The process used at the first step in the analysis, when the entire data base is considered, is the same as that for subsets at lower points in the tree.

Progress

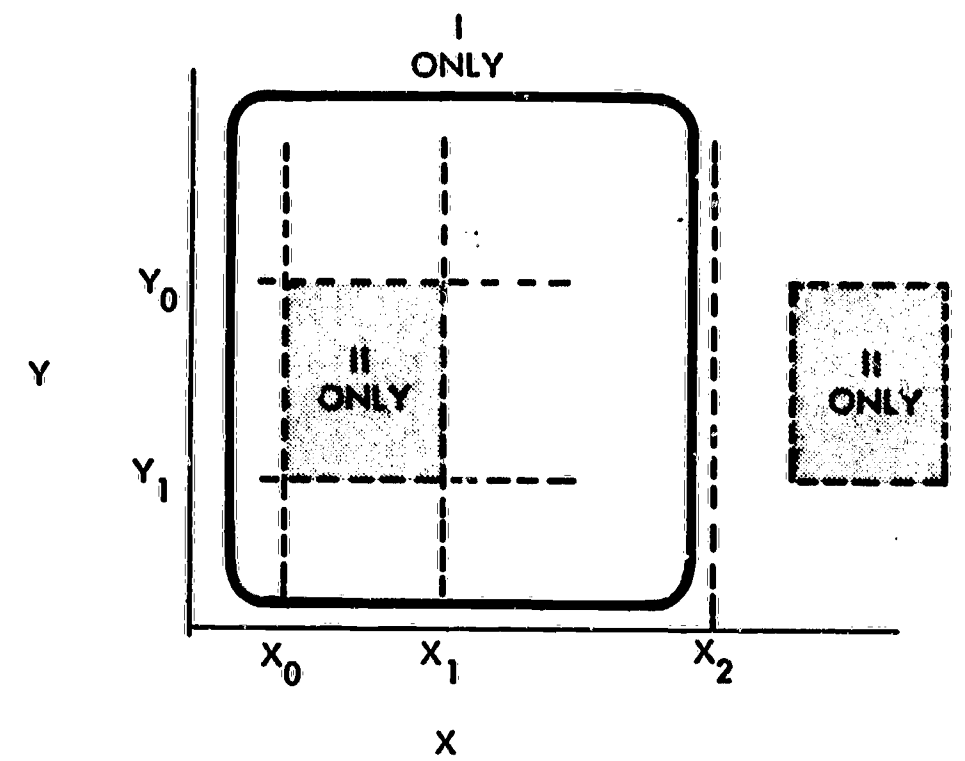
The type of display produced by IDEA on a CRT console is shown in Figure 6-5. IDEA accepts data from the TRACE system (see above) and can be accessed by a TRACE user. Figure 6-6 summarizes the measurement scales for which heuristics have been programmed. These and the executive routine [1] are operational in SDC's Time-Sharing System.

[1] Sebestyen, G. Decision making processes in pattern recognition. New York: Macmillan Co., 1962.

[2] Ball, G. Data analysis in the social sciences. AFIPS Conference Proceedings, 1965, 27, Part 1, 533-560.

[3] Hunt, E., Marin, J. and Stone, P. Experiments in induction. New York: Academic Press, 1966.

[4] Morgan, J. and Sonquist, J. Problems in the analysis of survey data--and a proposal. Journal of the American Statistical Association, 1963, 58, 415-534.



The independent variables are X and Y, which are measured on at least an ordinal scale. The dependent variable is nominal and takes on the values I or II in the indicated regions.

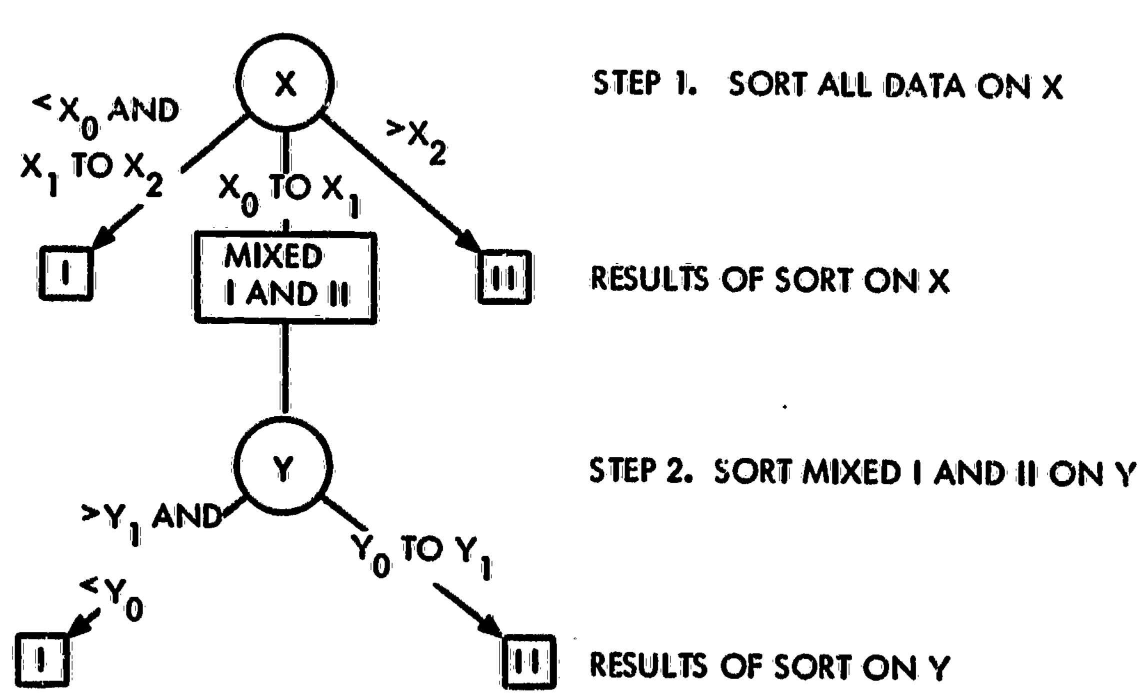


FIGURE 6-3. HYPOTHETICAL DATA BASE AND DECISION TREE

1. WHAT IS YOUR AGE? (AGE)
2. WHAT IS YOUR SEX? (SEX)
3. WHAT IS YOUR EDUCATION LEVEL? (EDUC)
4. HOW OFTEN DO YOU ATTEND CHURCH? (RELIG)
5. DID YOU PARTICIPATE IN THE RIOTS OR NOT? (RIOT)

Questions (variables) of hypothetical questionnaire for Watts residents

	AGE	SEX	EDUC	RELIG	RIOT
QUESTIONNAIRE $_1$	22	F	12	NEVER	NO
QUESTIONNAIRE $_2$	18	M	10	WEEKLY	YES
⋮	⋮	⋮	⋮	⋮	⋮
QUESTIONNAIRE $_N$	45	M	6	TWICE A YEAR	NO

(Sample Data Base)

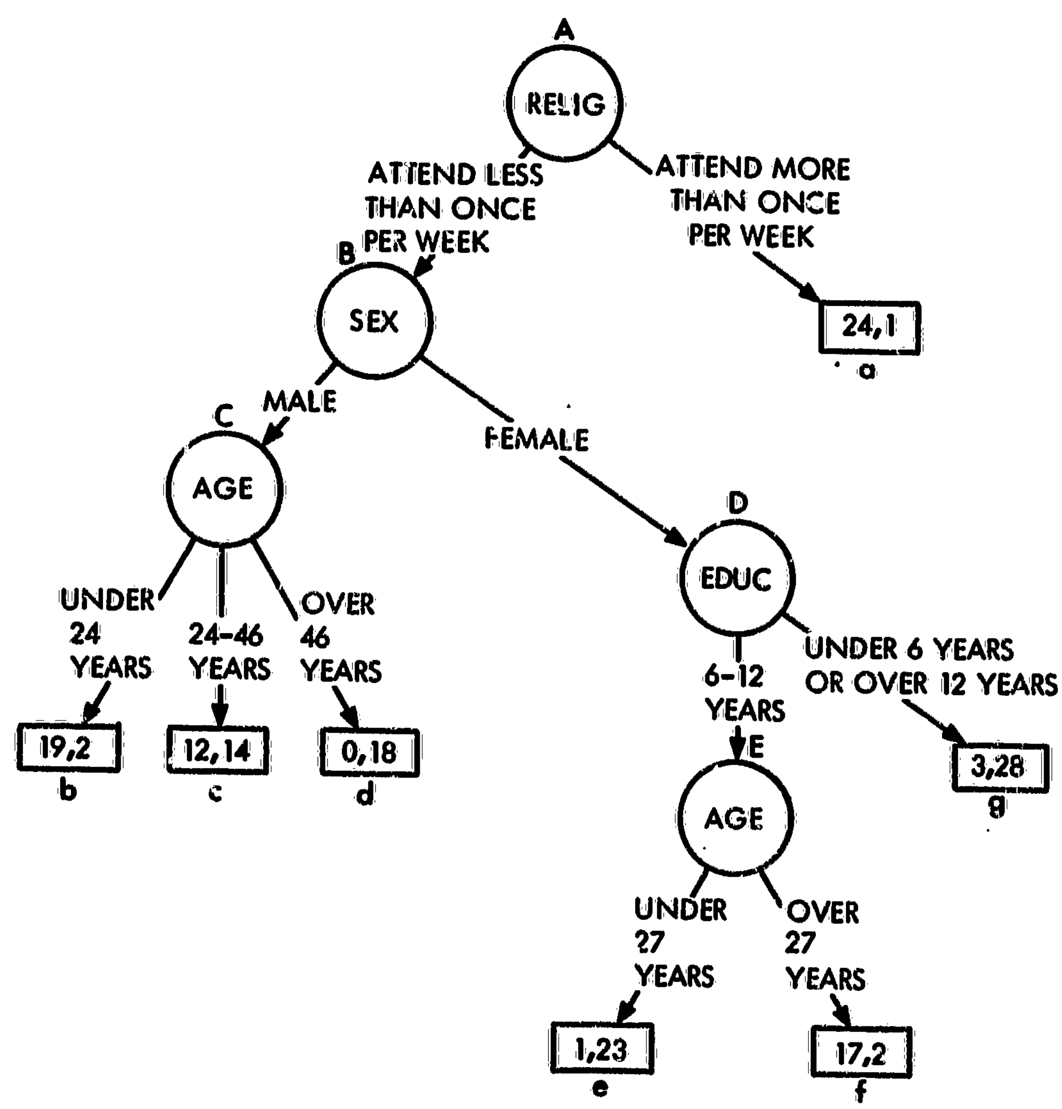


FIGURE 6-4. SAMPLE DECISION TREE FROM WATTS RIOT DATA

In this display, RIOT is used as the dependent variable. A node (circle) specifies how the chosen variable is used to partition the entities which sort to that node. A leaf (box) contains a function (in this case the frequency distribution for the dependent variable) of the subset of the entities in the region represented by the leaf.

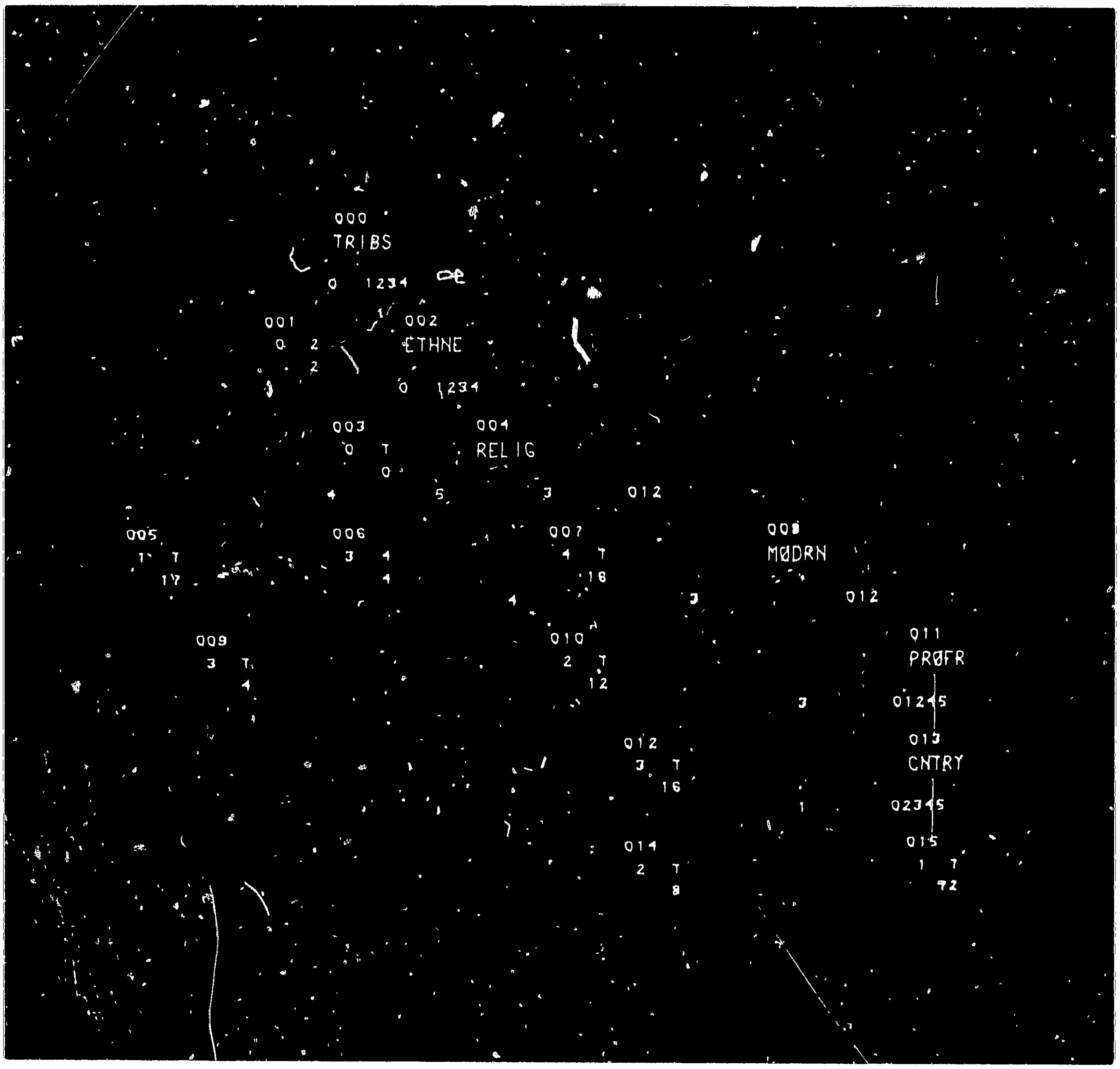


FIGURE 6-5. SAMPLE CRT DISPLAY FOR IDEA

This decision tree displays the predicted and actual responses of African nationals, based on 1023 replies to 61 socio-political questions. Each labeled node (such as ETHNE) represents questions selected by the program which should allow the prediction of the value of some dependent variable (in this case, the attitude of Africans toward voting). Each branch of the tree represents some mix of responses. For example, the first left branch in the tree (under TRIBS) indicates responses labeled "0" to some question; the first right branch indicates responses labeled "1, 2, 3, and 4." The numbers at the bottom of the branches indicate (from the bottom up) the number of respondents who "sorted" to that point on the tree, the number of those who made the predicted response (T indicates less than 75 percent), and the response predicted for the dependent variable. (The 3-digit numbers are merely node numbers, e.g., 000--at the very top--indicates the 0 node.)

Independent Variable	Dependent Variable: Nominal	Ordinal	Interval
Nominal	3	1	1
Ordinal	1	1	1
Interval	1	1	1

Currently operating / Planned for next version (shaded)

FIGURE 6-6. THE NUMBER OF HEURISTICS IN LIBRARY FOR VARIOUS COMBINATIONS OF LEVELS OF MEASUREMENT

Plans

Beyond the immediate goal of increasing and improving the library of heuristics, we also plan to perform two empirical studies. The goal of one study is to assess the validity and effectiveness of IDEA's heuristics in order to guide our ongoing effort to improve the library. In this study we will employ IDEA (largely in the automatic mode) in the analysis of data having (1) built-in, or at least well-known, structure, (2) random structure, and/or (3) structure about which related machine-induction programs disagree.

The goal of the second study is to determine how best to exploit IDEA's interactive capability. We have identified four or five different data exploration and analysis activities in which the interactive mode should be important. Empirical evidence for IDEA's utility will be obtained by having researchers use IDEA on their own data and then report their experiences with it as compared with other analyses made using the same data.

Project Documentation

1. Rogers, M. S. and Press, L. I. Inductive data exploration and analysis (IDEA)--progress report. SDC document TM-3205. October 12, 1966. 16 pp.

Display and Computer Aids to Human Problem Solving

J. R. Newman
M. S. Rogers

Description

Previous studies have shown that humans carry out a variety of systematic or rational operations when they are solving difficult problems. For the most part, these operations have been ignored in the experimental work on aiding problem solving. One purpose of this project was to carry out an experimental analysis of these operations by making computer and display aids available to the problem solver, thus enabling him to manipulate the basic elements of the problem he is trying to solve. A basic assumption underlying this work is that a combination of man and machine elements will perform certain problem-solving tasks better than either one alone.

Our experimental approach allowed subjects to manipulate symbols and kept a systematic record of every operation. The research vehicle used was an inductive reasoning task adapted from a card game. The subject was presented with a sequence of playing card symbols generated by the computer and displayed on a cathode ray tube. He was required to discover the underlying rule governing the sequence. The symbols were defined according to the familiar attributes: red-black, odd-even, high-low, major-minor. The rules governing the sequence were defined according to these attributes. For example, one rule might have been that all odd symbols are followed by black symbols.

This task was chosen because it involved inductive reasoning (i.e., hypothesis generation and verification), because it could be made as difficult as the experimenter desired, and because it represented an intellectual challenge for most subjects.

Progress

Modifications to the original set of computer and display aids for human problem solving were programmed and made operational on the Philco 2000. The new program retained in modified form all the original functions and added a new one called the Select and Count function which enabled the subjects to do a simple statistical analysis on the symbols representing the problems. The total set of display and computer aids, called symbol manipulation functions, enabled the subject to perform a series of transformations on the raw information defining the problem. These transformation functions included selection, deletion, counting, adding symbols, and recoding symbols into simpler form. The functions could be used singly or in combination.

A series of four experiments was conducted, and the results were analyzed. These experiments explored some of the factors involved in human problem solving for two major types of concept problems: classificational and relational. Both types of problems required subjects to discover a rule underlying a set of symbols. In the classification problems the rule assigned all valid symbols to a definite class defined by certain attribute values, such as that all valid symbols are either ODD or BLACK. In relational problems the rule required that the symbol attribute values bear a definite relation to one another, such as that all RED symbols are followed by ODD ones. Several levels of problem difficulty were used for each major type of problem.

The results showed that subjects provided with the display and computer aids performed better than subjects who did not have such aids. The usefulness of the symbol manipulation functions increased with problem difficulty within both major types of problems. These aids were used more extensively and had their greatest utility for the relational concept problems, which proved considerably more difficult than did classification concept problems.

There was considerable transfer of problem solving performance across problems, but this was more apparent for aided subjects than for nonaided subjects. This was especially true for the relational concept problems. The nonaided subjects in these experiments showed some improvement in relational problem solving ability across sessions; however, they consistently made more errors, took much longer, and solved a lower percentage of problems than did the subjects in the aided groups. As a result there was a greater difference in performance between the groups at the end of the sessions than at the beginning.

The results of these experiments support the hypothesis that subjects engage in a two-stage process in solving concept problems. In the first stage they recode the stimulus sample (symbol set) and in the second stage they search for the relevant attributes of this recoded symbol set to try to discover the underlying rule. The symbol manipulation aids can be used to facilitate both parts of this process. For relational problems the search phase is dominant; it is extremely difficult for most people unless they are given aids of some kind.

Plans

The experimental phase of this project was completed in August 1966 and the project will be terminated with the completion of the written reports.

Project Documentation

1. Newman, J. R. and Rogers, M. S. Experiments in computer-aided inductive reasoning. SDC document TM-3227. December 6, 1966. 78 pp.

VARIANC: Algorithm to Compute Predicted Preasymptotic Variance of Response Proportions

O. H. Heckeroth

Description

In recent years much effort has gone into an attempt to develop and evaluate several mathematical models to describe decision making behavior in the binary choice situation. The evaluation of these models has centered almost exclusively on their ability to describe behavior after it has stabilized, i.e., has reached an asymptotic value. The Weak-Strong conditioning mode, an N-element stimulus sampling model, developed by Myers and Atkinson, has been particularly successful in this respect. A reasonable next step was to evaluate the ability of this model to describe behavior before it has stabilized, i.e., preasymptotically.

Progress

Using the technique for deriving preasymptotic statistics described in a forthcoming paper by Heuckeroth and Myers,* an algorithm was written in FORTRAN II to obtain the predicted changes in the variance of unconditional response probability throughout the entire course of learning over blocks of trials of arbitrary size.

The program operates through two successive sets of matrix multiplications where the input to the vector of a second matrix multiplication is obtained as output from a first matrix multiplication. After the second matrix multiplication has been carried out K-n times, a new set of entries into the second vector is obtained through a first matrix multiplication. The variable K is the block size over which an estimate of variance is desired and n is an index whose value is determined by the number of times the first matrix multiplication has been carried out. This process is repeated until K-n=1.

All that is required as input for this program is the total number of trials over which predictions are desired, the number of trials in each block, and an assumed set of initial values to begin the first matrix multiplication. There is no practical limit to the total number of trials or the number that may be used in each block.

Plans

The project was completed and documentation is being prepared.

BINAL: A Program to Analyze Binary Sequences

M. L. Mochson
R. J. Meeker
G. H. Shure

Description

Many experiments yield data in the form of binary values in two parallel series. These series could represent trial-associated pairs of stimulus and response events, as in binary choice experiments, or trial-associated responses obtained from two-person, Prisoner's Dilemma experiments.

Progress

BINAL is a program written in FORTRAN IV to exhibit the sequential occurrence of simple events or conditionally defined patterns of events in binary-valued data formatted in two parallel series of up to 1000 events. The BINAL program displays the trial-by-trial pattern of a number of possible relationships within and between the elements in the paired series. It calculates frequencies and probabilities of the occurrences of specified events and relationships (digrams and trigrams composed of terms from one or both series) for the entire length of the series and for up to four block sizes selected as units of analysis. Group summary counts are also produced for all specified block sizes for all analyses.

Plans

The project was completed and documentation is being prepared.

*Heuckeroth, O. H. and Myers, J. L. Choice behavior and reward structure: Preasymptotic statistics. Journal of Mathematical Psychology, in press.

COMPLETED STUDIES

The following studies conducted by the Decision Processes Research staff were completed prior to 1966 and are not described in this report.

Communication Networks and Group Tasks

1. Shure, G. H., Rogers, M. S., Larsen, I. M. and Tassone, J. A. Group planning and task effectiveness. Sociometry, 1962, 25, 263-282. (Also available as SDC document SP-346/001/00.)

Decision Making in Structured Groups

1. Meeker, R. J., Shure, G. H. and Rogers, M. S. A review of the group decision-making research projects. SDC document TM-731. June 25, 1962. 47 pp.

Terminal Air Traffic Control Research

1. Alexander, L. T. and Cooperband, A. S. Schematic simulation: A technique for the design and development of a complex system. Human Factors, 1964, 6, 87-92.
2. Alexander, L. T. and Cooperband, A. S. The effect of rule flexibility on system adaptation. Human Factors, 1964, 6, 209-220.

Detection of Compound Motion

1. Alexander, L. T. and Cooperband, A. S. Visual detection of compound motion. Journal of Experimental Psychology, 1966, 71(6), 816-821. (Also available as SDC document SP-1946.)
2. Alexander, L. T. and Cooperband, A. S. A method for applying statistical decision theory to system task analysis. SDC document SP-2168/001/00. 14 pp.
3. Cooperband, A. S. The use of a computer in conducting psychological experiments. Behavioral Science, 1966, 11(4), 307-311. (Also available as SDC document SP-1516.)

Human Visual Capabilities in Reconnaissance

1. Simon, C. W. Rapid acquisition of radar targets from moving and static displays. Human Factors, 1965, 7(3), 1-12.
2. Simon, C. W. Effects of magnification and observation time on target identification in simulated orbital reconnaissance. Human Factors Journal, 1965, 7(6), 569-583.

Leviathan

1. Almquist, M. L. GOCI: A general-purpose vehicle to assist system communication. SDC document TM-805/001/00. December 5, 1963. 9 pp.
2. Rome, B. K. and Rome, S. C. Leviathan, and information handling in large organizations. In A. Kent and O. E. Taulbee (Eds.), Electronic Information Handling. Washington and London: Spartan Books and Macmillan and Co., 1965, Chap. 16.
3. Rome, B. K. and Rome, S. C. Programming the bureaucratic computer, IEEE Spectrum, 1964, 1(12), 72-93. (Expanded version also available as SDC document SP-1690.)
4. Rome, B. K. and Rome, S. C. Leviathan: An experimental study of large organizations with the aid of computers. SDC document TM-744. August 18, 1962. 83 pp.

Human Data Processing Behavior

1. Binder, A. and Wolin, B. R. Informational models and their uses. Psychometrika, 1964, 29(1), 29-54.
2. Wolin, B. R., Weichel, R., Terebinski, S. J. and Hansford, E. A. Performances on complexly patterned binary event sequences. Psychological Monographs, 1965, 79(7), 1-18.

Team Training

1. Alexander, L. T. and Cooperband, A. S. System training and research in team behavior. SDC document TM-2581. August 16, 1965. 67 pp.

Bargaining and Negotiation

1. Jones, W. H. and Shure, G. H. ITEMSEL - A program to select items by the Wherry-Gaylord method. Behavioral Science, 1966. (Also available as SDC document TM-2444.)
2. Rogers, M. S. and Shure, G. H. An empirical evaluation of item overlap on factorial stability. Journal of Psychology, 1965, 60, 221-232.
3. Shure, G. H. and Meeker, R. J. A personality/attitude schedule for use in experimental bargaining studies. SDC document TM-2543. July 13, 1965. 32 pp.
4. Shure, G. H. and Meeker, R. J. Probing behind the human decision. Proceedings of the 15th Military Operations Research Symposia, 1965. (Also available as SDC document SP-1698/001/00.)
5. Shure, G. H. and Rogers, M. S. Note of caution on the factor analysis of the MMPI. Psychological Bulletin, 1965, 63, 14-18.

Decision Making and Leader Selection

1. Binder, A., Wolin, B. R. and Terebinski, S. J. Leadership in small groups: A mathematical approach. Journal of Experimental Psychology, 1965, 69, 126-134.

2. Binder, A., Wolin, B. R. and Terebinski, S. J. Leadership in small groups: A resolution of discordance. Journal of Experimental Psychology, 1966.

3. Binder, A., Wolin, B. R. and Terebinski, S. J. Learning and extinction of leadership preferences in small groups. Journal of Mathematical Psychology, 1966.

4. Binder, A., Wolin, B. R. and Terebinski, S. J. Leadership selection when uncertainty is minimal. Journal of Psychonomic Science, 1965, 3, 367-368.

EDUCATION AND TRAINING

H. F. Silberman, Head
J. E. Coulson, Associate Head

The Education and Training staff is concerned with the research and development of improved procedures, materials, and computer programs to assist schools in such educational functions as instruction, school administration, counseling, and associated educational data processing.

Early work on computer-based instruction at SDC led to the development of a computer-based laboratory for automated school systems and of associated programs that provide branching, recording, display, and analysis capabilities. Because the effectiveness of machine teaching depends on the quality and organization of the subject matter content presented by the machine, extensive experimentation has been conducted to identify the important variables in instructional materials.

One study compared different procedures for determining self-instructional sequences; a branching method of instruction was found to require significantly less training time for students than the traditional linear method. Another study attempted to find an optimal sequence for presenting instructional stimuli and yielded inconclusive results. In another study, staff members developed a method of improving instructional materials by a succession of evaluation-revision cycles applied to each segment of the material. This method of designing instructional materials is applicable to the development of complete courses and leads to an instructional product that will establish specified skills and competencies in the student.

However, the practical effectiveness of even the most carefully developed instructional material, used with or without a computer, depends in part on factors outside the instructional program itself. Even the most effective laboratory system might not work in an operating school. How the teacher and the school counselor use the system is an important factor. In addition, the organizational and administrative problems that accompany the installation of an instructional innovation must be solved. Members of the Education and Training staff are presently conducting studies in each of these problem areas. Work is also continuing on methods of improving the design of instructional materials.

The problem of installing and testing the effectiveness of new instructional materials and procedures is receiving considerable attention. Previous evaluation studies have been concerned with normative assessment, where the primary objective was to maximize variability in the distribution of scores, rather than with evaluation of the instruction. In an attempt to find a solution to this problem, procedures were developed for constructing criterion instruments that will yield information on absolute levels of mastery. These evaluative instruments were used in installing new materials and conducting field evaluations. The investigation, carried out in the context of a large-scale study of foreign language instruction in California elementary schools, yielded 26 criterion-referenced tests in listening comprehension, speaking, reading and writing.

Another project initiated in 1966 is concerned with adapting the empirical trial-and-revision procedures (for development of instructional materials) to the learning needs of a specific student population in the intact classroom environment. First-grade children in schools with predominantly Spanish-American enrollments are being observed in the classrooms, and trial-and-revision procedures will be applied not only to the development of reading materials but also to the teacher's classroom management procedures and to her organization of the children within the classroom.

In another effort concerned with organizational problems and requirements in education, the Education and Training staff served as consultants to the Research Office of the Los Angeles County School District. Staff members analyzed the organization, objectives, and operations of the Research Office and made recommendations for organizational changes to increase its research potential.

A fundamental obstruction to the use of computers in education has been the problem of cost. Since the cost per student must be greatly reduced if computer applications are ever to become a normal part of the operating school, several projects using computer time-sharing have been initiated. The future of computers in the school is clearly tied to large-scale time-sharing networks that will allow many persons to have direct simultaneous access to a computer at reasonable cost.

One project is using time-sharing to improve the teaching of statistical inference to future research workers in education. A system is being developed that enables college students to conduct a variety of statistical experiments on-line with a computer that gives them diagnostic feedback. In connection with this project, an on-line teaching program (PLANIT) is being developed which has applicability to many content areas. The general-purpose lesson preparation language being developed as a part of this project is more powerful than other similar languages and yet is usable by the nonprogrammer.

Another project is using time-sharing to improve the school counseling function. A man-machine counseling system is being developed that will, among other functions, analyze data from students' records, print lists for the counselor resulting from such analyses, and conduct an automated counseling interview with the student by teletype.

Practical logistics problems that will accompany the innovation of individualized instruction are also being studied. An instructional management information system is being developed that will monitor student performance and provide instructors, counselors, and administrators with better information for making decisions. A computer simulation capability has also been developed that permits designers to construct detailed models of instructional plans. The capability makes it possible to build explicit dynamic models of students' activities connected with their courses. This tool is useful as a planning aid for organizing instructional resources and for purposes of training by simulation.

The next step in this program of educational research and development is to combine a number of the activities into one functional man-machine system. A demonstration of such an integrated system--featuring computer applications for instruction, counseling, scheduling, and administrative planning in a school district--is planned under SDC's contract with the Southwest Regional Laboratory. This Laboratory is presently conducting research and development activities in concert with other educational institutions in Southern California, Arizona, and Southern Nevada, under the provisions of Title IV of the Elementary and Secondary Education Act of 1965 (PL 89-10).

DEVELOPMENT OF A COMPUTER-BASED EDUCATIONAL SYSTEM*

H. F. Silberman, Principal Investigator
J. E. Coulson, Head of materials development effort
J. E. Bratten, Head of instructional management system effort
L. Gallenson L. T. Krebs
C. L. Geddes D. G. Marsh
B. Y. Kooi R. F. von Buelow

Consultants: F. W. Kittinger, Charter Oak School District, Covina; F. A. Yett, Pasadena City College

SWRL Consultants contributing to project: Martin Marchus and Eugene Devine, Institute of Government and Public Affairs, University of California, Los Angeles; Jerry Garlock, Los Angeles County Schools

Description

This project is devoted to the development of products and procedures that will improve the effectiveness of instruction within the operating school. It includes the direct application of computer technology to the instruction process--both in preparing materials and in carrying out instruction--and the technological support of those school functions, performed by teachers, counselors, and administrators, that indirectly facilitate instruction.

Three activities are currently in progress: the design of an instructional management system, the development of instructional materials for use in the instructional management system, and the design of a package of administrative planning aids.

The instructional management system being designed will permit schools to use individualized instructional materials on a large scale. At present, the problem of keeping track of the progress of many students working independently at their own rates prohibits the individualization of instruction. The goal for an instructional management system is to collect detailed data about each student's performance and to prepare displays for teachers telling what problems the student is having and suggesting material that may help him. In addition, the system will be used by curriculum designers to collect data for use in tailoring materials to achieve the instructional objectives of the school.

Development of instructional materials for use in the system requires the engineering of each small segment to achieve specified objectives. Initial work is focused on underlying prerequisite skills (such as reading and generalized problem solving) that cut across many subject areas.

The aim of the administrative planning activity is to develop computer programs and methods to aid school administrators in budget-planning with specific attention to the personnel budget.

Progress

Design of Instructional Management System. Approximately 10 design conferences were held, during which a cadre responsible for the design work hosted a number of specialist consultants. The purposes of the conferences were to specify the operational requirements for the system and to propose initial design solutions. One outgrowth of the conferences was the establishment of special study groups in the areas of equipment, economics, and noninstructional applications of the computer.

An initial version of the instructional management system was designed. In this version, students in a classroom will work on instructional materials for a specified time, answering questions on machine-readable forms. The forms will be collected daily and brought to SDC where they will be used to update the students' data base. Progress reports, containing assignments for the next lesson, will be prepared for each student and returned to the classroom. The instructional management system will also permit

*Supported by the Southwest Regional Laboratory for Educational Research and Development.

teachers to use an interactive teletype station (located in the school and linked to the SDC computing facility), request a specific student's record, and request the system to analyze the student's past performance record. The analysis programs will be designed to identify the specific learning areas in which a student requires help.

Development of Instructional Materials. Initial materials have been prepared, in reading and generalized problem solving, for use in the instructional management system. The materials are designed for children in the normal intelligence range in the 5-6 year age bracket. Behavioral objectives have been specified, and each small segment of the material is being subjected to empirical trial-and-revision procedures. In these procedures, one child is exposed to the first lesson segment, and his responses are used as the basis for revising the segment before it is given to the next child. The material is given to additional children and revisions are continued until the segment consistently achieves its specified behavioral objective. This trial-and-revision cycle is repeated for each successive portion of the lesson.

There are two behavioral objectives in the teaching of beginning reading skills. One is to enable children to read about 50 words generated from about 12 consonants or consonant combinations and three vowels. The second is to develop comprehension skills so that the child can answer who, what, where, and how questions based on the reading of passages consisting of words he has learned. Audiovisual materials have been developed to teach certain initial skills, e.g., visual discrimination of letters, and sounding of individual letters and letter combinations.

In the problem-solving area, the overall objective is to teach children to generate multiple solutions to problems, rather than to stop with the first, most obvious solution. The materials for the initial work in this area consist of sets of cards containing pictures, together with tape recordings of directions to the children. The children are required to arrange each set, containing from two to four pictures, in a time sequence that "tells a story." Responses are marked in answer booklets printed in a special ink that gives color-coded feedback. The program's effectiveness will be judged by whether the children can specify several alternative picture sequences that follow a logical order, and reject sequences that are logically improbable.

Design of Administrative Planning Aids. Steps being undertaken in designing computer-based aids for the administrator include: specification of the data necessary for solving the budget-planning problem; selection of formatting methods for storage of data in the computer; specification of the computational capabilities desired in the programs; determination of capabilities needed for later expansion; selection of a command language to be used for addressing questions to the data; and specification of the hardware to be used. It was decided that the initial version would be written in JOVIAL for the Q-32, that the command language will be user-oriented, and that administrators will be able to use the programs in an interactive mode.

Existing general-purpose systems for handling large data bases were investigated. It appears that the LUCID system, operating on the Q-32 time-sharing system, will be useful for performing some of the personnel planning functions. A set of programs is being written to operate with the LUCID system so that an administrator will be able to perform many essential operations connected with fiscal and administrative management, including the generation of a completed budget form showing the current and following year's entries. He may query and update personnel files, or predict student population at each grade level for the next fiscal year. In addition he may, upon demand, recover part or all of the data used or generated in the program.

Plans

Instructional Management System. A demonstration of the initial version of the instructional management system will be implemented in the fall of 1967. The prototype version will link two elementary schools in the immediate vicinity to the SDC S/360 Time-Sharing System via telephone line. Each school will have a teletypewriter as a means for interacting with the 360.

System development will focus on the use of existing programming systems as contrasted with the writing of special-purpose programs.

Long-range plans point to an operational version of the instructional management system in the early 1970s. This system will link the schools in a given district to a computer at the district office. Each school will have an input/output complex that may include such equipment as an optical scanner, tape storage, a high-speed printer, and a number of interactive consoles. This system will be used for instruction in much the same way as the demonstration version. In addition, the system can be used for administrative planning, for counseling, and for administrative data processing.

Development of Instructional Materials. In the coming year the materials development effort will continue through the sequence of intermediate objectives, with successive evaluation and revision stages, until a satisfactory proportion of children can attain the final objectives with a high degree of reliability. The materials will then be further adapted for use with larger numbers of students in the operational classrooms, as part of the instructional management system. Simple instrumentation, designed to free teachers of much routine instruction, will include printed booklets, tape recorders, and machine-readable answer sheets that can be read into a computer at SDC.

Design of Administrative Planning Aids. Future plans include expansion into a system enabling an administrator to prepare a total budget for his district and to project his expenses for a number of years into the future. In addition, a capability will be added to the system for computing income for the district.

Project Documentation

1. Silberman, H. F. First progress report: Instructional technology module. SDC document TM-(L)-3298/000/00. September 1, 1966. 64 pp.
2. Silberman, H. F. Second progress report: Instructional technology module. SDC document TM-(L)-3298/001/00. December 1, 1966. 50 pp.

COMPUTER-BASED INSTRUCTION IN STATISTICAL INFERENCE*

J. Rosenbaum
J. E. Coulson
H. F. Silberman
} Principal Investigators
S. L. Feingold
C. H. Frye

Consultant: Professor W. J. Dixon, University of California, Los Angeles

Student Associates: B. Brown, Pennsylvania State University; J. Hawkins, University of Utah

Description

Statistical inference is of central importance as a research tool in the behavioral sciences. While the application of statistical methods usually involves nothing more than computation of simple functions of sample data, the intelligent usage of these methods requires an understanding of concepts that are difficult to impart or grasp without the use of fairly advanced mathematics. Most students in the

*Supported in part by the National Science Foundation.

behavioral sciences have very limited mathematical training and understandably find it difficult to grasp the concepts of statistics. One solution to this problem, which is being used to a very limited degree in current statistical training, is to demonstrate statistical concepts through empirical sampling experiments. Such experiments are very time-consuming in the traditional classroom or statistical laboratory; however, they are markedly less so in a computer-based setting. Thousands of samples from a variety of populations can be produced in the computer in less than one minute. In the computer-based setting the student can see, for example, that stratified sampling does indeed reduce the variability of an estimate of a population average over that estimate obtained from simple random sampling. The student is more likely to remember these experiences than he is those statements he must take on faith from lectures or textbooks. In addition, the computer shows great promise in a tutorial role, particularly when the student is put on-line in a time-sharing system. The time-shared computer can be programmed to interact with the student, posing questions for him to answer or presenting exercises to be worked. The student submits his answers for evaluation and within a few seconds is informed by the system whether his responses are correct or incorrect. (He can also spare himself much computational drudgery by using the mathematical calculation features of the system to work through the exercises.) Perhaps the most important tutorial role the system can play is to branch the student to more advanced materials, to remedial sequences, to specific textbook references or to his instructor, as requirements for these are indicated by the student's responses.

Progress

A two-year project has been initiated to develop computer-assisted instruction in statistics for use at UCLA. This project is now at midpoint. In the first year a software system called PLANIT* was developed and used to prepare and present instructional material. Both the system and the materials were tried in shakedown tests with seven undergraduates at the University of Southern California during the 1966 summer session. More systematic field testing has been scheduled for the second year with students enrolled in introductory statistics courses for the social sciences at UCLA and possibly at USC. A student station (teletypewriter) has been installed at each of these two universities; each is linked by telephone lines to the SDC Time-Sharing System in Santa Monica.

PLANIT has been used to prepare (and subsequently to edit) four instructional sequences or lessons:

1. INTRO is an introductory lesson designed to teach the student how to communicate with his mechanical tutor and calculational assistant. The average student can communicate quite effectively with subsequent lesson materials after spending 45 minutes to an hour with INTRO.

2. DESCRB is a lesson in descriptive statistics covering (1) the construction of frequency distributions, and the calculation of means, medians, and variances from sample data; (2) their utility in describing and summarizing a set of experimental observations. The student is given further practice in the use of calculation conventions and symbols in the numerical exercises. (For example, sets of pseudorandom scores are generated and presented to him. He is then required to obtain numerical values for the average score, or the variance for a set of scores.) About an hour at the teletypewriter was required for completion of this lesson in shakedown tests.

*PLANIT (Programming LANguage for Interactive Teaching) is described on p. 7-8.

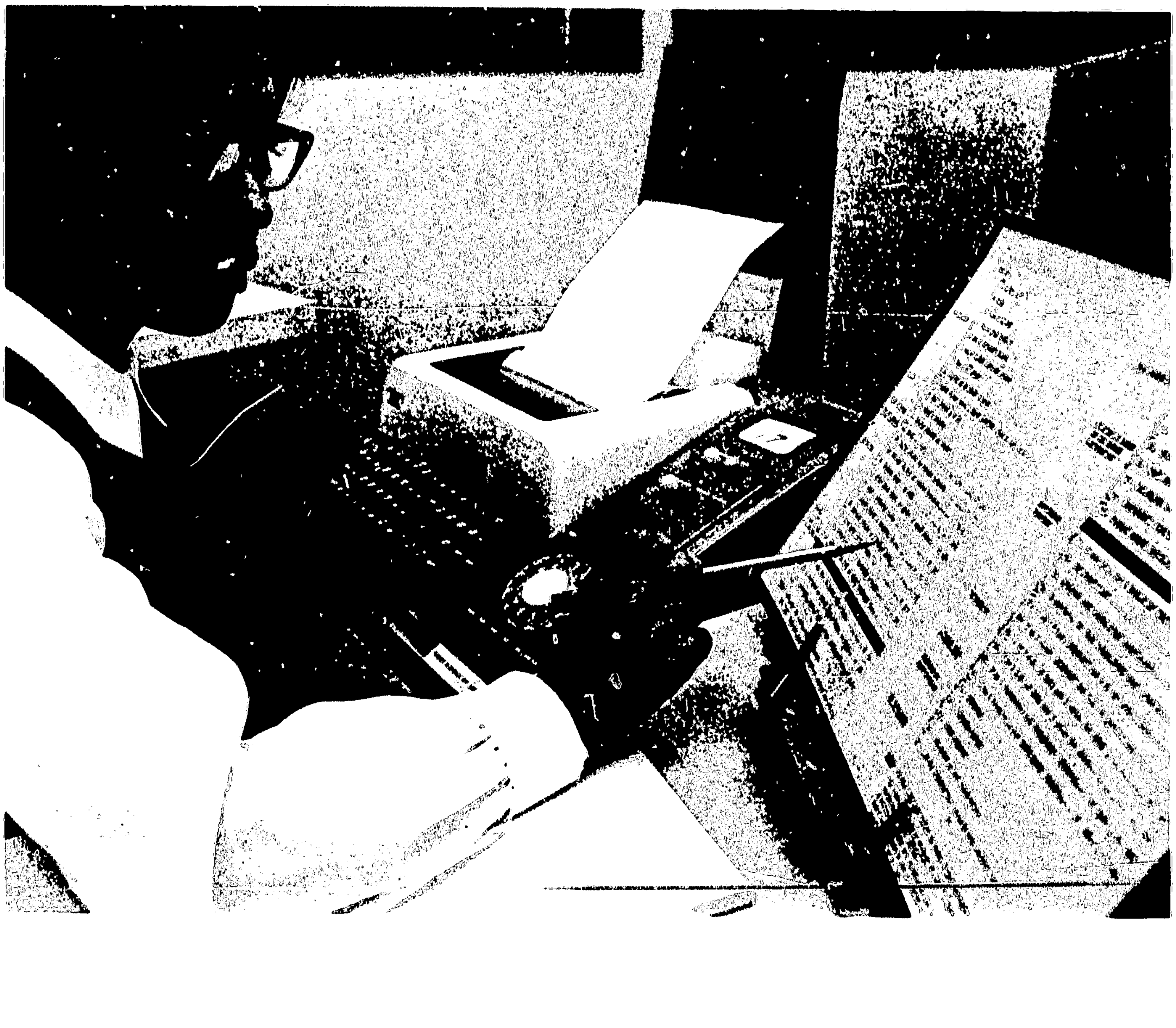


FIGURE 7-1. STUDENT WORKING A LABORATORY EXERCISE ON STATISTICAL INFERENCE

3. PROB1 is a tutorial sequence in elementary probability. After a brief introduction to elementary set theory, the notion of an event is shown to be an application of the notion of set. The student is exercised in translating from verbal descriptions of composite events to the equivalent but more "mathematical" description of these events as sets of possible outcomes of a sampling experiment. The lesson concludes with applications of the addition theorem for probability. The students used in shakedown tests required an hour to one and one-half hours to complete this lesson.

4. NORM is a lesson on the properties of the univariate normal distribution and the use of normal distribution tables in statistical work. This lesson has not yet been edited or tried with students.

The students who participated in the shakedown tests were uniformly enthusiastic about this mode of instruction. None had any previous exposure to a computer and part of their enthusiasm may be due to the novelty of their experience. However, they exhibited certain behavior which indicates that this enthusiasm may persist even when the novelty has worn off. For example, they were very appreciative of the ease and speed of computation with PLANIT as contrasted with calculations on a desk calculator. Also when they were engaged in a question/answer dialogue, their absorption was intense and their motivation high, as evidenced by their persistence in working on a problem until PLANIT "informed" them they were correct. On the negative side, they were quite vocal about their dissatisfaction with unanticipated delays in their instruction. These delays were caused by breakdown or malfunction in some aspect of the time-sharing system or in the communication link to it.

Plans

1. Additional lesson material will be prepared covering the elements of statistical hypothesis testing and exercises in their use. A two- to three-hour lesson sequence designed to teach some aspects of on-line programming will also be attempted; this sequence will conclude with a programming exercise requiring the student to prepare a complete program for the calculation of a statistic such as the Pearson product moment correlation coefficient.

2. Systematic field tests of the lessons described above, augmented by additional sequences, will be carried out during 1966-67 with approximately 20 students. The students will be pretested to determine their proficiency in the topics to be covered in the computer-based material. They will then be posttested following their computer-based instruction to measure the learning which has occurred.

Project Documentation

1. Frye, C. H. and Rosenbaum, J. Student's guide to STAT. SDC document TM-2910/000/00. March 31, 1966. 172 pp.

PLANIT: Programming LANguage for Interactive Teaching*

S. L. Feingold
C. H. Frye

Description

The objective of this activity is to provide a Lesson Designer (LD), or teacher, with a powerful and flexible tool for entering and changing course content in the computer, for presenting this material to the student, and for prescribing the behavior of the computer in connection with a student's response. The language used must be simple enough to allow a nonprogrammer to communicate his intent to the computer.

Progress

The initial design started in January of 1966. PLANIT became operable in June of 1966. It uses

*Supported in part by the National Science Foundation.

	Dialog	Explanation
User:	*CO	*COmmence operation in the lesson building mode. The asterisk is printed by the system when a user input is required.*
System:	P/Q/M/D/C	*The system is asking the user—in the concise form—to select the type of lesson frame he will build.*
User:	*?	*The user requests explanation of the above printout.*
System:	(P)ROBLEM/(Q)UESTION/(M)ULTIPLE CHOICE/ (D)ECISION/(C)OPY	
User:	*Q	*The user specifies the Question frame.*
System:	1. FRAME 25.00 LABEL = * HISTORY	*The system assigns a number to the lesson frame (25). The user assigns a label (HISTORY).*
System:	2. SQ.	*Specify Question.*
User:	*WHO INVENTED THE ELECTRIC LIGHT?	
System:	3. SA.	*Specify anticipated student Answers.*
User:	*A+ EDISON *B MARCONI *C BELL	*The + after A indicates to the program that this is the correct answer.*
System:	4. SAT.	*Specify Action to be Taken, depending on answer given by the student.*
User:	*A+ F: GOOD WORK. B: 5 *B R: THAT WAS THE WIRELESS TELEGRAPH. TRY AGAIN. *C R: BELL WAS THE TELEPHONE MAN. TRY AGAIN. *- C: B: 1	*The user specifies a Feedback command to be executed if the student answers correctly; then the program is instructed to Branch to Lesson Frame 5. A Repeat command is given in case student gives answer B or C. If student gives any other response, the program is instructed to print the Correct answer and to follow this by Branching back to Lesson Frame 1 for remedial instruction.*

FIGURE 7-2. A TYPICAL INTERCHANGE USING PLANIT

A dialog between the computer and a user who wishes to construct a lesson is shown above. As illustrated, the lesson builder may combine PLANIT commands in a number of ways.

the IBM Q-32 computer by means of an interactive console under the SDC Time-Sharing System.

Some important features are:

1. Interaction. PLANIT interacts with the Lesson Designer (LD) as well as with the student. The program, at each step, "prompts" the LD, who can choose the level of detail the prompting should take. New users want much guidance, whereas experienced users prefer short prompts.

2. Interpretation. The interpretive feature of the program ensures that there is no waiting time for compilation before the lesson can be used. It also allows PLANIT to catch errors immediately while the LD is entering course content. The LD can enact the role of student or switch to another mode to edit the lesson at any point. Turnaround time between changes is usually less than one second.

3. Calculation Capability. PLANIT has an on-line calculational capability that allows the LD and the student to perform calculations involving trigonometric functions, elementary algebraic functions, and matrix computations. In addition, the student's calculations can be compared with the results of some function previously defined by the LD.

4. Criterion Branching. The PLANIT language allows the LD to specify conditions for branching based on students' performance over any portion of the lesson involving response latency, errors, help received, etc.

5. Service Routines. PLANIT also provides service functions for evaluating student answers that depart from the expected response by making phonetic comparisons between the correct answer and student's response, keyword search of student's answer, and algebraic equivalent matching. This last routine allows the student credit for an answer that is algebraically equivalent to any anticipated response; e.g., if a problem involving temperature has the anticipated answer 5/9(F-32), then 5(F-32)/9 or 1/9(5F-160) would be acceptable. The LD can have any combination of the service routines turned off and on at will (even between anticipated answers). Figure 7-2 shows a lesson design.

Materials currently in production via PLANIT are: statistics (aimed at graduate and undergraduate students in psychology--see p. 7-5); spelling and vocabulary (for children, 3-8 years of age--see p. 7-4); economics (for undergraduate students in economics). Agencies who have used, or are using, PLANIT are: System Development Corporation; Southwest Regional Laboratory; University of California at Los Angeles; University of Southern California; University of California at Irvine; United States Naval Personnel Research Activity, San Diego, California; NEEDS (New England Educational Data Systems), Cambridge, Massachusetts; Lackland Air Force Base, San Antonio, Texas.

Since PLANIT was conceived as a language designed for computer/human interaction, rather than for teaching alone, its use is limited only by the user's imagination. For example, the development of a vocational counseling interview using this system is well under way.

Plans

PLANIT is being converted to run under SDC's S/360 Time-Sharing System. Initial efforts are concentrated on producing equivalent capability on the 360 computer. The language will then be increased but will contain the original Q-32 language as a subset. All material written under the Q-32 will run on the 360. Future goals include use of a cathode-ray tube and a Graphic Input Tablet for lesson building, editing, and presentation. Additional capabilities for lesson presentation will include slides and tape recorders under control of the lesson.

Project Documentation

1. Frye, C. H. and Feingold, S. L. User's guide to PLANIT: Programming LANguage for Interactive Teaching. SDC document TM-3055/000/01. October 17, 1966. 214 pp.

VOCATIONAL COUNSELING*

J. F. Cogswell, Principal Investigator
C. P. Donahoe, Jr.
D. P. Estavan
B. A. Rosenquist

Consultants: Dr. David Tiedeman, Harvard College; Dr. John Loughary, Oregon System of Higher Education; Dr. C. Gilbert Wrenn, Arizona State University; Dr. William Schill, University of Illinois; Dr. Margaret Crawford, Los Angeles Trade and Technical College; Dr. Melvin L. Barlow, University of California, Los Angeles; Mr. F. Parker Wilbur Los Angeles Trade and Technical College; Dr. G rard Haigh, Psychological Service Association, Westwood, California

Description

The objectives of the study are to survey and report counseling and related information processing practices in the vocational education field; to explore ways in which information processing technology can be applied to aid the counselor by relieving him of certain routine information processing functions and by providing him with additional information on which to base his decisions; and to develop and evaluate an experimental computer-based man-machine counseling system.

The need for occupational information and for vocational guidance and counseling is explicit in most statements of goals and purposes concerning decisions about vocational education. Persons making these decisions should have access to the counsel of specialists who understand the world of work and its complexities. The ideal counselor, then, is a person who has a professional background in pupil personnel services and who is, at the same time, a specialist in occupational information, vocational guidance, and counseling.

To fill these roles effectively, a vocational counselor must be able to collect, organize, and use large amounts of detailed information about many different students and vocations. The information processing task alone has been greatly complicated by an increase in student population, by the accompanying growth of student information, and by a rapidly changing pattern of vocational fields. In addition, progressively greater amounts of valuable counselor time are being spent in collecting, processing, and maintaining fairly routine information, so that there is less time for such vital activities as personal interviews with individual students.

Some solutions to this problem have already been explored. During the past few years, studies at SDC have been concerned with ways in which the computer might be used to assist in the functions of student appraisal and student interviewing. These studies suggest the feasibility of employing the computer to assist in counseling operations. However, before widespread application can be encouraged, many problems must be considered. Automated procedures, if not implemented carefully, could tend to force rigid standardization in the counseling procedures. Techniques of design and implementation must be developed that allow the individual counselors to maintain their unique ways of doing things, at the same time providing them with the advantages of advanced information processing technology. Another problem that must be considered is the reaction of counselees to automated procedures. Care must be taken so that the use of automated procedures will not interfere with the counseling relationships.

In the present study, 12 vocational education installations have been surveyed. One of these was selected for the detailed study.

A computer-based man-machine counseling system will be developed by a planning team consisting of counselors at the field site, researchers, and consultants who are recognized authorities

*Supported in part by the Office of Education, U. S. Department of Health, Education, and Welfare.

```
PLEASE TYPE YOUR NAME AND ID NUMBER

* DEBORAH ANDERSON 42

ACCORDING TO MY RECORD THIS IS HOW YOU DID LAST SEMESTER.

CODE  COURSE                   GRADE
 1     ENGLISH 1                 A
 2     GENERAL SCIENCE           A
 3     EUROPEAN HISTORY          C
 4     ALGEBRA                   B

IS THIS CORRECT?  IF SO, TYPE  YES  IF NOT, TYPE  NO

* YES

ARE YOU HAVING ANY PROBLEMS WITH ANY OF THESE COURSES?  YES/NO

* YES

TYPE THE NUMBER OF EACH COURSE THAT IS GIVING YOU TROUBLE.  WHEN
YOU ARE ALL THROUGH, TYPE  /

* 3/

WHAT KIND OF PROBLEM DO YOU HAVE WITH EUROPEAN HISTORY?
PLEASE TYPE IN THE NATURE OF THE PROBLEM.  A FEW WORDS WILL BE O.K.
WHEN YOU ARE ALL FINISHED, TYPE  /

* I DON'T TURN MY WORK IN ON TIME./

I SEE... WOULD YOU LIKE TO CONTINUE WITH THIS INTERVIEW AT THIS TIME,
OR WOULD YOU LIKE TO DISCUSS YOUR PROBLEMS WITH YOUR COUNSELOR BEFORE
YOU CONTINUE?  TYPE  GO IF YOU'D LIKE TO CONTINUE.  TYPE  STOP  IF
YOU'D LIKE TO STOP.

* GO

DO YOU PLAN TO CONTINUE YOUR FORMAL EDUCATION BEYOND HIGH SCHOOL?

* YES

WHICH OF THE FOLLOWING IS MOST LIKE THE SCHOOL YOU PLAN TO ATTEND?
1  IVY LEAGUE COLLEGE
2  OTHER COLLEGE OR UNIVERSITY
3  CALIFORNIA STATE COLLEGE
4  JUNIOR COLLEGE
5  BUSINESS OR VOCATIONAL SCHOOL
6  I HAVE NOT YET DECIDED
TYPE THE APPROPRIATE CODE.

* 3

DO YOU INTEND TO MAJOR IN ENGINEERING OR ONE OF THE SCIENCES?
YES/NO

* YES

HERE ARE SOME AREAS OF SCIENCE AND SOME EXAMPLES

CODE          AREA                               EXAMPLES
  1   PHYSICAL SCIENCES......PHYSICS, CHEMISTRY, GEOLOGY, ETC.
  2   BIOLOGICAL SCIENCES....BIOLOGY, PRE-,ED, PRE-NURSING, ETC.
  3   SOCIAL SCIENCES........PSYCHOLOGY, SOCIOLOGY, ANTHROPOLOGY, ETC.
  4   MATHEMATICS............MATH, STATISTICS, COMPUTER PROGRAMMING,
  5   ENGINEERING............ARCHITECTURE, DRAFTING, ELECTRONICS, ETC.

PLEASE TYPE THE CODE NUMBER OF THE KIND OF SCIENCE YOU INTEND TO
MAJOR IN.

* 3

ALRIGHT.  LET'S SEE IF WE CAN PINPOINT YOUR AREA
1  PSYCHOLOGY
2  SOCIOLOGY
3  ANTHROPOLOGY
4  SOCIAL WORK
TYPE THE APPROPRIATE CODE.

* 1

OK.....
THE GRADES YOU GET IN HIGH SCHOOL LARGELY DETERMINE THE TYPE OF
COLLEGE YOU WILL BE ABLE TO ATTEND.
LET'S JUST TAKE A LOOK AT YOUR PROJECTED HIGH SCHOOL GRADES
BASED ON YOUR PRESENT PERFORMANCE.
STUDENTS WHO GOT GRADES LIKE YOURS IN JUNIOR HIGH, DID AS FOLLOWS
AT SAN FERNANDO VALLEY STATE COLLEGE--
70 PER CENT EARNED  A OR B
20 PER CENT EARNED  C+
10 PER CENT EARNED  C

ON THE BASIS OF YOUR PROJECTED GRADES, YOUR PRESENT COLLEGE CHOICE
INDICATES THAT YOU ARE THINKING PRETTY REALISTICALLY ABOUT YOUR
FUTURE LIFE.
```

FIGURE 7-3. A PORTION OF A SAMPLE AUTOMATED COUNSELING INTERVIEW

This automated counseling program resulted from an earlier, exploratory SDC study. A printout of the dialog between student and computer can be furnished to the counselor for evaluation and followup.

in the areas of vocational education and counseling. The system will have the following features:

. A data base containing student information with an input/output system for keeping it up-to-date and for retrieving information from it.

. Automated procedures to handle or assist in functions such as appraisal of student data, interviewing of students, tracking of student performance, and identification of students who are experiencing difficulty.

Progress

Survey of Counselor Practice. Eighty-seven counselors from 12 different educational settings were interviewed to obtain a picture of the variations in counseling practice. In addition, each counselor described, by using a set of Q-sort cards, how he was actually spending his time and how he would ideally like to spend his time. The educational settings included vocational high schools, trade schools, junior colleges, and academic high schools.

Analysis of the Q-sort data indicates that there are no significant differences, between one site and another, in the kinds of responses as a whole. However, the data do indicate a marked difference between what the counselors are doing and what they would like to do: As a group they would prefer to reduce greatly the amount of time that they must devote to routine information processing tasks and increase the time they devote to working with students. They would like more time for following up their students, working with administrators in improving curricula, and conducting research. They would like to see an increase in the use of data processing equipment, presumably to relieve them of the information processing task.

Selection of the Experimental Field Site. After the collection of survey data, an experimental field site was selected: a large high school/junior high school complex in the Los Angeles School District. The high school has a population of 5,000 students with 10 counselors; the junior high has 1,800 students and five counselors.

System Analysis of Counseling Procedures. A detailed description of all of the counseling procedures employed at this school complex was obtained by interviewing all of the 15 counselors. General flow of procedures for each of the two counseling subsystems was defined. In addition, the idiosyncratic functions and procedures of each counselor were described. These descriptions were reviewed by the counselors to clarify omissions and misconceptions.

Training of Counselors. After the system analysis, the counselors came to SDC for three two-hour workshops on advanced information processing technology. Several demonstrations of automated interviewing, automated teaching, and on-line computer applications in information retrieval were presented. Discussion and readings were combined with the demonstrations.

The Initial Design Workshops. When the training was completed, the counselors and the four experimenters worked together in formulating the specifications for the initial model of the man-machine system.

The research staff met for 21 hours with the high school counselors and for 12 hours with the junior high counselors. The following ideas for the system emerged from the design workshops:

. Storage of all student data in the computer for ready processing and retrieval.

. Tracking and monitoring of student progress to alert counselors to possible problems.

. On-line generation and application of multiple regression formulas for predictive and research analysis.

. Automated generation of all routine reports such as grade and progress reports.

Automated interviews to help students in the areas of post-high-school planning, course programming, and vocational exploration.

Laboratory Development of Systems. Laboratory demonstrations of some of these information processing procedures are being developed. Work is progressing on the development of the information storage and retrieval system and on an automated interview for exploring occupations and related educational requirements.

Plans

In the spring of 1967 the specifications for the Model I machine and human functions will be crystallized into a design proposal for the second phase of the project. In this phase the information processing applications will be developed and implemented. The counselors will be trained in system procedures; in addition, a counselor training program, including sensitivity training, will probably be developed and used to strengthen counselor skills in the human aspects of their work. The counselors will then use the system and evaluate it.

When the system has been implemented and is operating smoothly, the final evaluation data will be collected for the project.

It is hoped that both the system development procedures employed in this study and the operational innovations will provide a useful model for the development of man-machine counseling systems in other installations.

Project Documentation

1. Cogswell, J. F., Donahoe, C. P., Jr., Estavan, D. P., and Rosenquist, B. A. The design of a man-machine counseling system. SDC document SP-2576/001/01. September 30, 1966. 17 pp.

2. Cogswell, J. F. and Estavan, D. P. Explorations in computer-assisted counseling. SDC document TM-2582/000/00. August 6, 1965. 29 pp.

EVALUATION AND REVISION OF CLASSROOM PROCEDURES*

R. J. Melaragno
G. Newmark

Description

Attempts to solve educational problems have either taken a research approach or attempted more immediate "practical" solutions. However, research projects often lead only to reports, and "practical" efforts seldom have research evidence to support them.

In order to bridge the gap between educational research and practice, two conditions must be met: (1) research must be guided by a product orientation, since experience indicates that a tangible product is needed to bring about educational improvements; and (2) the product must be developed by applying a procedure involving successive evaluations and revisions of the product until it is known to accomplish its objectives.

This project is designed as a step toward the solution of the problem of translating research findings into practice. The study is focused on a specific target population of learners (culturally disadvantaged Mexican-American students) and a specific content area (the development of a first-grade reading skill).

An evaluation-revision procedure developed earlier at SDC is being used as the model for this study. (See Completed Studies, TM-895/200/00, p. 7-21.) The procedure calls for the following steps: (1) operational definition of the instructional objective; (2) development of instruments to assess the attainment of this objective; (3) specification of intermediate goals leading to mastery of the objective; (4) selection and/or development of initial materials and procedures designed to achieve the objective; and (5) empirical improvement of materials and

*Supported in part by The Fund for the Advancement of Education, Ford Foundation.

procedures, through repeated trials and revisions, until the instructional objective is attained by a large proportion of the target population.

The study is being performed in two phases. The first phase involves extensive classroom observation and testing to isolate a specific reading problem for Mexican-American students and to facilitate the choice of materials that show the most promise of overcoming the problem. The second phase involves adapting and improving instruction through the systematic investigation and use of the successive evaluation-revision procedure model in the classroom.

Progress

Two Los Angeles City elementary schools with populations of 80-90% Mexican-Americans have been selected for the study. School administrators and 10 teachers have assisted in preliminary planning of the study.

The first phase, now in progress, includes regular observations of reading instruction in first-grade classes and some observations of reading instruction in second-grade and remedial-reading classes. Tests are being developed, or purchased and modified, that will give diagnostic information about specific reading problems of Mexican-American students. A body of research is being collected on approaches to reading instruction for Spanish-background students, to assist in the selection of initial materials and procedures.

Plans

The second phase of the study will be conducted during 1967. Two principal end products are expected: (1) development of procedures for empirically adapting classroom instruction for a particular set of learning objectives and a particular student population, using the evaluation-revision model; and (2) demonstration of the effectiveness of the model by manifesting improved reading instruction for culturally disadvantaged students--comprising a segment of society in which the need is great.

FOREIGN LANGUAGE STUDY*

G. Newmark
H. F. Silberman } Principal Investigators

Consultants: J. D. Bowen, University of California; P. Pimsleur, Ohio State University; S. Sapon, Rochester University; Y. Lightfoot, Pasadena City School District; P. Herrera, Santa Monica School District; M. Smith, Los Angeles School District; F. Almeida, Los Angeles School District

Description

The purpose of this project was to develop absolute assessment procedures as a basis for making decisions about selecting a language course, developing improved instructional materials, modifying learning conditions, and revising course objectives. It was designed to obtain greater objectivity in evaluating achievement in the four basic language skills (i.e., listening, speaking, reading, and writing) than had been possible with existing methods.

Progress

A field test (involving 60 elementary schools and approximately 1,800 pupils) of three different approaches to teaching Spanish to sixth-grade students was conducted in California schools. The three approaches were: (1) instruction by a qualified foreign language teacher (MLA language course); (2) programmed self-instruction (SPA language course); and (3) instruction by television (UAE language course). Different tests were developed for the three Spanish courses (MLA--8; SPA--16; UAE--4), 28 tests in all. These tests were unique to each course of study. They were not used to compare the three courses of instruction with each other, but were designed to evaluate

*Supported in part by the California State Department of Education under a grant from the U. S. Office of Education.

the extent to which each language course achieved its own objectives under specified learning conditions. The test instruments are referred to as criterion-referenced tests, since they are based on an absolute criterion rather than on a relative standard of achievement.

This study demonstrated the feasibility of criterion-referenced testing as applied to a large-scale field test of three foreign language courses. It indicated that the traditional method of evaluating only a small sample of the specific linguistic objectives of a language course may obscure serious deficiencies in materials and/or learning conditions. The test results in the study showed that, with the exception of a few specific goals, all three language courses failed to achieve their basic linguistic objectives under the learning conditions involved in this study. Substantial modifications to materials and/or learning conditions would be required to achieve the basic goal of foreign language instruction--enabling almost all students to acquire sufficient mastery of the language skills to profit from the next sequence of instruction.

The field evaluation portion of the study was completed in 1965. The final report was completed during the first half of 1966 and is being published by the California State Department of Education.

Project Documentation

1. Newmark, G. and Diller, E. Emphasizing the audio in the audio-lingual approach. Teaching English as a Second Language, 1965, 1(46), 353-356.

2. Newmark, G., Melaragno, R., Silberman, H. The development of criterion-referenced tests in four skills for field testing three approaches to teaching Spanish in elementary schools. SDC document TM-3063. July 22, 1966. 57 pp.

3. Newmark, G., Sweigert, R., Mueller, K., Melaragno, R., Johnson, D., and Silberman, H. Final report: U. S. Office of Education Cooperative Research Project D-177. A field test of three approaches to the teaching of Spanish in elementary schools. Sacramento, California: State Department of Education (in press).

METHODS OF PRESENTING INSTRUCTIONAL MATERIAL*

R. J. Melaragno

Description

This study, which was completed in 1966, examined two methods for adapting self-instructional materials to individual differences among learners. The two adaptation methods were compared with each other and with a third method involving minimal adaptation.

Most programmed instruction provides for individuality among learners only with regard to the amount of time spent on instruction. Such programming has been called "linear," since all learners progress through the program in a straight-line fashion, receiving identical sequences of instruction. An alternate form of programmed instruction has evolved through the introduction of the digital computer into the physical makeup of teaching machines. Called "branching," this procedure involves keeping track of a learner's performance data by means of a computer, which modifies the sequence of instruction on the basis of prior performance. Recently, proposals have suggested using pre-instructional characteristics of learners as criteria for the selection of appropriate instruction. These proposals involve a procedure, called "prediction," in which information about the learner is input to the machine and used for specifying instructional sequences.

*Supported in part by the Office of Naval Research.

Both branching and prediction procedures accommodate learner individuality by varying the amount and kind of instruction as well as by allowing varying time for instruction. In the present study, branching and prediction procedures were compared with each other, and with the minimally adaptive linear procedure. The study was conducted in two phases: During Phase I, branching and prediction strategies were developed; during Phase II, the three instructional methods were compared.

Progress

Phase I. The instructional program dealt with geometric inequalities. Thirty-two high school students were administered a battery of seven pretests. Six tests were drawn from Guilford's Structure of Intellect model and were felt to be logically related to learning of geometry; the seventh was a test of geometry fundamentals.

After the pretesting, the 32 subjects received the inequalities program individually or in very small groups. The experimenter observed the subjects in order to determine locations in the program where branching was needed. Two distinct types of branching points were determined from these empirical trials: locations in the program where subjects could skip redundant instruction because their prior performance was satisfactory; and locations where subjects' performance indicated the necessity for additional remedial instruction. The branching structure that evolved had 9 points where skipping could occur and 14 points where remedial material could be provided.

After the empirical trials were completed and the branching structure was determined, the experimenter scored all seven pretests for the participating subjects and related the pretest scores to performance on the program. Multiple cutoff technique was found to discriminate best between subjects who branched or did not branch at each of the 23 decision points in the program. Application of the multiple cutoff technique resulted in the elimination of two pretests and the use of the remaining five as elements in prediction statements.

Phase II. In the second phase the procedures empirically developed in Phase I were examined in SDC's computer-controlled instructional laboratory (CLASS). Forty-four high school students were assigned randomly to three treatment groups, administered the five pretests, and provided with instruction under one of three methods. Subjects in the Linear group all received a common set of instructional materials, composed of 285 items. Subjects in the Prediction group had their sequences of instructional materials predicted on the basis of their scores on the five pretests. The Branching group subjects had their unique, individual sequences determined on the basis of their responses to certain items in the program.

The amount of time each subject spent during instruction was recorded, and each subject was administered a posttest on geometric inequalities. The two dependent variables, training time and posttest score, were analyzed simultaneously by multiple analysis of variance; a significant difference beyond the .005 level was obtained. Comparison of treatment pairs showed no difference between the Linear and Prediction conditions, no difference between the Prediction and Branching conditions, and a significant difference between the Linear and Branching conditions in favor of Branching.

Further analyses of each dependent variable yielded the following results: There were no differences among the three groups on posttest scores; the Prediction condition required significantly less training time than the Linear condition, and the Branching condition significantly less time than the Prediction condition.

The results lead to three conclusions:

1. The strategy of varying instruction on the basis of learners' abilities can reduce training times significantly.

2. A branching strategy can take significantly less training time than either prediction or linear strategies.

3. When both the amount learned and the training time are of interest, branching is significantly superior to a linear presentation.

Project Documentation

1. Melaragno, R. J. Final report: A study of two methods for adapting self-instructional materials to individual differences. SDC document TM-2932/000/01. June 22, 1966. 51 pp.

SIMULATION OF A FLEXIBLE SCHOOL*

J. F. Cogswell, Principal Investigator
J. E. Bratten
D. G. Marsh

Consultants: R. L. Egbert, Brigham Young University; F. A. Yett, Pasadena City College

Description

The purposes of this study were: (1) to develop procedures for the analysis and computer simulation of instructional systems; (2) to explore and define uses for system analysis through the conduct of studies at five selected innovating secondary schools; and (3) to draw conclusions from the results of these analyses about some of the major problems faced by schools in their attempt to make instruction more responsive to individual differences among students.

Progress

The names of 200 secondary schools involved in instructional innovation were obtained both from state departments of education and from recognized leaders in education. Questionnaires were sent to obtain detailed information; 89 of these questionnaires were returned. Twenty-six schools were visited by project personnel to collect amplifying information. The data analysis resulted in selection of five schools as subjects for the system analysis studies: Brigham Young University Laboratory School, Provo, Utah; Nova High School, Ft. Lauderdale, Florida; Theodore High School, Theodore, Alabama; Buena Vista High School, Saginaw, Michigan; and Garber High School, Essexville, Michigan.

Each of the five schools was visited several times and data were collected that enabled the investigators to define the system characteristics of an exemplary course or department at the school. Each such system was viewed as a sequence of discrete activities through which students were processed according to rules governing their movement. A set of computer programs (EDSIM), using the JOVIAL programming language on a Philco 2000 computer, was written and used to model the instructional systems. Nine separate system analyses were conducted during this project.

Seven of the systems analyzed were modeled on EDSIM; two were described by verbal models. In each simulation study, individual simulated students were processed through simulated activities in accordance with the decision rules for processing students through the course. Recordings were made of the pathway of each simulated student through the system and summaries were printed out by the computer. In five of the studies, the simulation used rules employed in the actual instructional systems and the validity of simulation was demonstrated by comparing simulated results with data from the actual system. In two studies, theoretical rules (representing proposed systems) were simulated in order to predict the effects of the proposed systems on students.

*Supported in part by the Office of Education, U. S. Department of Health, Education, and Welfare.

Two uses for system analysis were identified in the study. The first was in defining current problems in an existing instructional system and proposing changes designed to remedy the problems. Simulation was used to show the effects of the changes. The second use for system analysis was in exploring new ideas for future instructional systems. In this case, simulation concretized the ideas and predicted the resources needed for implementation.

The study identified three major problems faced by schools that are attempting to make instruction responsive to the unique requirements of individual students: conflicts between group and individual instruction, lack of resources, and lack of adequate information about students.

1. Conflicts Between Group and Individual Instruction. Some schools dedicated to meeting the diverse needs of individual students have purchased expensive electronic systems that, at best, provide only a few channels that will allow different students to receive different material at the same time. Several of the schools permitted students to study, prepare assignments, and take tests in an individual mode; in most of these schools, however, deadlines were set or procedures were used that encouraged the students to pace themselves so that the class was fairly homogeneous. As long as teachers feel that they must meet students in groups or that students must finish a course by the end of the year, procedures will be used that are inimical to individual progress.

2. Lack of Resources. The schools included in the study required human resources beyond their normal teaching staff to accomplish their objectives. The prospect of providing adequately designed materials for individualized instruction appears dim unless substantial support external to the schools is forthcoming. Three other critical problems concerning resources in innovative schools were uncovered by the study: (1) the amount and kind of space needed; (2) the assignment of teaching personnel; and (3) the scheduling of students.

3. Lack of Adequate Information About Students. It is unlikely that individual progress courses can operate on a large scale in any school without access to a computer-based system to keep track of the progress of students. Extrapolation from data produced by simulating a plan for modified individual progress indicates that with a population of 900 students, school personnel would have to schedule 30 to 40 individual changes between courses each day in addition to administering about 300 individual mastery tests each day. What is needed to facilitate individualized instruction is an instructional management system to: (1) collect and store information on student expectancies and student performance; (2) provide instructors with displays on student progress so that their time can be allocated to the most pressing student needs; and (3) provide course designers with detailed data about the effectiveness of their instructional techniques.

The study was completed in March 1966 and the final report for the project was completed on April 19, 1966. The work is being continued in connection with an SDC contract with the Southwest Regional Laboratory (see p. 7-3).

Project Documentation

1. Cogswell, J. F. Systems analysis and computer simulation in the implementation of media. Audiovisual Instruction, 1965, 10(5), 384-386.
2. Cogswell, J. F. The design of man-machine systems in education. SDC document SP-1856/000/01. November 10, 1965. 9 pp.
3. Cogswell, J. F. and Bratten, J. E. Plan for final phase of school analysis and simulation study. SDC document TM-1493/180/00. March 4, 1965. 15 pp.
4. Cogswell, J. F., Egbert, R. L., Marsh, D. G., and Yett, F. A. Construction and use of the school simulation vehicle. Journal of Educational Measurement, 1965, 2(1), 5-14.

5. Cogswell, J. F., Bratten, J. E., Egbert, R. L., Estavan, D. P., Marsh, D. G., and Yett, F. A. Final report: Analysis of instructional systems. SDC document TM-1493/201/00. April 19, 1966. 276 pp.

COMPUTER-ASSISTED TEACHING OF MATHEMATICS

J. Staudhammer, Consultant at SDC
(Arizona State University)

Description

The purpose of this project is to explore the possibilities of improving the teaching of applied mathematics by computer assistance. By the use of carefully designed programs, the scope of material covered can be extended and many more problems can be solved in a given length of time. When the machine carries the burden of lengthy, routine calculations, the student can spend his time more profitably on the technique of problem formulation and on testing his methods and solutions. The basic expectation is twofold: (1) The student can do much more of what he is now doing; and (2) he can do things far beyond his scope at present. To achieve this expectation, it is not sufficient merely to automate some well-known mathematical procedures; rather, the power of the machine must be coupled to the manner and order of presentation of the material.

Progress

Numerous small matrix programs have been written and checked out. These programs trace the development of the simple triangularization procedure for finding determinants to a fairly sophisticated in-place inversion routine. Three routines were developed that mechanize and streamline the following basic matrix operations: (1) determinant evolution; (2) solution of simultaneous equations; and (3) finding the matrix inverse. Two more packages have been constructed--a general linear-equation solver, and a matrix-function calculator.

The procedures used are direct in that they involve no iterations. They are relatively easy to explain, easy to troubleshoot, and essentially machine-independent.

Plans

This project has been terminated.

Project Documentation

1. Staudhammer, J. Matrix manipulation by computer: Direct numerical procedures. SDC document TM-2891/000/00. April 6, 1966. 87 pp.
2. Staudhammer, J. Matrix manipulation by computer: The general solution of linear algebraic equations. SDC document TM-2891/001/00. April 20, 1966. 31 pp.
3. Staudhammer, J. Matrix manipulations by computer computation of functions of a matrix. SDC document TM-2891/002/00. May 27, 1966. 38 pp.

COMPLETED STUDIES

The following studies conducted by the Education and Training staff were completed prior to 1966 and are not described in this report.

Variables in a Manually Simulated Teaching Machine

1. Coulson, J. E. and Silberman, H. F. Effects of three variables in a teaching machine. Journal of Educational Psychology, 1960, 51, 135-143.

Negative Reinforcement in Automated Teaching

1. Melaragno, R. J. Effect of negative reinforcement in an automated teaching setting. Psychological Reports, 1960, 7, 381-384.

Confirmation and Prompting Techniques in Instruction

1. Silberman, H. F., Melaragno, R. J., and Coulson, J. E. Confirmation and prompting with connected discourse material. Psychological Reports, 1961, 9, 235-238.

Comparison of Branching and Fixed-Sequence Programs

1. Coulson, J. E., Estavan, D. P., Melaragno, R. J., and Silberman, H. F. Effects of branching in a computer controlled auto-instructional device. Journal of Applied Psychology, 1962, 46, 389-392.

2. Silberman, H. F., Melaragno, R. J., Coulson, J. E., and Estavan, D. P. Fixed sequence versus branching autoinstructional methods. Journal of Educational Psychology, 1961, 52, 166-172.

Development and Evaluation of Self-Instructional Materials for Underachieving and Overachieving Students

1. Silberman, H. F., Coulson, J. E., Gunn, E. M., and Melaragno, R. J. Development and evaluation of self-instructional materials for underachieving and overachieving students. SDC document TM-727. July 1962. 115 pp.

Tutorial Techniques for Program Development and Improvement

1. Coulson, J. E. Research with a program on geometric inequalities. SDC document TM-895/102/00. June 26, 1964. 84 pp.

2. Melaragno, R. J. Research with an elementary arithmetic program. SDC document TM-895/101/00. June 4, 1964. 28 pp.

3. Newmark, G. Research in programmed instruction in Spanish with seventh-grade students. SDC document TM-895/103/00. June 29, 1964. 102 pp.

4. Silberman, H. F. Exploratory research on a beginning reading program. SDC document TM-895/100/00. June 15, 1964. 190 pp.

5. Silberman, H., Coulson, J., Melaragno, R., and Newmark, G. Use of exploratory research and individual tutoring techniques for the development of programming methods and theory. SDC document TM-895/200/00. June 29, 1964. 58 pp.

Nonprogram Variables in the Application of Programmed Instruction

1. Coulson, J. E., Melaragno, R. J., and Silberman, H. F. Final report: Nonprogram variables in the application of programmed instruction. SDC document TM-2176/201/00. July 21, 1965. 36 pp.

2. Coulson, J. E. and Silberman, H. F. A tutorial approach to program design and improvement. Programed Instruction, 1965, 4(8), 7-8.

3. Marsh, D. G. JOVIAL in CLASS. Annual Review in Automatic Programming, 1964, 4, 167-181.

An Optimal Method of Presenting Paired-Associate Items

1. Dear, R. E., Silberman, H. F., Estavan, D. P., and Atkinson, R. C. Final report: An optimal strategy for the presentation of paired-associate items. SDC document TM-1935/101/00. March 10, 1965. 56 pp.

MATHEMATICS AND OPERATIONS RESEARCH

C. J. Ancker, Jr., Head

The principal projects in this area are listed below:

1. Mathematical Models of Stochastic System Elements
2. Statistical Methods in Operations Research
3. Mathematical Programming
4. Numerical Analysis
5. Vehicular Traffic Study
6. Statistical Decision Procedures for Mathematical Learning Models
7. Computer-Assisted Teaching of Mathematics
8. Celestial Mechanics and Differential Equations
9. Data Compression Techniques

These projects fall into two general categories. Those in the first category are intended to contribute to the development of those tools of applied mathematics that are particularly relevant to operations research and systems analysis (among SDC's principal areas of competence). Although all of applied mathematics is, in principle, potentially useful for such practical purposes, a handful of especially relevant techniques have been pushed to the forefront in recent years. We have been particularly concerned with some of the more important of these techniques: probability theory (project 1), mathematical statistics (project 2), optimization techniques (project 3), and numerical analysis with special emphasis on digital computer applications (project 4).

Projects in the second category are designed to provide support for activities centered in another part of the Corporation or to initiate corporate capability in some new field. Project 5 is the first corporate endeavor in transportation science; projects 6 and 7 provide direct support to SDC's computer-assisted education research and development activities; projects 8 and 9 are closely related to corporate space projects.

The two groupings are interrelated. Some of the work in the second category results in the development of new analytical techniques; for example, 5 has motivated a study of parameter estimation by simulation and 6 has generated an optimization technique that is not confined solely to learning theory. Conversely, most of the work on techniques (projects 1-4) is coupled to a particular field of SDC interest. For example, in 1, the waiting-line models are all solutions to abstractions of previously unsolved problems arising in other SDC applications and the traffic models are directly related to 5; 2 contributes new results in simulation and modeling; in 3, one substudy has investigated a computer core allocation problem; and 4 is particularly concerned with the computational aspects of certain space trajectory problems.

During the past year a contract was obtained with the Bureau of Public Roads for partial funding of the traffic study. The simulation model of a diamond interchange is well under way; an initial version of the model's first component was coded, debugged and exercised. Preliminary validation was completed. A second version (final and more extensive) is being coded. Validation and input data are to be obtained by both aerial photography and surface photoelectric techniques. The equipment (aerial reconnaissance camera, film reader, photocells, etc.) has been obtained and checked out.

Initial data at an oversaturated intersection have been obtained photoelectrically and will be used for input to the simulation and for construction of a theoretical submodel. A first theoretical model of this situation was finished.

In space-related work, a new project was activated and initial studies begun on the problem posed by the enormous amount of data transmitted from spaceborne hardware. In celestial mechanics, the work on periodic and quasiperiodic orbits continued and new procedures for existence proofs were derived and applied. Also, a very efficient numerical technique was derived, and partially checked out, for solving certain differential equations applicable to orbit trajectories.

In the education area, the work on optimal strategies for item presentation in stimulus sampling models was brought to a conclusion with a proof of optimality for certain rules in certain cases. The project on computer-assisted teaching of mathematics was also terminated with the final development of programs to teach matrix algebra, linear equation solving, and matrix function calculation.

The theory of stochastic duels was extended to the case where time-of-flight plays an important role. An exhaustive survey of the field was also completed.

In the field of statistics as applied to operations research, new results were obtained in life-testing and validation of simulation models. The factor analysis project was terminated with the resolution of the generalized Heywood case by a process of minimizing the sum of squares of off-diagonal residuals.

The work in mathematical programming saw the successful completion of the computer-core allocation algorithm. The dynamic programming and control theory project was also concluded with the derivation of a good approximate algorithm for very large "traveling salesman" problems. Finally, some results in scheduling under resource constraints (where partial ordering and resource profiles are specified) have been obtained.

Note: Two studies conducted by members of, or consultants to, the Mathematics and Operations Research staff are described in other sections of this report. These include:

R. A. Totschek - Analyses of Time-Sharing, see under Programming Systems, page 3-9.

J. Staudhammer - Computer-Assisted Teaching of Mathematics, see under Education and Training, page 7-20.

MATHEMATICAL MODELS OF STOCHASTIC SYSTEM ELEMENTS

Mathematical Models of Vehicular Traffic*

C. J. Ancker, Jr.
A. V. Gafarian

Description

Current analytical models of vehicular traffic phenomena may be categorized as either deterministic or stochastic (containing random elements). Although deterministic models (which predominate in the literature) have some limited usefulness, the outstanding characteristic of traffic flow is its randomness. This project concentrates on developing stochastic models in the belief that ultimately these will prove to be more useful.

This project augments the computer simulation of vehicular traffic reported on page 8-12.

Progress

Some prior work on throughput at an oversaturated signalized intersection has been refined, simplified, and recast in the form of a renewal theory model [1]. The model is based on the following assumptions:

1. An infinite queue of vehicles is stacked at a signalized intersection.
2. The go-phase of the signal has a fixed time-length T.
3. The interval of time between the initiation of the go-phase and the entry of the first car into the intersection is a positive random variable X.
4. All subsequent entry-time headways are assumed equal to a fixed minimal time τ plus the random variable X.
5. All random variables are assumed to be independent.
6. If a vehicle reaches the intersection before the end of a go-phase, it is committed to a passage through.

A renewal theory analysis of the distribution of the number of vehicles passing through the intersection per go-phase is given. The special case of a general Erlang distribution is thoroughly analyzed.

Plans

Some data on oversaturated intersections, collected in the Vehicular Traffic Study (see p. 8-12), are being analyzed in the light of the assumptions above. In particular, the assumptions of independence and identical distributions from position to position are being scrutinized. From preliminary analysis, it seems clear that a more comprehensive theory must be developed. Models of other common congestion situations are also contemplated.

Project Documentation

1. Ancker, Jr., C. J. and Gafarian, A. V. A simple renewal model of throughput at an oversaturated signalized intersection. SDC document SP-2122/000/01. October 28, 1966. 17 pp.

Models of Combat

C. J. Ancker, Jr.

Description

Military command, control, and logistic systems are primarily concerned with combat operations or combat-support functions. The expected outcome of tactical and strategic decisions is vitally important in the operation of these systems. Clearly, a theory of combat (or the elements of a theory) could contribute to improved system operation and performance. In the past, analytical theories of combat generally have used differential equations (Lanchester's), game theory, and optimization theory. All of these tend to consider the problem in global terms and usually (but not always) bypass the probabilistic nature of combat. The approach taken here begins with local fundamentals and emphasizes the stochastic nature of the phenomena. The duel of one versus one is analyzed extensively, with the principal ingredients of the fire fight being accuracy of fire, rate of fire, distribution of fire, cover, concealment, mobility, and surprise. The extension of

*Supported in part by the Bureau of Public Roads, U. S. Department of Commerce.

the duel to combat involving more than two contestants is also considered.

Progress

The fundamental duel in the Theory of Stochastic Duels was extended to include projectile time-of-flight [3]. Two firing procedures were assumed. In the first, refiring proceeds as rapidly as possible; in the second, the duelist delays (observing the effect of each round) before he fires the succeeding round. Both fixed (discrete) and continuous (random) firing-times were considered. Also included was the case where time-of-flight varies uniformly with elapsed time. General solutions and examples were given.

A complete survey of the literature in the field (to include work being done elsewhere) was completed [4].

Plans

Other extensions are contemplated. Among these are random bursts of several rounds at a time as well as combined time and ammunition limitations.

Project Documentation

1. Ancker, Jr., C. J. and Gafarian, A. V. The distribution of the time-duration of stochastic duels. Naval Research Logistics Quarterly, 12(3&4), September-December 1965, 275-294. (Also available as SDC document SP-1017/007/00.)

2. Ancker, Jr., C. J. Stochastic duels of limited time-duration. Canadian Operational Research Society (CORS) Journal, 4(2), July 1966, 69-81. (Also available as SDC document SP-1017/005/00.)

3. Ancker, Jr., C. J. Stochastic duels with time-of-flight included. Opsearch (Journal of the Operational Research Society of India), 3(2), 1966, 71-92. (Also available as SDC document SP-1017/009/00.)

4. Ancker, Jr., C. J. The status of developments in the theory of stochastic duels--II. SDC document SP-1017/008/01, September 28, 1966. 32 pp.

STATISTICAL METHODS IN OPERATIONS RESEARCH

Generally Applicable Statistical Techniques

J. E. Walsh

Description

This project seeks to develop statistical results of practical utility for use in reliability testing, experimentation, validation of simulation models, and other areas. These techniques are intended to be valid under such general circumstances that there will seldom be doubt about their usability. They are intended for application to commonly encountered practical situations.

Progress

Results were developed in two areas: (1) life-testing, and (2) validation of simulation models.

In life-testing, the items of a group are simultaneously considered; their times to failure are the observations. These observations are assumed to have lognormal probability distributions. With the use of very few observations, and for specified high probabilities, lower bounds were determined for the length of time that a typical item would survive [1].

Validation of simulation models is obtained by comparison of results from operational systems with corresponding results obtained from simulation. The considerations and concepts involved, as well as potentially useful statistical methods, are outlined in reference [2] along with some illustrations for a situation involving traffic flow.

Plans

The material on validation of simulation models will probably be extended. Development of generally applicable regression functions for estimating outcomes from simulation results, and for biomedical uses, will be considered. Generalizations of paired comparison methods may be developed in conjunction with the regression results.

Project Documentation

1. Walsh, J. E. and McCulloch, A. J. Life-testing results based on a few heterogeneous lognormal observations. SDC document SP-2038/000/01. June 8, 1966. 4 pp.
2. Walsh, J. E. and Gafarian, A. V. Statistical approach for validating simulation models by comparison with operational systems--Illustrated for traffic flow. SDC document SP-2367. February 24, 1966. 17 pp.

Factor Analysis Methodology

H. H. Harman
Y. Fukuda

Description

Factor analysis is a multivariate statistical technique especially useful in the social sciences, where numerous interacting measurements are frequently obtained. The observed variables are presumed to be expressible in terms of some new hypothetical constructs, or factors, with residual elements. Usually the model is linear, with the number of common factors (m) considerably smaller than the number of observed variables (n), e.g.,

$$z_j = \sum_{k=1}^{m} a_{jk}F_k + d_jU_j \qquad (j=1,2,\ldots,n)$$

with a unique factor provided to take up the unexplained variance. One of the principal objectives of the analysis is to estimate the a's of this model.

In the study of statistical relationships, some techniques are concerned with analysis of dependence (e.g., regression analysis), while others deal with analysis of interdependence. Factor analysis falls in the latter class, addressing itself to the study of interrelationships among a total set of variables, no one of which is selected for a different role than any of the others. Since it is concerned primarily with the resolution of a set of variables in terms of factors (usually small in number), its application has been directed largely toward the attainment of scientific parsimony or economy of description.

Progress

In the course of developing the minres method of factor analysis [5], the troublesome situation arose of communalities greater than one. This problem--referred to as the generalized Heywood case--has been resolved by means of a process of minimizing the sum of squares of off-diagonal residuals [1]. The resulting solution is superior to the otherwise very efficient original minres method and does not require additional computing time. The computer program is written in FORTRAN II for the Philco 2000 and is available at SDC.

Plans

This project has terminated.

Project Documentation

1. Harman, H. H. and Fukuda, Y. Resolution of the Heywood case in the minres solution. SDC document SP-2334. January 20, 1966. 17 pp.
2. Harman, H. H. Direct oblimin program for the Philco 2000. SDC document TM-2956. May 3, 1966. 6 pp.
3. Harman, H. H. Program for the minres method of factor analysis. SDC document TM-2958. June 9, 1966. 8 pp.
4. Harman, H. H. Program for the maximum-likelihood method of factor analysis. SDC document TM-2960. June 15, 1966. 8 pp.
5. Harman, H. H. and Jones, W. H. Factor analysis by minimizing residuals. Psychometrika, 31(3), September 1966, 351-368. (Also available as SDC document SP-2134.)

MATHEMATICAL PROGRAMMING

Strategies of Item Presentation in Learning Processes

R. E. Dear
W. Karush

Description

We deal with the problem of choosing a strategy of item presentation to teach a subject n items in the course of N trials, N typically much larger than n. The items may be chosen in arbitrary sequence with any number of repetitions. A prototype is the learning of paired-associates. The effectiveness of a strategy is measured by the expected value of a subject's terminal score in which each item has equal weight. A certain model of learning (stemming from the stimulus-sampling theory of learning) is assumed, and investigation is directed to the problem of determining an optimal decision rule for trial-by-trial item selection under this hypothesis.

The model postulates that the subject occupies at any stage one of m possible states of knowledge, or conditioning, with respect to each item. These states are not observable, but what is observable is the response (correct or incorrect) to a test or presentation of an item. Each state corresponds to a certain probability of a correct response: the higher the state, the higher this probability. A trial consists of a test on a selected item followed by a reinforcement action on that item and a resulting transition to the next higher state with a given probability. This transition probability is called the learning rate. The model assumes that the learning of an item proceeds independently of the learning of other items.

Mathematically, the subject is characterized at any stage by a state probability vector $\lambda_i = (\lambda_i(1), \lambda_i(2), \ldots, \lambda_i(m))$ for each item i; here, $\lambda_i(j)$ is the probability of state j for item i. This vector for a given item undergoes a known transformation between the beginning and end of a trial, depending upon the response of the subject. By means of dynamic programming, an optimal strategy can be expressed as the solution of a recursive functional equation involving branching. The parameters of the problem are the initial state probability vectors λ_i^0, the learning rate θ and the probabilities $\gamma(j)$, $j=1,2\ldots,m$, of a correct response in state j.

Progress

Consider the following rule (which depends upon a fixed positive integer k) for selecting an item in a given trial: Proceed as if k trials remain in the process (including the current trial) and determine an optimal strategy for the next k trials; then select the first item that occurs in this "locally optimal" strategy for the current trial. If this procedure is followed at each trial, we call it the k-trial look-ahead decision rule $R^{(k)}$; when k trials actually remain to the end, the full locally optimal strategy is used to complete the total strategy. Most of our work involved the investigation of the proposition that for an m-state model of learning, the rule $R^{(m-1)}$ is an optimal one.

Our theoretical results have been obtained mainly for the important case of the two-state theory, m=2. There, we were successful in proving mathematically that $R^{(1)}$ was optimal for the model described above and for several modifications of the model. One modification allowed the learning rate following an incorrect response to be different from the rate following a correct response; this variation envisages the realistic possibility that the corrective action depends upon the response. We also derived results concerning the dependence of an optimal strategy on the parameters of the two-state model; in particular, we proved that the optimal strategy is independent of the response probabilities and of the learning rates, under the assumption that (1) the subject starts in the lower state of conditioning for each item and (2) the probability of a correct response is 1 in the higher state.

For the case of m=3 or higher, the problem of an optimal strategy has defied theoretical resolution. We have carried out numerical experiments on a computer to study the look-ahead rules $R^{(k)}$, k=1,2,...,N, and to test the conjecture that $R^{(m-1)}$ is optimal. A computer program in a time-sharing system was developed and used for m=3 and for processes of up to 6 trials; this program obtained optimal strategies by enumeration of all strategies and also yielded strategies generated by the various look-ahead rules. Our computations showed that $R^{(m-1)}$

always generated at least one optimal strategy; however, the whole set of strategies generated by $R^{(m-1)}$ could contain nonoptimal strategies or fail to contain some optimal strategies. Further analysis of the data led to the introduction of a notion of equivalence of strategies; this concept allowed certain groups of distinct strategies to be identified as single strategies. Using such an identification, we found that the data supported the stronger conclusion, i.e., that all optimal strategies were included in the set of strategies generated by $R^{(m-1)}$.

Plans

This project has been terminated.

Project Documentation

1. Karush, W. and Dear, R. E. Computational studies of presentation strategies for a multilevel model of learning. SDC document TM-3035. July 8, 1966. 29 pp.

2. Karush, W. and Dear, R. E. Optimal stimulus presentation strategy for a stimulus sampling model of learning. Journal of Mathematical Psychology, 1966, 3, 19-47. (Also available as SDC documents SP-1869 and SP-1898.)

3. Karush, W. and Dear, R. E. An optimal procedure for an N-stage learning process. SIAM Journal, Series A: Control, 1966, 4(1), 116-129. (Also available as SDC document SP-1922.)

Scheduling Under Resource Restraints

W. Karush

Description

The type of problem addressed here is that of optimization in a discrete space; the approach taken involves techniques of combinatorial analysis. These problems present serious mathematical difficulties: No computationally effective general algorithm is at hand and each mathematical model requires the invention of a unique procedure for efficiently searching out optimal solutions.

The question to which we have given the most attention is that of scheduling a project represented as a network of activities that share a common resource over time. This scheduling problem arises in various applications of operations research; notable examples are project management systems such as PERT and the Critical Path Method. Our model takes a project to be a set of activities with each activity having a known fixed duration time. We assume a partial order relation, or technological ordering, among the activities; the order relation specifies that certain activities must be completed before others can begin. With each activity is associated a known profile of resource requirements defined over its duration interval and allowed to vary over its interval. Also given is an overall profile of resources available to the project as a whole. A feasible schedule is an assignment of start times for all activities that respects technological ordering and meets resource restraints in every time period; an optimal schedule is a feasible schedule that minimizes the lifetime of the project. The model is formulated in discrete terms.

Progress

Results have been obtained on two problems. The first deals with reducing the cardinality of of the class of feasible schedules that must be searched to find an optimal one; the second deals with the assignment of individual workers to activities in such a way as to achieve maximal continuity of assignment.

In the first problem, we introduce the subclass of feasible schedules generated by left-packing of activities in any linear sequence that is consistent with the given partial ordering of the activities. We prove that when the resource profiles of all activities are nonincreasing, then the subclass contains an optimal schedule. A counterexample shows that the result fails if the nonincreasing condition is dropped. The result forms the basis for a branch-and-bound scheme of searching for an optimal schedule based on enumerating linear arrangements of activities. Our results apply to projects with nondecreasing profiles by using right-packing; they also apply to the case of several resources.

In the second problem, we show that maximal continuity of worker assignment can be achieved when the profile of every activity is single-peaked (this includes increasing, decreasing and constant profiles). When no assumption on profiles is made, maximal continuity is in general not possible, as we show by a counterexample.

Plans

It would be desirable to extend our results on reduction of the cardinality of the searched space to profiles of a less restrictive character than nonincreasing ones; one important extension would be to include single-peaked profiles. A desirable goal is to discover a subclass of schedules that could become the basis of a branch-and-bound computational searching procedure. Other extensions or modifications of the model can be considered for important operations research applications dealing with the allocation and scheduling of resources such as money, manpower, facilities, and services.

Project Documentation

1. Karush, W. A counterexample to a proposed algorithm for optimal sequencing of jobs. Operations Research. 1965, 13(2), 323-325.
2. Karush, W. On scheduling a network of activities under resource restraints over time. SDC document SP-2654. November 15, 1966. 30 pp.

Static Allocation of Computer Core-Storage Space

Y. Fukuda

Description

A general class of static computer storage allocation problems may be defined in the following way: There is given a collection of memory-consuming entities (routines, tables, data, etc.) whose space requirements are known. These entities make up one or more programs. During the course of using the programs, these entities are read in and out of the storage in a known sequence. It is desirable to allocate storage space to the entities in such a way that the maximum memory space required during the operation of the programs is minimized.

Specifically, the problem considered here is a combinatorial one. Given n entities with space requirement v_j and column requirement c_j, a primary objective is to arrange these n entities in a two-dimensional space (horizontal axis corresponding to columns, vertical axis to storage space) in such a way that the maximum requirement for storage space among all the columns is minimized. A column here is synonymous with a period of time, and a column requirement is a specification of columns (or periods) in which a given entity is required for computation. An entity may not be moved horizontally due to the column requirement, but it may be moved vertically. The minimum space requirement for a given column is a sum of v_j for all entities occupying the given column. A blank space is often found in a given column as a result of the interactions among the space and column requirements of the neighboring entities. If the longest column in a given arrangement has no blank space, that arrangement is a solution; such a solution may be called a packed solution.

Progress

An algorithm has been derived for obtaining an optimal allocation scheme. This algorithm closely resembles, in spirit, a branch-and-bound algorithm for the "traveling salesman" problem.

Plans

This project has been terminated.

Project Documentation

1. Fukuda, Y. A problem in static storage allocation. SDC document SP-2435. April 20, 1966. 25 pp.

Dynamic Programming and Control Theory

M. Ash

Consultant: J. Staudhammer, Arizona State University

Description

A classic unsolved problem is that of the traveling salesman. The problem is to construct an efficient algorithm, from the computer programming standpoint, which yields the sequence in which a number of cities should be visited so that the tour distance is minimized.

An immediate incentive for determining feasible solutions is an application to the stellar sequencing class of problems as faced on the Orbiting Astronomical Observatory (OAO) project. Efficient computer algorithms are required for determining the order in which up to 5000 celestial objects (mainly stars) should be tracked by the OAO satellite, during its useful life of one year, in order to minimize a prescribed cost functional. The two main characteristics of the OAO class of traveling salesman problems are the very large number of objects, and the system and environmental constraints.

Progress

A method is described for obtaining a solution to very large symmetric traveling salesman problems and establishing an absolute lower bound for the optimal tour [1]. The method presented is a variation of the "nearest city approach." Segments of the initial path are chosen from a list of all possible links arranged by increasing magnitudes. For large problems, only about 1 percent of the paths are examined initially; disjoint strings are found at this stage. The interconnection of these strings gives a path that is usually not very much longer than the optimal tour. The interconnected strings will miss a few cities; these are interpolated into the string after small loops are removed. (The method described here will not detect or rectify loops containing more than four links.) The tour obtained by this algorithm is optimal in many instances, but in any case where the solution is unknown it cannot be stated how closely the derived tour approaches the optimal one.

An estimate of the shortest possible path is obtained by summing the two shortest distances emanating from each node. In almost all test cases this "utopian" path and the minimum tour differed by no more than 15 percent. Since this figure might well fall within the accuracy specified in the problem, some "sufficient solutions" may be obtained by the given method. If further refinement is indicated, any one of a number of published procedures may be employed.

On the other hand, if the initial solution and the "utopian" path differ greatly, the initial solution may still be close to the optimal tour; but there will be no indication of this fact.

Several published large traveling salesman problems were recalculated by this procedure, and the actual optimal paths given in the original articles were obtained. Further application was made to a set of meter-reading problems, and the optimal solution was found in the majority of cases.

Plans

This project has been terminated.

Project Documentation

1. Staudhammer, J. and Ash, M. A sufficiency solution of the traveling salesman problem. SDC document SP-2514. August 26, 1966. 64 pp.

NUMERICAL ANALYSIS

J. Dyer

Consultant: T. S. Motzkin, University of California, Los Angeles

Description

The purpose of this project is to apply recent results in the theory of predictor-corrector methods of numerical integration to the problem of satellite trajectory computation. A traditional k-step stable difference equation (method) for the numerical solution of the differential equation $y'=f(x,y)$ is severely limited in the order to which it approximates the differential equation. Any traditional difference equation whose order of accuracy is greater than k+2 is necessarily unstable. Recently it has been shown that a slightly generalized method can produce a much higher order of accuracy without the loss of stability, at least for $k < 8$.

Multistep methods for the special second-order equation $y''=f(x,y)$ also suffer this limitation. It is interesting to ask whether stable, similarly generalized methods also exist here with high orders of approximation. If they exist, it is of interest to study analytically and experimentally the behavior of accumulated truncation error and round-off error.

Methods for solving the second-order equation are important in themselves. Moreover, the Gauss-Jackson algorithm for satellite orbit trajectory computation is based in part on a method of this type. A possible goal, then, is the improvement of this algorithm.

The Gauss-Jackson algorithm employs "summed" forms of simple difference equations. The advantage of such forms in the control of round-off error has been theoretically established, but in practice it is not entirely clear. The problem needs further study. Truncation errors in the summed form and orthodox form are the same.

Most familiar multistep methods are based on polynomial approximation. In the constructive part of the work discussed here, a Hermite-like interpolation of some intrinsic interest has been used. Related theory is being developed.

Progress

A first phase of the work has been completed. A method of polynomial approximation has been devised for the solution to $y''=f(x,y)$. For $k < 8$, slightly generalized stable explicit and implicit k-step difference equations have been constructed with orders of accuracy higher than possible with traditional methods.

Some of these new methods have been tested by computing an unperturbed satellite orbit trajectory. To give a rough indication of their usefulness, the graph in Figure 8-1 indicates the number of decimal digits of accuracy as a function of interval of integration, h, for the computation of a standard trajectory by three different methods. Computation was done in double-precision FORTRAN on the IBM 7094 Computer.

Concerning the orbit we give only that

(eccentricity) $e = .2$

orbit period $\sim$ 155 minutes

perigee distance $\sim$ 800 miles

apogee distance $\sim$ 3200 miles

and omit initial values for the solutions computed. Methods A and E both require starting values of five equally spaced time points. Both methods consist of one "prediction" and one "correction." Thus they require about the same length of machine time for one iteration. Method A is the unsummed version of the Gauss-Jackson method. Method E is a generalized method. One can perhaps conclude that to compute the trajectory for three revolutions and retain nine decimal digits of accuracy in the solution, the Gauss-Jackson method requires roughly three times as much computing effort as method E. Accumulated round-off error has not yet begun to make itself significantly felt. Method D has only a three-step corrector but requires two predictors.

Some problems pertaining to the polynomial interpolation have been examined and results given [1]. The constructive methods used also permitted definition of new stable explicit operators for solving a first-order differential equation.

Plans

Future work will proceed along the following lines:

1. aer analytical and experimental work will be done in the study of the behavior of accumulated truncation error and round-off error for new methods.

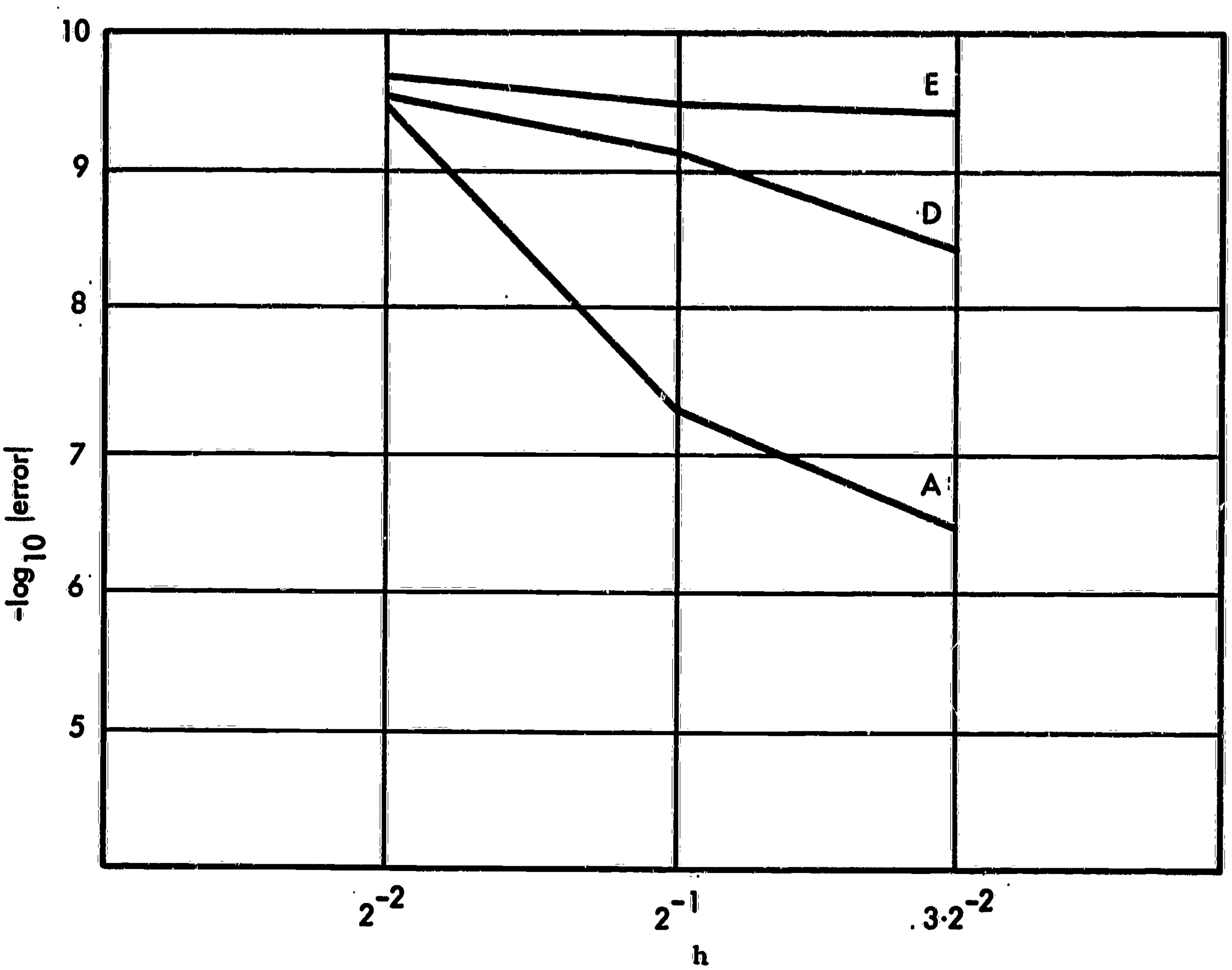


FIGURE 8-1. PRECISION IN COMPARED METHODS, t = 498 min.

2. Related polynomial interpolation theory will be developed more comprehensively.

3. An important question in trajectory work is the advantage of the use of summed difference equations. This is a complex problem depending in part on the precision used in computation. Other organizations are occupied with this problem. If their work is not sufficiently general, a study should be done at SDC. The outcome is important for this project.

4. If results continue to be favorable, new methods will be incorporated into algorithms, analogous to the Gauss-Jackson algorithm, for solving the general equations of motion of an earth satellite.

Project Documentation

1. Dyer, J. Generalized multistep methods and applications to satellite orbit trajectory computation. SDC document SP-2631. October 28, 1966. 63 pp.

VEHICULAR TRAFFIC STUDY*

A. V. Gafarian, Principal Investigator
C. J. Ancker, Jr.
R. K. Gray
E. Hayes

Consultant: W. W. Mosher, Jr., Institute of Traffic and Transportation Engineering, University of California, Los Angeles

Student Associate: R. Smith, Rutgers University

Description

The principal aim of this project is to develop and validate a digital computer simulation model of a diamond interchange with sufficient flexibility to perform research on (1) the control of such interchanges, and (2) the influence of certain aspects of geometric design on its performance. This study is one of a group of related studies being sponsored by the Improved Utilization of High Speed Highways Task Group, Traffic Systems Division, Office of Research and Development, Bureau of Public Roads. Ancillary aims of this project include (1) the collection of vehicular performance data usable as inputs to the design of the diamond interchange model (and as fundamental data applicable to other studies), and (2) the development and subsequent analysis of mathematical models of certain aspects of traffic flow present in the diamond interchange. (See Mathematical Models of Vehicular Traffic, p. 8-3.)

The interchange is complex and will be partitioned into three simpler submodels. Model 1 will deal with the merging of the freeway with an on-ramp. Model 2 will consist of a combination of Model 1 and its upstream off-ramp. Model 3 will simulate the signalized intersections of ramps and the surface arterial. Model 4 is a proper composition of the prior three models and constitutes the whole interchange. Each of the first three models will be designed, programmed, debugged, and validated (with subsequent modification of the design, if necessary) before extensive work is accomplished on the next model. It is expected that this procedure will produce a valid total model much sooner and more efficiently than would an attempt to design the total simulation at the outset.

Progress

1. Simulation

Model Programming

The first version of Model 1 (the merging of an on-ramp with the freeway), as well as its data reduction program, has been coded for the Philco 2000, debugged, and extensively exercised. The results are reported in [1] and [2] and pertain to both the operating characteristics

*Supported in part by the Bureau of Public Roads, U. S. Department of Commerce. The cooperation of both the California State Division of Highways and the Los Angeles City Traffic Department is acknowledged; the State for permitting the use of the freeways and supplying crews for field work and the City for supplying helicopter and pilot for aerial data collection.

and the validity of the model. Several forms of computer output are available under program control. Two of these are shown in Figures 8-2 and 8-3. Figure 8-2 is always produced at the end of each run in addition to any other kind of output desired. It indicates some of the variables and parameters considered in Model 1, including vehicle generating rates, on-ramp signal phase times, vehicle-driver type distributions, maximum free-flow velocities, acceleration factors, etc. Figure 8-3 shows selected information printed at the end of specified periods (30 seconds in this case). Columns 2 and 4 show the number of freeway cars (referred to as OD1) removed from the system and their total average travel time, while columns 3 and 5 show the number of on-ramp cars (referred to as OD5,12) removed from the system and their total average travel time. The sets of three columns under the headings V3, V26, V25, V24, V23 (the various submodels comprising Model 1) show respectively the occupancy at the end of the time period--i.e., the percent of maximum number of cars the submodel can accommodate--the number of cars removed during the particular time interval, and their average travel time through the submodel.

A unique feature of our model is that it is macroscopic. The usual approach is to simulate microscopically and thus determine, as a function of time, the location of each vehicle in the system. The macroscopic approach was taken because:

It has a higher likelihood of resulting in a valid model. A microscopic simulation requires knowledge of the laws of car-following, gap-acceptance, weaving, etc. The lack of accurate knowledge in these areas, and the difficulty of acquiring it, decreases the likelihood of a valid simulation. On the other hand, a grosser simulation, though it obviously cannot answer certain kinds of questions, employs aggregated performance characteristics which are easier to measure or to estimate correctly.

. The simulated-time to computer-time ratio is more favorable for the macroscopic approach. This factor is exceedingly important: Reliable estimates of the measures of effectiveness require either many replications (or very long single runs) because of the stochastic nature of highway traffic. The ratio for microscopic simulation is poor, since every vehicle must be checked and moved, if necessary, at relatively small intervals of simulated time. Time increments as small as .1 second have been used for models of this type.

The principal reason the macroscopic approach has not been used before on simulations of this kind is that the program logic and coding are considerably more difficult than in microscopic simulation.

Results from Version 1 of Model 1, including preliminary validation results, indicate that the present macroscopic modeling method has a high likelihood of resulting in a total model with realistic performance characteristics. Accordingly, the present version of Model 1 is being extended, by use of the same macroscopic approach, to a second version that will simulate more realistically the formation and dissolution of congestion. This refinement is extremely important, since congestion situations are the critical ones.

Version 1 was programmed in machine code for the Philco 2000. This version was recoded in JOVIAL to assess the effects of higher-order languages (FORTRAN, PL/I, JOVIAL, etc.) on computer running time. Comparison of results of the machine-coded and JOVIAL-coded programs showed the following:

. Production runs with the JOVIAL program took from 2½ times (best case) to 5 times (worst case) longer, with an average of between 3 and 4 times.

. The JOVIAL program required 50 percent more instructions.

VEHICULAR TRAFFIC STUDY PARAMETERS

FREEWAY - DIAMOND INTERCHANGE..MODEL 1

10-01-65 -- VERSION 01 -- RUN 02

VEHICLE GENERATING	FREEWAY		POCKET LANE	CURB LANE	
RATES PER HOUR	= 3800		(GREEN) = 600	(GREEN) = 540	(RED) = 60
V23 ENTRY PROBABILITIES			= .778	= .872	= .745
VSTBL LENGTHS (FT)	V3 = 2210	V23,24 = 506	V25 = 590	V26 = 360	V27 = 175

SIGNAL PHASE TIMES (SEC)

1	= 12
2	= 09
3	= 18
4	= 09
5	= 06
6	= 06

DISTRIBUTION OF DRIVER-TYPES (%)		FWY FFVF (FT/SEC)	FWY A-FCT (FT/SEC/SEC)	ON-RAMP VO (FT/SEC)	ON-RAMP A-FCT (FT/SEC/SEC)
SLOW	= 07	75	3.0	6	3.0
MEDIUM	= 24	85	4.5	11	3.5
NORMAL	= 38	95	6.0	16	4.0
FAST	= 24	104	7.5	21	4.5
VERY FAST	= 07	114	9.0	26	5.0

FREEFLOW SYSTEM TRAVEL TIMES (SEC)

DRIVE TYPE	O.D. 1	O.D. 5.12
0	36.6	32.2
1	32.3	28.5
2	28.9	25.4
3	26.5	23.2
4	24.1	21.1

AVG VEH LENGTH + GAP (FT)	HEADWAY (SEC)	RUN TIME (MIN)	TINC TIME (SEC)
22	2.0	25	1.5

% OCCUPANCY =	5	10	15	20	25	50	100
PROBABILITY =	.700	.400	.100	.080	.050	.010	.000

TABLES USED TO DETERMINE VELOCITIES DURING "CONGESTION"

V3 to V26		V23,24 TO V25		V25 TO V26		V26 TO V27		QUEUES TO V3		& TO V23, V24	
OCPY (CT)	VELOC (FT/SEC)	OCPY (CT)	VELOC (FT/SEC)	OCPY (CT)	VELOC (FT/SEC)	OCPY (CT)	VELOC (FT/SEC)	OCPY (CT)	VELOC (FT/SEC)	OCPY (CT)	VELOC (FT/SEC)
50	75	13	63	50	72	12	75	370	75	10	24
55	64	15	53	55	61		64	375	64	12	20
60	53	17	43	60	50	22	53	380	53	14	17
65	42	19	33	65	39	27	42	385	42	16	13
69	31	21	23	69	28	32	31	390	31	18	10
73	21	23	13	73	17	37	20	395	21	20	6
77	10	25	3	77	6	42	9	400	10	22	3

FIGURE 8-2. SAMPLE COMPUTER OUTPUT OF A PROLOGUE PAGE

SYSTEM SNAPSHOT --- MOST CUMULATIVE VALUES ARE RESET EVERY 30 SECONDS

TIME	VEH REMOVED		AVG SYSTRT		V3			V26			V25			V24			V23		
(SEC)	OD1	OD5.12	OD1	OD5.12	OCPY	REM	AVTRT	OCPY	REM	AVTRT	OCPY	REM	AVTRT	OCPY	REM	AVTRT	OCPY	REM	AVTRT
30	3	2	26.5	25.4	7.9	8	22.3	4.9	6	3.6	3.8	2	7.3	8.6	0	.0	13.0	3	13.7
60	39	8	31.2	28.6	9.7	39	25.7	8.6	45	4.1	15.3	9	9.5	.0	3	13.7	8.6	9	14.2
90	82	16	33.1	30.1	8.2	45	27.2	12.3	50	5.1	19.2	8	9.8	4.3	2	14.5	26.0	7	14.7
120	114	29	33.3	31.5	6.7	38	26.8	18.5	47	5.2	15.3	14	11.1	4.3	2	15.4	8.6	11	14.6
150	152	40	33.7	32.5	11.2	34	25.6	12.3	’8	5.9	19.2	9	10.6	4.3	2	16.3	17.3	8	14.0
180	202	49	33.7	32.6	10.4	50	25.7	13.5	60	5.5	.0	11	10.9	.0	1	16.3	4.3	5	15.5
210	253	55	34.0	32.7	8.9	46	27.2	9.8	53	5.2	.0	4	7.3	4.3	0	.0	17.3	4	12.5
240	291	62	33.9	32.0	11.7	45	25.7	17.2	48	4.5	11.5	9	7.6	4.3	3	13.0	8.6	9	13.2
270	336	73	34.0	32.0	10.7	51	27.5	17.2	60	5.7	19.2	9	9.6	.0	2	15.4	26.0	9	13.0
300	386	83	34.7	32.3	8.4	47	27.0	16.0	60	9.0	15.3	12	9.6	.0	0	.0	8.6	11	14.8
330	431	94	35.0	32.8	7.4	40	26.5	9.8	54	5.9	15.3	9	9.6	4.3	3	13.7	26.0	6	13.9
360	466	106	34.7	32.7	6.7	34	24.8	13.5	45	4.5	19.2	14	10.7	.0	3	13.6	8.6	12	14.3
390	502	119	34.6	32.5	9.2	30	25.8	7.4	46	4.8	11.5	11	9.5	8.6	4	13.4	17.3	5	12.5
420	538	128	34.6	32.4	8.7	42	26.3	13.5	48	4.8	7.6	11	10.6	.0	2	16.2	4.3	8	13.6
450	577	136	34.5	32.3	8.2	37	26.7	8.6	46	5.2	19.2	5	8.1	.0	3	14.3	30.4	5	13.0
480	610	147	34.5	32.2	8.9	34	26.4	12.3	45	4.7	19.2	14	10.3	4.3	1	15.3	8.6	13	15.2
510	653	160	34.5	32.4	7.9	41	26.0	7.4	56	6.0	19.2	11	9.6	4.3	1	12.5	21.7	10	13.0
540	691	174	34.4	32.0	9.7	33	24.3	1.2	52	4.7	3.8	14	9.1	.0	1	12.5	4.3	9	12.6
570	730	180	34.3	31.9	8.9	44	26.7	11.1	41	4.5	11.5	5	8.0	.0	2	16.3	34.7	5	13.2
600	773	190	34.2	32.0	8.7	43	25.2	12.3	54	4.9	11.5	12	10.8	.0	1	14.3	4.3	11	15.3
630	813	198	34.1	32.0	7.7	44	25.0	11.1	51	4.7	19.2	6	8.7	.0	3	14.3	30.4	5	13.0
660	850	209	34.1	32.0	8.9	35	26.8	9.8	49	4.7	19.2	13	10.6	.0	0	.0	8.6	13	15.7
690	891	220	34.1	32.1	7.2	40	27.2	7.4	51	5.1	23.0	9	10.6	4.3	2	12.1	21.7	8	14.1
720	926	232	34.0	32.1	5.7	33	24.4	7.4	46	4.6	23.0	13	10.9	8.6	2	13.6	4.3	11	14.7
750	958	244	34.0	32.2	10.4	28	24.5	2.4	44	5.0	15.3	12	10.7	4.3	3	13.0	26.0	7	14.3

FIGURE 8-3. SAMPLE COMPUTER OUTPUT OF A SYSTEM SNAPSHOT PAGE

. The storage space in the machine remained approximately the same.

It is expected that similar ratios will exist when future models are programmed in JS JOVIAL for SDC's S/360 Time-Sharing System. This means that substantial areas of the model will require machine coding if reasonable simulation-time to computer-time ratios are to be obtained in the much more complicated Version 2 which is presently being constructed. Our plan is to code initially in JS and to recode substantial areas in machine language (subsequent to debugging of the JS version) to increase efficiency.

Data Collection for the Model

In constructing the diamond interchange model, many driver-vehicle performance characteristics will need to be determined. One example is the driver's desired velocity on the freeway. This will be determined by measuring velocities on the freeway when traffic is extremely light. Presumably, under these conditions, people are traveling at their desired velocities. These measurements will be made using aerial photography (the procedure is described in the validation section, below).

Another characteristic required as an input to the model is the manner in which waiting vehicles (say at a signalized off-ramp) are discharged during a go-phase of the signal cycle. Field work on this was begun by collecting data on cars in the middle southbound lane of the Pacific Coast Highway where it intersects the California Street incline in Santa Monica. A photoelectric cell, placed two feet upstream of the stop line, was used to measure the successive departure time headways of cars queuing up during the stop phase of the signal cycle. Figure 8-4 shows a queue of cars and the taped wires from the photoelectric cell, buried in the pavement of the middle lane, that led to the instrumentation. Figure 8-5 shows the tape recorder and other electronic equipment located in a position not visible to the drivers.

Statistical analyses are being performed on these data in order to determine: (1) the kind of statistical distribution functions that fit the data with sufficient goodness and (2) whether successive headways are independent. These distributions will then be used in the macroscopic simulation model whenever a queue of vehicles is being discharged.

2. Validation

Data Collection

The evaluation of data collection and reduction techniques was completed. It was decided to use a 70 mm. aerial reconnaissance camera mounted on a helicopter for collecting the data and a semi-automatic film record reader for taking the data off the film. Figure 8-6 shows a 70 mm. frame taken from a 50-foot roll of film exposed at approximately 3400 feet above ground level at the Roscoe Boulevard/San Diego Freeway interchange. The camera, borrowed from Ohio State University, was mounted on a Los Angeles City Fire Department helicopter.

The Benson-Lehner Type 29E Telereadex semi-automatic film record reader has been delivered and is operating. Figure 8-7 shows a photograph of the film reader with its peripheral equipment. It is presently attached to an IBM 519, which is being used only for summary punching. The 519 is being replaced by an IBM 523 summary card punch. A Maurer Model 220 Pulse Sequence Camera, a 38 mm. Zeiss Biogon f/4.5 lens with a set of filters, and a 100-foot wraparound magazine have also been delivered. The intervalometer and control box has been designed and fabricated, and the design for the camera mount has been initiated.

The use of a photoelectric cell for collecting headway data has been described in an earlier paragraph. By using two or three photoelectric cells in line along the center line of a lane, it is also possible to compute velocity and acceleration. This technique will be investigated and

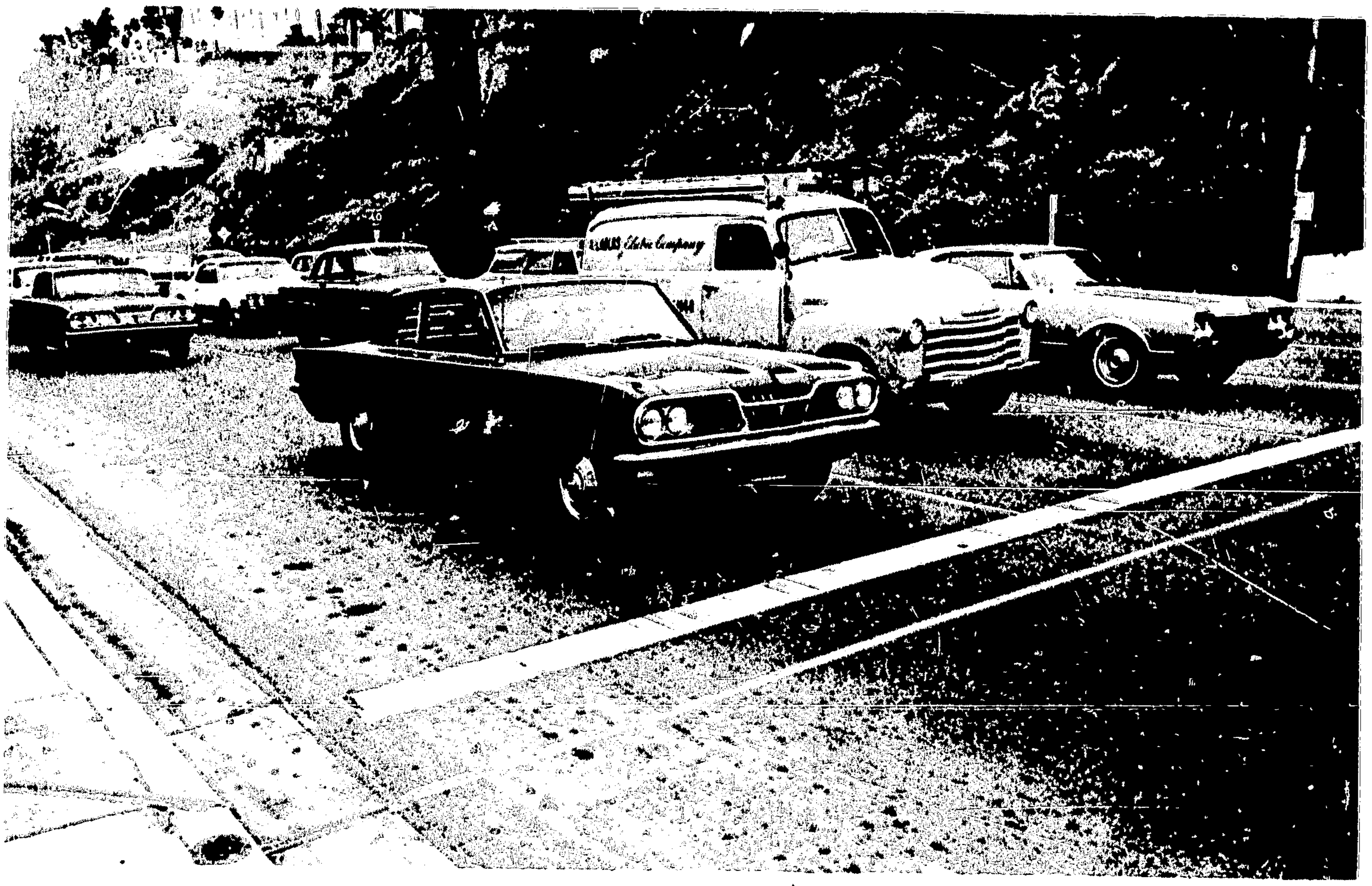


FIGURE 8-4. INTERSECTION WITH PHOTOELECTRIC CELL AND WIRES IN FRONT OF STOP LINE

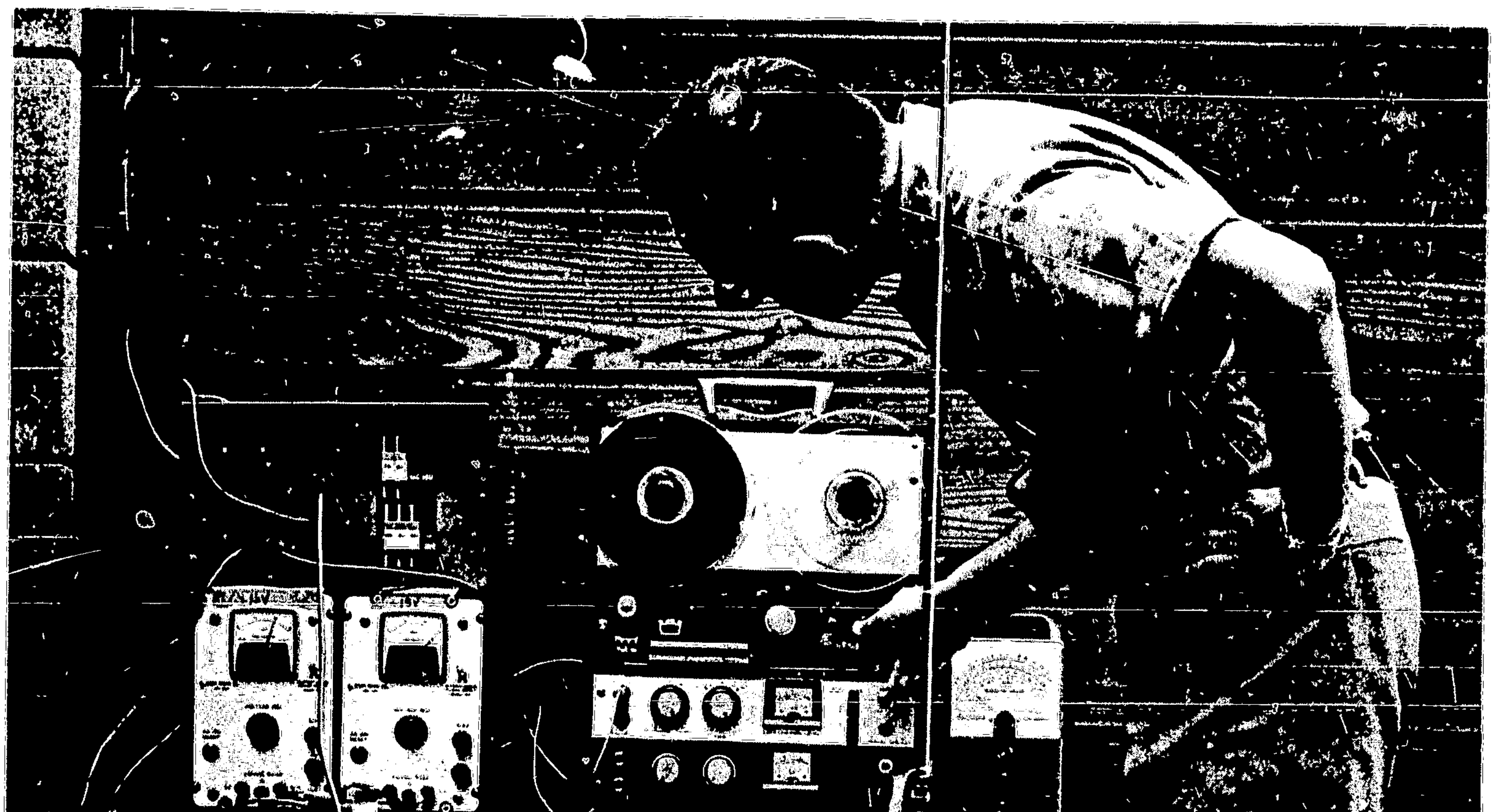


FIGURE 8-5. ELECTRONIC EQUIPMENT RECORDING IMPULSES FROM PHOTOELECTRIC CELL IN STREET

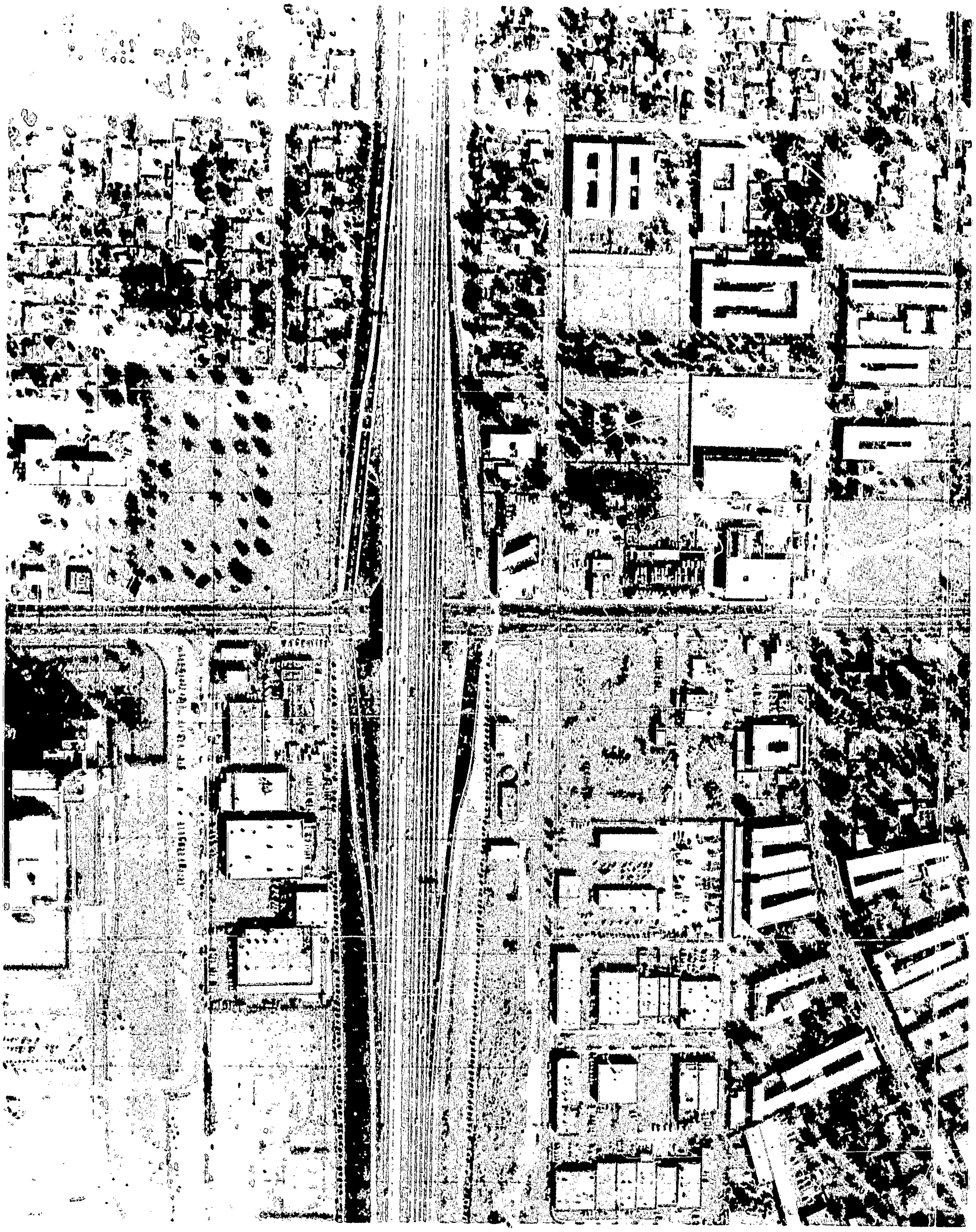


FIGURE 8-6. PHOTOGRAPH OF ROSCOE BLVD.-SAN DIEGO FREEWAY DIAMOND INTERCHANGE TAKEN WITH 70 MM. PULSE SEQUENCE CAMERA 3365 FEET ABOVE GROUND LEVEL

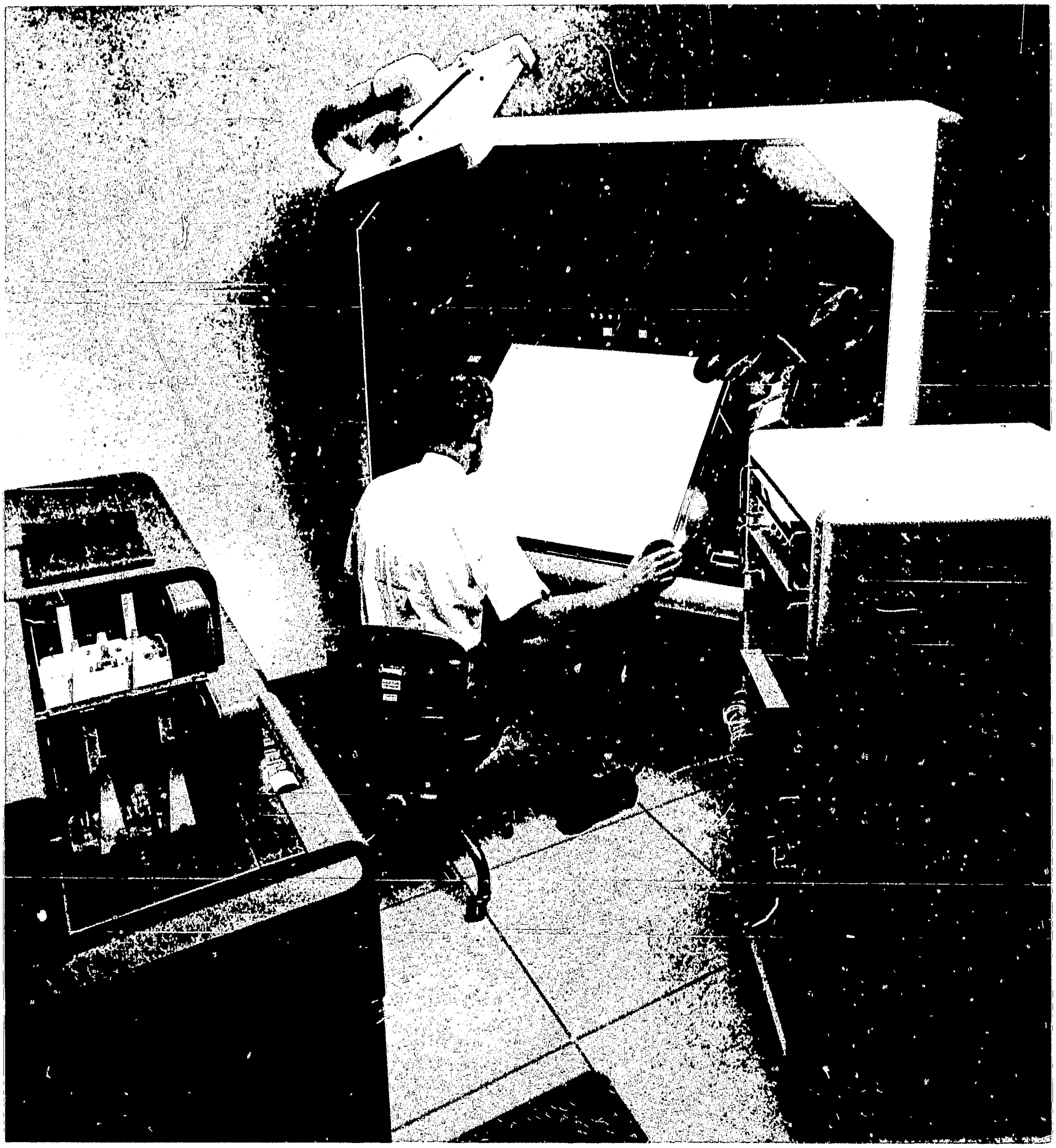


FIGURE 8-7. FILM READER AND PERIPHERAL EQUIPMENT IN OPERATION

Operator reads vehicle position from film image and causes the parameters of the vehicle to be punched on cards.

possibly used for determining driver performance characteristics to be used in the model.

Data Analysis

The principal figure of merit that will be used to validate the model is travel time by origin destination (in Model 1, the two origin destination combinations are (1) freeway-in/freeway-out, and (2) on-ramp-in/freeway-out). Travel times measured in the field will be compared with those obtained from the computer simulation when operated under conditions identical to those in the field. Other performance measures to be used include velocity, density, and number of vehicle passings.

The comparison of the field data with computer output must involve statistical procedures, since the flow of traffic is stochastic in nature. Some of the problems and solutions are described in the section on Statistical Methods in Operations Research, p. 8-4.

Plans

Programming, coding, and debugging of Version 2, Model 1, will be completed. Programming of Model 2 will begin. Programming, coding, and debugging of the required software for using photographic data will be initiated and completed. Validation tests will begin for Version 2, Model 1.

Project Documentation

1. Gafarian, A. V., Hayes, E., Mosher, Jr., W. W. The development and validation of a digital simulation model for design of freeway diamond interchanges. SDC document SP-2159. April 5, 1966. 95 pp.
2. Hayes, E. Programming a digital simulation model of a freeway diamond interchange. SDC document SP-2228/000/01. June 23, 1966. 36 pp.

STATISTICAL DECISION PROCEDURES FOR MATHEMATICAL LEARNING MODELS

R. E. Dear

Consultant: R. C. Atkinson, Stanford University

Description

Since 1950, mathematical theories of learning have developed very rapidly. These theories have their mathematical foundations in certain aspects of the theory of stochastic processes. Several other SDC research activities make use of various mathematical learning models (e.g., computer-assisted instruction and human data processing). This project is intended to develop statistical decision procedures needed in these areas.

In statistical games, strategies or decision procedures for experimenters are often classified in terms of combinations of three basic components: (1) the choice of an experiment (sample space), (2) the choice of a sampling plan (stopping rule), and (3) the choice of a terminal decision function that enables the experimenter to draw conclusions. Most statistical decision problems can be described by indicating which of these three strategic components may be considered as given and conversely which components may be manipulated to obtain an optimal procedure.

Progress

We considered several types of statistical decision problems, all having their basis in models of W. K. Estes' stimulus sampling theory of learning. Bayes solutions to parameter estimation for parameters in a one-element and a two-element model of the theory were described in last year's Annual Report. Those results have been revised to emphasize their relationship to results obtained from other principles of estimation. The revised version will appear in Journal of Mathematical Psychology (see TM-1734/002/00 under Completed Studies).

Research was initiated on Bayes and likelihood ratio solutions for determining the number of elements operating in applications of multi-element models of stimulus sampling theory. Computational aspects of these solutions are currently being studied.

W. Karush and R. E. Dear have been studying methods of obtaining optimal designs (presentation strategies) for learning experiments. This work is summarized under Mathematical Programming, p. 8-5. A report on an experiment which utilized certain kinds of optimal presentation strategies will appear in Behavioral Science [1]. This was one of several such experiments carried out by R. E. Dear, with H. F. Silberman and D. P. Estavan of the Education and Training staff and R. C. Atkinson of Stanford University.

Plans

This project is in a terminal phase.

Project Documentation

1. Dear, R. E., Silberman, H. F., Estavan, D. P. and Atkinson, R. C. An optimal strategy for the presentation of paired-associate items. Behavioral Science, in press. (Also available as SDC document SP-1935/101/00.)

CELESTIAL MECHANICS AND DIFFERENTIAL EQUATIONS

R. B. Barrar

Description

The problems of celestial mechanics and the development of mathematics have always been closely related. To cite just a few instances: Newton developed calculus to solve the two-body problem; Cauchy investigated functions of a complex variable in order to study the convergence of Kepler's equation; and the astronomer Bessel introduced Bessel functions in connection with certain problems of astronomy.

During the last 25 years, before the advent of artificial satellites, celestial mechanics was neglected by mathematicians. However, with the many practical problems introduced by satellites, mathematicians have once again returned to a study of the field. The basic, unsolved problem of celestial mechanics is to develop methods of describing the motion (over all time) of $n (n \geq 3)$ bodies under the influence of the laws of gravity. This problem can be described, in more mathematical terms, as the search for methods of solving differential equations that give solutions valid for all time. These are the problems that motivate this project.

Progress

In celestial mechanics, one of the few problems that has been solved exactly, with the answer valid over all time, is the famous two-body problem dating back to Newton. The solution is that both bodies move around their center of mass in elliptical orbits. If the number of bodies is three or more, all known solutions of the equations of motion are only approximations. In the case of the sun-earth-moon system, these approximations are very accurate; but from a mathematical viewpoint the results are still not correct for all time.

One type of orbit that has been studied extensively in the past is the periodic orbit. A periodic orbit is one that returns to its initial position and velocity after some time and then continues to repeat this process. Hence, if the trajectory of a periodic orbit is known for one period, it is known for all time. In studying the motion of several bodies, it is natural to define a periodic orbit as one in which all of the bodies return to their initial position and velocity at the same time. One of the main reasons that periodic orbits are studied is that Poincaré, the greatest mathematical astronomer of the last century, conjectured that the motion of any arbitrary orbit can be approximated as closely as one pleases by the motion of a periodic orbit. This is an approximation in the same sense that any real number can be approximated arbitrarily closely by a rational number.

One of the most far-reaching methods used by Poincaré to study periodic orbits was his last geometric theorem. Birkhoff was the first mathematician to give a rigorous proof of the theorem. In [3] we have given a modern proof of the theorem that depends on topological tools not utilized by Birkhoff.

Another approach to the study of orbits, which has been given great impetus in the last few years by mathematicians of such stature as A. N. Kolmogorov and C. L. Siegel, is the study of quasiperiodic orbits in which the orbiting object almost returns to its original position and velocity (in a given period) but never does exactly. In the sense of modern measure theory, the density of rational numbers is insignificant when compared to the density of real numbers. Similarly, periodic orbits are insignificant in number when compared to quasiperiodic orbits. In [1], we have formulated more abstract procedures for the proof of the existence of quasiperiodic orbits, so that they may be freely utilized. We have extended these procedures to cases not previously studied [2].

Plans

One of the great limitations in studying both periodic and quasiperiodic orbits is that all results apply only to very small perturbations. It is hoped that methods may be developed with greatly extended ranges of validity.

Project Documentation

1. Barrar, R. B. Smoothing operators, Siegel's theorem and celestial mechanics. SDC document SP-2335. January 20, 1966. 18 pp.
2. Barrar, R. B. Small eccentricity and the restricted problem of three bodies. SDC document SP-2394. March 17, 1966. 14 pp.
3. Barrar, R. B. Proof of the fixed point theorems of Poincaré and Birkhoff. SDC document SP-2235. October 4, 1966. 20 pp.
4. Barrar, R. B. A proof of the convergence of the Poincaré Von Zeipel procedure in celestial mechanics. American Journal of Mathematics, 1966, 88, 206-220. (Also available as SDC document SP-1926.)
5. Barrar, R. B. Existence of conditionally periodic orbits for the motion of a satellite around the oblate earth. Quarterly of Applied Mathematics, 1966, 24, 47-55. (Also available as SDC document SP-1935.)

DATA COMPRESSION TECHNIQUES

N. M. Dor

Description

This project is investigating techniques for improving the efficiency of transmission of useful information from source to observer (or experimenter). In this connection we are concerned with correct sampling of the source; processing to extract required information; coding for optimal transmission; reception, decoding and reprocessing for the final extraction and display of relevant information; and adaptive feedback processing techniques. The subject has become more important in the aerospace industry because large amounts of redundant data are being transmitted from space vehicles and stored. Related subjects have been called data compaction, data editing, redundancy reduction, and information processing. Although the subject has been related to space communications, it is of more general interest and could be applied to such diverse fields as medical analysis and general information handling.

Progress

This project began in June 1966. A large amount of literature has been reviewed and a survey of the present state of the art has been established. Two facets of the subject have been examined: computer buffer storage requirements for data compression under certain conditions [1], and more flexible encoding techniques using variable-length messages.

Plans

It is planned to examine the efficiency of machine-aided techniques whereby a human monitor may extract relevant information from a data source and adaptively improve preprocessing.

New methods for coding variable-length words for continuous transmission to improve the compression in transmission will also be studied.

Project Documentation

1. Dor, N. M. Guide to the length of buffer storage required for random (Poisson) input and constant output rates. SDC document TM-3118. October 10, 1966. 9 pp.

COMPLETED STUDIES

The following studies conducted by the Mathematics and Operations Research staff were completed prior to 1966 and are not described in this report.

MATHEMATICAL MODELS OF STOCHASTIC SYSTEM ELEMENTS

Waiting Line Models

1. Ancker, Jr., C. J. and Gafarian, A. V. Queueing with multiple Poisson inputs and exponential service times. Operations Research, 1961, 9(3), 321-327. (Also available as SDC document TM-503.)
2. Ancker, Jr., C. J. and Gafarian, A. V. Queueing with impatient customers who leave at random. Journal of Industrial Engineering, 1962, 13(2), 84-90. (Also available as SDC document SP-210/000/02.)
3. Ancker, Jr., C. J. and Gafarian, A. V. Some queueing problems with balking and reneging--I. Operations Research, 1963, 11(1), 88-100. (Also available as SDC document SP-372/001/01.)
4. Ancker, Jr., C. J. and Gafarian, A. V. Some queueing problems with balking and reneging--II. Operations Research, 1963, 11(6), 928-937. (Also available as SDC document SP-327/002/01.)
5. Ancker, Jr., C. J. and Gafarian, A. V. Queueing with reneging and multiple heterogeneous servers. Naval Research Logistics Quarterly, 1963, 10(2), 125-149. (Also available as SDC document SP-372/000/01.)
6. Krishnamoorthi, B. On the concept of FIFOV in queues. Operations Research, 1965, 13(3), 365-374. (Also available as SDC document SP-1729.)
7. Krishnamoorthi, B. On the concept of FIFOV in queues and a detailed study of the case of a Poisson queue with two heterogeneous servers. SDC document SP-1729/001/00. September 16, 1964. 33 pp.
8. Coffman, Jr., E. G. and Krishnamoorthi, B. Preliminary analyses of time-shared computer operation. SDC document SP-1719. August 19, 1964. 32 pp.
9. Krishnamoorthi, B. The stationary behavior of a time-sharing system under Poisson assumptions. SDC document SP-2090. September 3, 1965. 26 pp.
10. Krishnamoorthi, B. and Wood, R. C. Time-shared computer operations with both inter-arrival and service times exponential. Journal of the ACM, 1966, 13(3), 317-338. (Also available as SDC document SP-1848/000/01.)

Models of Combat

1. Williams, G. T. and Ancker, Jr., C. J. Stochastic duels. Operations Research, 1963, 11(5), 803-817. (Also available as SDC document SP-1017/000/01.)
2. Ancker, Jr., C. J. Stochastic duels with limited ammunition supply. Operations Research, 1964, 12(1), 38-50. (Also available as SDC document SP-1017/001/00.)
3. Ancker, Jr., C. J. and Williams, G. T. Some discrete processes in the theory of stochastic duels. Operations Research, 1964, 13(2), 202-216. (Also available as SDC document SP-1017/002/00.)
4. Williams, G. T. Stochastic duels--II. SDC document SP-1017/003/00. September 13, 1963. 61 pp.
5. Ancker, Jr., C. J. and Gafarian, A. V. The distribution of rounds fired in stochastic duels. Naval Research Logistics Quarterly, 1964, 11(4), 303-327. (Also available as SDC document SP/1017/004/00.)
6. Williams, G. T. Stochastic duels--III. SDC document SP-1017/006/00. June 22, 1964. 72 pp.
7. Ancker, Jr., C. J. The status of developments in the theory of stochastic duels. Proceedings Fourteenth Military Operations Research Symposium, Fall 1964, 260-269.
8. Williams, G. T. Stochastic duels with homing. SDC document SP-1017/106/00. May 18, 1965. 34 pp.

STATISTICAL METHODS IN OPERATIONS RESEARCH

General

1. Pugh, E. L. The exact distribution of the Kolmogoroff-Smirnoff statistic. SDC document SP-647. May 2, 1962. 75 pp.
2. Walsh, J. E. Distribution-free tolerance intervals for continuous symmetrical populations. Annals of Mathematical Statistics, 1962, 33(3), 1167-1174.

3. Walsh, J. E. Some two-sided distribution-free tolerance intervals of a general nature. Journal of the American Statistical Association, 1962, 57(4), 775-784.

4. Gafarian, A. V. Confidence bands in straight line regression. Journal of the American Statistical Association, 1964, 59, 182-213. (Also available as SDC document SP-1181/000/01.)

5. Walsh, J. E. Approximate distribution of extremes for nonsample cases. Journal of the American Statistical Association, 1964, 59(2), 429-436. (Also available as SDC document SP-727/000/01.)

Techniques for Analysis of Medical and Biological Data

1. Walsh, J. E. Confidence intervals for difference of frequencies in matched samples. SDC document SP-979. October 1, 1962. 4 pp.

2. Walsh, J. E. and Geoghagen, R. R. M. A method of emphasizing specified alternatives in sequential and nonsequential tests for symmetry about zero. Proceedings Koninklijke Nederlandse Akademie van Wetenschappen, Series A, 1963, 66(2), 189-191. (Also available as SDC document SP-543/000/01.)

3. Walsh, J. E. Simultaneous confidence intervals for differences of classification probabilities. Biometrische Zeitschrift, 1963, 5(4), 231-234. (Also available as SDC document SP-1015.)

4. Walsh, J. E. Bounded probability properties of Kolmogorov-Smirnov and similar statistics for discrete data. Annals of the Institute of Statistical Mathematics, 1963, 15(2), 153-158. (Also available as SDC document SP-848/000/01.)

5. Walsh, J. E. Loss of test efficiency due to misclassification in 2x2 tables. Biometrics, 1963, 19(1), 158-162. (Also available as SDC document SP-728.)

6. Walsh, J. E. Nonparametric implications of random conversion of discontinuous data to nearly equivalent continuous form. SDC document SP-463/000/01. March 12, 1965. 18 pp.

Reliability Theory

1. Pugh, E. L. On reliability inference. SDC document SP-935/000/01. November 5, 1962. 15 pp.

2. Pugh, E. L. The best estimate of reliability in the exponential case. Operations Research, 1963, 11(1), 57-61. (Also available as SDC document SP-895.)

Monte Carlo Technique

1. Clark, C. E. The utility of statistics of random numbers. Operations Research, 1960, 8(2), 185-195. (Also available as SDC document SP-79.)

2. Clark, C. E. and Holz, B. W. Exponentially distributed random numbers. Baltimore: The Johns Hopkins Press, 1960.

3. Clark, C. E. Sampling efficiency in Monte Carlo analysis. SDC document SP-643. December 15, 1961. 23 pp.

4. Clark, C. E. Importance sampling in Monte Carlo analyses. Operations Research, 1963, 9(5), 603-620. (Also available as SDC document TM-505.)

5. Pugh, E. L. Some examples of stochastic distortion, a Monte Carlo technique. SDC document SP-1584. March 6, 1964. 17 pp.

6. Pugh, E. L. Some tests of randomness of the Philco pseudorandom-number generator. SDC document TM-2164. December 4, 1964. 7 pp.

7. Pugh, E. L. A gradient technique of adaptive Monte Carlo. SIAM Review, 1966, 8(3), 346-355. (Also available as SDC document SP-1921.)

8. Gafarian, A. V. and Ancker, Jr., C. J. Mean value estimation from digital computer simulation. Operations Research, 1966, 14(1), 25-44. (Also available as SDC document SP-2005/000/01.)

Factor Analysis Methodology

1. Harman, H. H. Modern factor analysis. Chicago, Illinois: University of Chicago Press, 1960.

MATHEMATICAL PROGRAMMING

General

1. Karush, W. A general algorithm for the optimal distribution of effort. Management Science, 1962, 9(1), 50-72. (Also available as SDC document TM-616.)

2. Karush, W. Duality and network flow. SDC document TM-1042/201/00. March 15, 1963. 38 pp.

3. Pugh, E. L. Gradient techniques for non-linear programming. SDC document TM-695. February 27, 1962. 21 pp.

4. Karush, W. Decentralized manpower allocation by shadow pricing. SDC document SP-1851. October 27, 1964. 12 pp.

5. Karush, W. On an unsolved problem in optimal sequencing. SDC document SP-1750. August 27, 1964. 8 pp.

The Maximum Transform

1. Karush, W. and Bellman, R. On a new functional transform in analysis: The maximum transform. Bulletin of the American Mathematical Society, 1961, 67(5), 501-503. (Also available as SDC document SP-370.)
2. Karush, W. and Bellman, R. Mathematical programming and the maximum transform. Journal of the Society of Industrial and Applied Mathematics, 1962, 10(3), 550-567.
3. Karush, W. and Bellman, R. On the maximum transform and semigroups of transformations. Bulletin of the American Mathematical Society, 1962, 68(5), 516-518. (Also available as SDC document SP-719.)
4. Karush, W. and Bellman, R. On the maximum transform. Journal of Mathematical Analysis and Applications, 1963, 6(1), 67-74.
5. Karush, W. and Bellman, R. Functional equations and the theory of dynamic programming--No. 12. An application of the maximum transform. Journal of Mathematical Analysis and Applications, 1963, 6(1), 155-157.

Strategies of Item Presentation in Learning Processes

1. Karush, W. and Dear, R. E. Optimal procedure for an N-stage testing and learning process--II. SDC document SP-1922/001/00. October 18, 1965. 18 pp.

Dynamic Programming and Control Theory

1. Ash, M. Bang-bang reactor control. Journal of Nuclear Sciences and Engineering, 1963, 16(2), 208-212. (Also available as SDC document SP-262/000/02.)
2. Ash, M. and Jones, W. Optimal strategies for maximum number games. Journal of Mathematical Analysis and Applications, 1964, 9(1), 138-140.
3. Ash, M. An intuitive notion of the maximum transform. SDC document TM-1341. June 28, 1963. 10 pp.
4. Ash, M. Application of dynamic programming to stochastic optimal time control. SDC document SP-1363. January 31, 1964. 10 pp.
5. Ash, M. Application of dynamic programming to optimal shutdown control. SDC document SP-2187. August 27, 1965. 24 pp.

VEHICULAR TRAFFIC STUDY

1. Stern, S. Traffic flow data acquisition using magnetic-loop vehicle detectors. SDC document SP-2101/010/00. July 19, 1965. 31 pp.

STATISTICAL DECISION PROCEDURES FOR MATHEMATICAL LEARNING MODELS

1. Dear, R. E. and Atkinson, R. C. Optimal allocation of items in a simple two-concept automated teaching model. In J. E. Coulson (Ed.) Programmed learning and computer-based instruction. New York: John Wiley and Sons, 1962.
2. Dear, R. E. Sequential design of experiments for alternative objective functions in automated teaching programs. SDC document TM-1161. April 9, 1963. 116 pp.
3. Dear, R. E. Bayes decision procedures for stimulus sampling models: I. Nonsequential experimentation. SDC document TM-1734. July 28, 1964. 59 pp.
4. Dear, R. E. Solutions for optimal designs of stochastic learning experiments. SDC document SP-1765. September 11, 1964. 27 pp.
5. Dear, R. E. Bayes estimation for some stimulus sampling models. Journal of Mathematical Psychology, in press. (Also available as SDC document TM-1734/002/00.)

CELESTIAL MECHANICS AND DIFFERENTIAL EQUATIONS

Orbit Studies

1. Barrar, R. B. Some remarks on the motion of a satellite of an oblate planet. The Astronomical Journal, 1961, 66(1), 11-15; 1962, 67(1), 105. (Also available as SDC document SP-184.)
2. Barrar, R. B. and Deutsch, R. Determination of satellite trajectories from track-while-scan radar measurements. Transactions of the Professional Group on Military Electronics of the Institute of Radio Engineers, 1961, Mil-5(4), 306-312. (Also available as SDC document TM-559.)
3. Barrar, R. B. Two-impulse transfer versus a one-impulse transfer--Analytic theory. American Institute of Aeronautics and Astronautics Journal, 1963, 1(1), 65-68. (Also available as SDC document SP-698.)
4. Barrar, R. B. An analytic proof that the Hohmann-type transfer is the true minimum two-impulse transfer. Astronautica Acta, 1963, 9(1), 1-11. (Also available as SDC document SP-723.)
5. Barrar, R. B. Periodic orbits about an oblate planet. SDC document SP-1067. January 9, 1963. 16 pp.
6. Barrar, R. B. Periodic orbits near the critical inclination angle. SDC document SP-1067/001/00. January 28, 1963. 11 pp.

7. Barrar, R. B. Nearly circular periodic orbits about an oblate planet. SDC document SP-1067/002/00. February 28, 1963. 5 pp.

8. Barrar, R. B. Existence of periodic orbits of the second kind in the restricted problem of three bodies. Astronomical Journal, 1965, 70(1), 3&4. (Also available as SDC document SP-1488.)

9. Barrar, R. B. New proof of a theorem of J. Moser concerning the restricted problem of three bodies. Mathematische Annalen, 1965, 160, 363-369. (Also available as SDC document SP-1488/001/00.)

Partial Differential Equations

1. Barrar, R. B. On Schauder's paper on linear elliptic differential equations. Journal of Mathematical Analysis and Applications, 1961, 3(2), 171-195.

2. Barrar, R. B. Some estimates for solutions of parabolic equations. Journal of Mathematical Analysis and Applications, 1961, 3(2), 373-397. (Also available as SDC document SP-226.)

MODELING IN OPERATIONS RESEARCH

1. Walsh, J. E. and Garfunkel, I. M. Method for first-stage evaluation of complex man-machine systems. Naval Research Logistics Quarterly, 1960, 7(1), 13-19. (Also available as SDC document SP-41.)

2. Walsh, J. E. and Brady, R. P. Efficiency criterion for comparing cargo aircraft in deliveries between two terminals. Operations Research, 1960, 8(3), 356-361.

3. Walsh, J. E. Remarks about methodology in operations research. Journal of the Operations Research Society of Japan, 1962, 4(2), 87-90.

4. Walsh, J. E. and Billing, P. B. A model for maritime transportation systems. Journal of the Operations Research Society of Japan, 1963, 6(1), 1-16. (Also available as SDC document SP-1053/000/01.)

5. Walsh, J. E. Approximate determination of costs for optimum air cargo operation. Proceedings of the 3rd International Conference on Operations Research. (Ed. by G. Kreweras and G. Morlat), Dunod, Paris. 1964, 297-306. (Also available as SDC document SP-735.)

6. Walsh, J. E. and Taylor, J. L. Simulation of military logistics operation in an underdeveloped country. Recherche Operationnelle et Problemes du Tiers Monde (Colloquium on the possibilities of operational research in developing countries), Dunod, Paris, 1964, 265-272. (Also available as SDC document SP-1090/000/01.)

7. Walsh, J. E. and Taylor, J. L. Planning by resource allocation methods--Illustrated by military applications. Operations Research, 1964, 12(5), 693-706. (Also available as SDC document SP-480/000/03.)

BIOMEDICAL MODELS AND ANALYSIS

1. Walsh, J. E. Analyzing medical considerations. IRE Transactions on Medical Electronics, 1960, ME-7, 362-366.

2. Walsh, J. E. Statistical prediction from tolerance regions. Bulletin of the International Statistical Institute, 1961, 38(3), 313-317.

3. Walsh, J. E. Computer-feasible method for handling incomplete data in regression analysis. Journal of the ACM, 1961, 8(2), 201-211.

4. Eisenberg, H. B., Geoghagen, R. R. M. and Walsh, J. E. A study of surgical mortality probability for several pre-operative health conditions. Biometrische Zeitschrift, 1963, 5(3), 173-187. (Also available as SDC document SP-788.)

5. Eisenberg, H. B., Geoghagen, R. R. M. and Walsh, J. E. A general probability model for binomial events, with application to surgical mortality, Biometrics, 1963, 19(1), 152-157. (Also available as SDC document SP-612.)

6. Eisenberg, H. B., Geoghagen, R. R. M. and Walsh, J. E. Statistical analysis of some case histories of bacterial endocarditis. SDC document TM-1331. June 26, 1963. 26 pp.

7. Walsh, J. E. Use of linearized nonlinear regression for simulations involving Monte Carlo. Operations Research, 1963, 11(2), 228-235. (Also available as SDC document SP-487.)

8. Walsh, J. E., Eisenberg, H. B. and Geoghagen, R. R. M. Mathematical models for analysis of bacterial endocarditis data. SDC document SP-1619/000/01. August 31, 1964. 18 pp.

9. Eisenberg, H. B., Geoghagen, R. R. M. and Walsh, J. E. A general use of the Poisson approximation for binomial events, with applications to bacterial endocarditis data. Biometrics, 1966, 22(1), 74-82. (Also available as SDC document SP-1256.)

METHODOLOGY

1. Karush, W. On mathematical modeling and research in systems. SDC document SP-1039. November 28, 1962. 37 pp.

2. Karush, W. and Garvin, P. L. Data processing and mathematics. In P. L. Garvin (Ed.) Natural language and the computer. New York: McGraw-Hill, 1963. (Also available as SDC document SP-771.)

3. Karush, W. The use of mathematics in the behavioral sciences. In P. L. Garvin (Ed.) Natural language and the computer. New York: McGraw-Hill, 1963. (Also available as SDC document SP-807.)

MISCELLANEOUS

1. Ancker, Jr., C. J. and Gafarian, A. V. The function $J(x,y) = \int_0^x \frac{\gamma(y,\xi)}{\xi} d\xi$ — Some properties and a table. SDC document TR-3. April 1962. 36 pp.

COMPUTER CENTER DEPARTMENT

G. H. Dobbs, Manager

The Computer Center Department was chartered within the Research & Technology Division in February 1965 to design, develop, and operate a centralized time-shared computing facility in support of corporate and external users. Such a system has been under development and operation on a set of IBM S/360 computers.

In carrying out its responsibilities, the Department operates several computing facilities for the Corporation, including the Q-32 (IBM AN/FSQ-32) computer and facility, which supports the Corporate Research & Technology Laboratory, several remote SDC and non-SDC users, and small portions of corporate administrative data processing via time-shared software. It is anticipated that CCD will continue to operate the Q-32 facility until fall of 1967, when the machine will be retired.

During the reporting year the Philco 2000 computer, which had supported, for more than five years, the bulk of the information processing needs in support of SDC's Independent Research program, was sold to Pratt Institute to support its efforts in engineering and education.

The replacement computer systems for these machines, a series of system 360 computers, have been operating over the period of the last year in the form of a 360 Model 50, which was operational from October 1965 through July 1966, and a 360 Model 65, which has been in operation from July 1966 to the present.

It is anticipated that, over time, all of the work currently handled on the Q-32, plus the work that was formerly handled by the Philco 2000 and an IBM 7094, will be absorbed by the 360/65. This system is already supporting extensive system and production testing of a new simulation production programming system designed and implemented by SDC's Defense Systems Division. In addition, conversion of Research & Technology Division programs from the Q-32 to the 360 has begun. At the same time, extensive use of the facility is being made by the CCD Programming Branch in support of the development of a time-shared software capability.

In addition to operational responsibility for the Computer Center, the CCD is charged with the design, implementation, and system testing of a time-shared multiple-access operating system on the 360. Design and development of this capability were begun in late 1965 and have reached the stage where a minimally useful capability was demonstrable on November 28, 1966. The minimal time-shared capability provides for on-line executive and debugging support of six interaction terminals with predicted mean time to failure of approximately one hour. It is projected that a reasonably complete time-shared software capability will be made available to users in January of 1967. In addition to the basic executive system capability implied by the time-shared operating system software, the Department is responsible for a variety of supporting aids. These include the 360 JOVIAL compiler, on-line and off-line debugging tools, program and text editing, file maintenance, and assembly language processors.

A review of progress over the last year indicates that the Department was only partially successful in achieving some of its immediate goals. Problems that contributed to the inability to perform to expectations have been attributable to late equipment deliveries, unreliable and unpredictable equipment performance, software design problems, and turnover in design personnel. However, most of the difficult problems appear to be in the past. As indicated by the descriptions of activities in the programming and operations activities, the Department is on the verge of making available, for corporate use, a system and facility that will allow time-shared multiple access by a variety of SDC professional users and support them in an efficient, economical way with advanced data processing techniques.

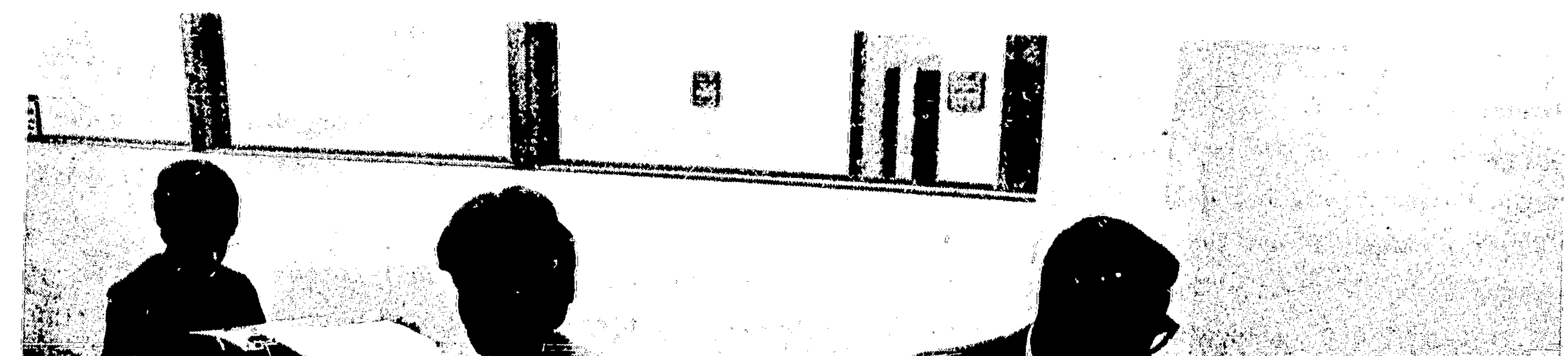


FIGURE 9-1. ON-LINE USERS OPERATING SDC'S S/360 MODEL 65 TIME-SHARING SYSTEM

PROGRAMMING BRANCH

C. A. Irvine, Head
W. C. Dennis

The Computer Center Programming Branch is responsible for the development of the general-purpose, time-sharing, operating system for the IBM 360 through which multiprogram systems and individual programs mav be operated. The system will provide programming-oriented system support tools such as compilers, interactive debugging systems, file maintenance routines, and report and so t generators. In addition, software will be provided that requires a minimum of programming training, such as data management and test-processing services. The responsibility for the design and implementation of these facilities rests with the Operating Systems Group and the Support Systems Group.

Special Projects Staff

E. DuBois
P. Kribs
W. J. Erickson

The Special Projects Staff was formed to assist the Branch by initiating the investigation of potential problem areas. They currently are assisting in the planning and scheduling activities, the establishment of charging and scheduling algorithms, preparation of documentation guidelines and schedules, and development of techniques for absorption of IBM software systems into the SDC operating system.

Operating System

P. A. Cramer, Head	D. J. McAllister
W. C. Brandstatter	R. B. McCracken
K. A. Hinman (on loan from ASD)	S. Perlman
W. A. Hudson	E. J. Siegal
G. W. Ingram	J. A. Slaybaugh
J. Krinsky	P. J. Sykes
	A. Tschekaloff

Description

The time-shared operating system being developed for the IBM S/360 computer, Model 65, is designed to permit up to 60 users to employ the computer concurrently in an interaction mode, with a nominal response time of five seconds. At least one user program will be guaranteed immediate service not to exceed 500 milliseconds. Noninteractive or background jobs will normally be operated at a lower priority, to fill out the remainder of slack-time cycles, although they can be automatically raised to a higher priority to meet a deadline.

The operating system is file-oriented, i.e., all information is processed in the form of files. This includes initial inputs, final outputs, stored intermediate results, source-language programs, executable programs, etc. Portions of the operating system are stored as executable files for loading as required, and swapping involves a "partitioned" file containing the pages of the current time-shared programs.

The operating system is constructed in two parts. One part resides permanently in core memory. The other part is a set of executable files. The operating system is divided logically into the Executive, the service function, and peripheral support functions.

The Executive is composed of four parts. The Cataloger is responsible for the maintenance of auxiliary storage, the catalog of files contained therein, and control of physical volumes (e.g., disc packs and tape reels). The Allocator oversees the subdivision of current memory and swap storage, and provides loading and swapping functions. The Dispatcher controls logical input/output operations for the system and users as well as physical input/output operations for auxiliary storage, peripheral devices, and interaction consoles; monitors input/output interrupts; and initiates necessary error recovery procedures when required. The Scheduler initiates and maintains jobs and tasks and governs the scheduling and switching of tasks.

The resident service functions set and maintain the hardware timer and also maintain the current time of day and date. Accounting support of the system is also provided, including recording of CPU, auxiliary storage and peripheral device utilization. The nonresident service functions provide user sign-in (LOGIN) and sign-off (QUIT), and operator control of the system.

Peripheral support consists of a set of resident and nonresident components to support four types of peripheral devices: card reader, card punch, printers, and interaction consoles.

Progress

The batch-processing Interim Operating System was extended through four releases during 1966. These extensions included the addition of a Basic Assembler; storage protection for the user and system; overlapped card-to-tape, tape-to-printer, tape-to-punch operations; increased disc utilization including systems residence, increased capability through a system access method to insert, modify, and delete records on disc storage; additional operator controls; and the conversion and expansion of the system for the Model 65 configuration.

Users began testing a prerelease version of the Time-Sharing System in October 1966. This initial version was formally released to the user community in late November 1966. This version supports IBM 1050 terminals and utilizes IBM 2311 discs for swap storage. An initial debugging capability is provided that allows the user to stop his program, display registers and locations of his program, make patches, and return control either to the point of interruption or to a specific location. The major emphasis of this initial version has been upon system reliability. It is expected that the mean free period to error of the system upon formal release will exceed one hour.

Plans

The first implementation of extended capabilities for the Time-Sharing System is planned for January 1967. These capabilities will include the ability to run background jobs, peripheral supervisor support for line printers, additional input/output capability for blocking and buffering, and extensions to Cataloger support. Follow-on stages in the implementation of the time-shared operating system call for: recoding of key portions to obtain more concise programs; revisions to the scheduling and core allocation algorithms as determined by hardware changes; additional catalog and device support for the IBM 2321 Data Cell; additional device supervisors and utility routines for display consoles; general tuneup of the system as indicated by characteristics of the load that it assumes; and investigation of methods for transferring the time-sharing system to an IBM S/360 Model 67.

Project Documentation

1. Kennedy, P. R. A user's tutorial for the S/360 Time-Sharing System. SDC documents TM-2800/000/00--004/00.
2. Operating System Group. S/360 operating system--Stage 3. SDC TM-2614 series.

Support Systems

J. B. Porges, Head
J. C. Hale
M. Martin
J. D. McCabe
L. Padgett
N. A. Sandin
G. H. Weisbord

Description

This project is responsible for developing support-type program systems for use in the SDC S/360 Time-Sharing System. To that end, the project is currently concerning itself with the JS-S/360 compiler, the basic S/360 assembler, the on-line debugger, and file maintenance and editing.

Progress

Early this year the project assumed responsibility for continued development and maintenance of the JS-S/360 compiler. Immediately, efforts were applied to join under a single compiler system several adjunct tasks, namely, program compool generation, set-used generation, and object module generation. Since that point, the capability of generating and retrieving data from a master compool has been incorporated. Currently three activities are being pursued: maintenance of the batch-system version of the compiler, checkout of a version for inclusion within the Time-Sharing System, and changes to make the compiler conform to Basic JOVIAL specifications.

The project has recently assumed responsibility for maintenance of the basic S/360 assembler.

In March 1966 the project released a debug system for inclusion within the interim operating system. Via a fully symbolic language, capability is provided to the user for data reduction and generation and conditional operations. Debug was retrofitted for the on-line operation and is currently being checked out on the Time-Sharing System. In addition to providing the user with a means of program testing, debug is to be utilized by the system to service interrupts induced by object program errors.

In May 1966 design of a file maintenance and line editor started. At present they have been checked out and are operational within the Time-Sharing System. Extensions to provide limited capabilities for document production and editing have been added to the line editor and are currently in checkout phase.

Plans

After the changes required to insure conformity with Basic JOVIAL specifications are incorporated in the compiler, considerable effort will be spent to optimize the code generated. Also, mechanisms will be developed to utilize common code and enhance the Time-Sharing System's capability to reduce swap time and storage allocation via inclusion of external symbols, control sections, and development of a linkage editor. Providing these will enable the library, currently in symbolic format, to be in object module format, thereby reducing compile time.

It is also planned to expand debug capabilities to include such things as limited tracing, flow charting, and utilization of display consoles.

The requirements for a context editor are presently being spelled out with anticipated implementation in mid-1967.

Other long-range goals include, where feasible, incorporating such things as the IBM OS/360 Assembler and Fortran IV and PL/I Compilers in the SDC Time-Sharing System.

COMPUTER CENTER ENGINEERING

R. K. Gray, Head
M. A. Painter

The Computer Center Department Engineering staff serves to resolve a broad range of technical problems by conferring with CCD and laboratory users and maintenance personnel to define requirements, prepare technical specifications and procedures, conduct studies, and design hardware where appropriate.

Through participation in the Laboratory Equipment Committee, the staff has assisted in the selection of a large-scale graphical display system (Figure 9-2 shows one of the consoles) and a multiterminal tabular display system. A RAND Tablet interface has been designed for the IBM S/360. The unit is now in checkout and will be modified for use with the IBM 1800 Process-Display Controller when that equipment is installed in the spring of 1967.

The staff has worked with members of the Mathematics and Operations Research staff, Research Directorate, on their Vehicular Traffic Study (see p. 8-12). The staff evaluated the requirements for validation data for the diamond interchange digital simulation model. This led to the selection of the components of a photographic system for acquiring the validation data, including aerial camera, a film reader, and other photographic and data processing equipment.

FIGURE 9-2. IBM 2250 DISPLAY CONSOLE UNIT

A device for measuring traffic headway and entering the data directly into the computer system was designed and used to gather fundamental information for the traffic study project.

A significant portion of this year's work has been concerned with computer terminal equipment and the local and remote communications problems encountered in its use. A comparison of printer-keyboard devices is in preparation and reports will be prepared on acoustical coupling, remote communication requirements, communication tariffs and other related subjects. A communications panel will be specified and obtained to provide increased flexibility in coupling terminals to the S/360.

STATISTICAL SERVICES

R. L. McCornack, Head
S. G. Swerdlow
L. T. Villone

Description

This project provides the corporation with a full range of services relating to statistical analysis of data by computer, including statistical consulting, preparation of data for input to the computer, processing of data using a library of about 50 programs, and the writing of special-purpose programs.

Progress

The volume of data processing requests continues very heavy. In a typical month, 198 jobs were processed--about 10 per working day. The bulk of these jobs consisted of runs on the IBM 7094 computer, the average run time being 3 minutes. About 80 percent of all runs were accomplished using only 10 of the 50 programs. However, about 15 percent required writing a special program to satisfy the request. Requests for data processing were received from throughout the corporation. At least 50 jobs were processed from the Personnel, Research, and Technology Directorates, and Operational Systems, Training Systems, and Western Development Departments.

Maintenance of the present program library has been minimal, due to the anticipated change to the IBM S/360 computer. Present data processing capabilities must be converted to run on the new computer. A new program system has been designed and partially coded. The Preprocessor, Frequency Distribution, Correlation, and Analysis of Variance programs have been written and checked out on the IBM 7094. Substantial progress has been made on the Plotting, Multiple Regression, Factor Analysis, and Missing Data Correlation programs.

Plans

The main goal is to maintain the present level of services with a minimum of disruption due to computer changes. Final checkout of the new programs must await a FORTRAN capability on the IBM S/360. The most commonly requested types of analyses will be converted first, and other programs will be converted on a user-demand basis.

RESEARCH AND TECHNOLOGY LABORATORY

W. J. Hanna, Head
M. H. Butler
P. E. Chaney
F. T. Fiala
J. J. Stateler
F. T. Steo

Background

The R&T Laboratory represents the combined facilities of the former Command Research Laboratory and the Systems Simulation Research Laboratory, which were merged in 1965, plus the facilities of the nascent S/360 system.

The Command Research Laboratory, built around the IBM AN/FSQ-32 computer, has been operational since the spring of 1962. The system continued to be used for the SDC/ARPA Time-Sharing System throughout the year, although ARPA support for the project ended on 17 November 1966. Computer

FIGURE 9-3. MAIN CONSOLE AREA OF Q-32 COMPUTER

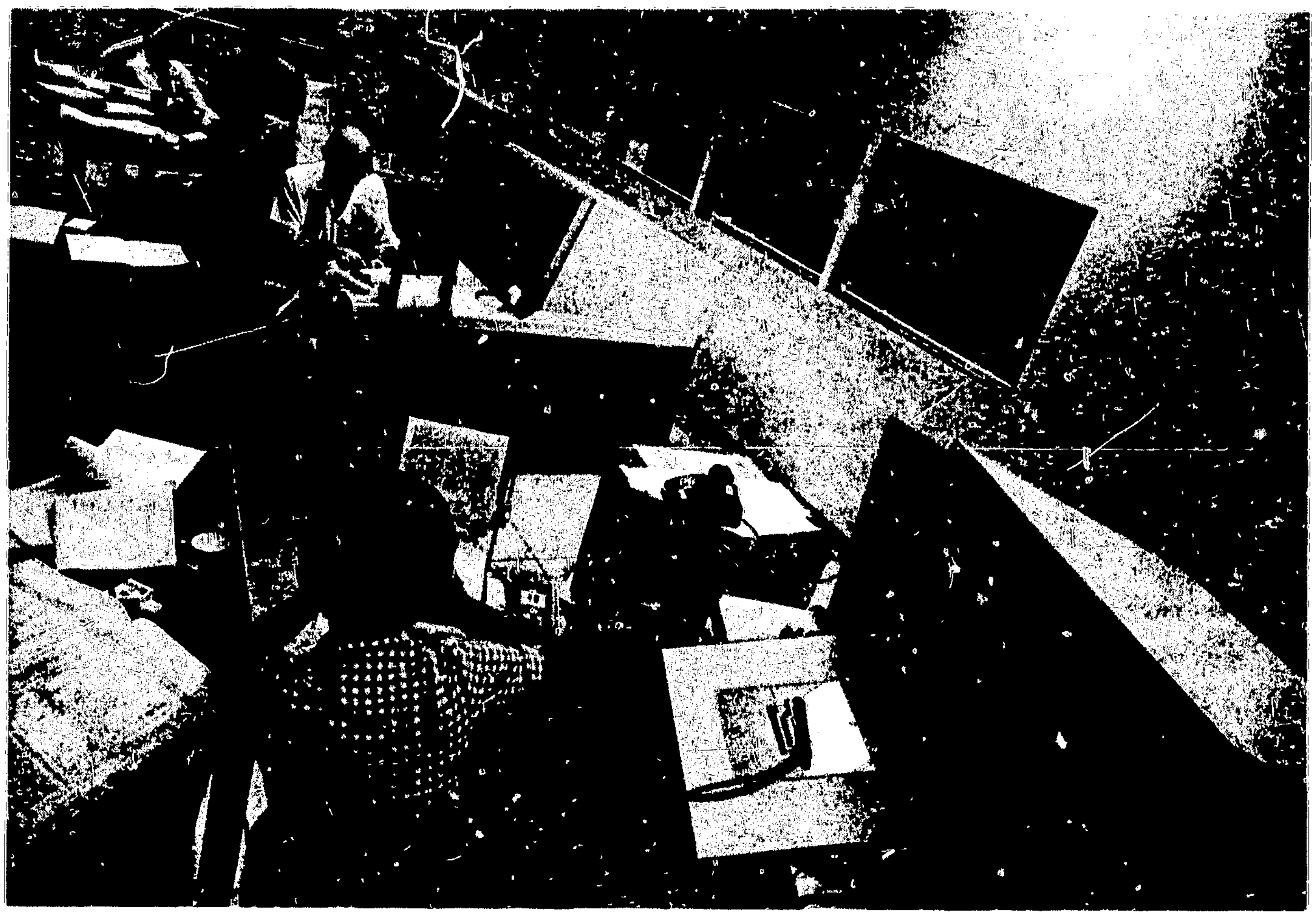


FIGURE 9-4. SECTION OF R&T LABORATORY

support for the remainder of the calendar year from 18 November was obtained by selling shares to various SDC and non-SDC agencies.

The Systems Simulation Research Laboratory, built around a Philco 2000 computer, was a general-purpose laboratory for the study of man-machine systems. With the sale of the Philco 2000, operation of the SSRL was terminated 30 September 1966 after five years of service.

The Computer Center Department started building up the required S/360 hardware to continue supporting the projects that had been using the Philco 2000 and Q-32 systems. The initial hardware installed to provide this support includes a Graphic Input Tablet and three Sanders Display Units interfaced to an IBM 2701 Control Unit, attached to an IBM S/360 Model 65. Additional Sanders display consoles, IBM Model 2250 display consoles, and IBM 1800 Process-Display Controller interfaced to the 360, and IBM 2741 communications terminals will be installed in 1967.

General Description

The Q-32 portion of the R&T Laboratory occupies space on two levels. The lower level contains most of the computing equipment and the operations console area; the upper level contains the display consoles, teletype stations, maintenance area, and additional communications facilities. The characteristics of the CRL equipment are given in Figure 9-5.

Since it became operational, the Q-32 has steadily increased in capability through modifications and equipment additions. A memory protection feature was added that safeguards against inadvertent user destruction of the time-sharing executive, or of another user's program. This feature allows protection of each of the five core storage units and certain input/output (I/O) operations.

A small computer, the PDP-1, was added to the Q-32 in 1963 to provide an efficient method of I/O processing that frees the Q-32 from the low-speed processing necessary for the many I/O stations in the time-sharing environment. The PDP-1 processes messages to and from the local and remote communication terminals and transfers these data to a high-speed core storage unit (providing 16,000 48-bit words of storage), which is also directly accessible by the Q-32. It also transfers light-pen inputs from the CRTs to the Q-32. This method of real-time I/O processing has enhanced system response time considerably.

To meet the requirements for additional storage for the many users of time-sharing, 4 million words of disc storage were made available by the addition of a parallel disc file to the high-speed peripheral control unit. The disc, which provides faster access than tapes, has improved the system efficiency for many users of large programs and has reduced tape handling. Due to the use of drums for swapping programs to and from core storage in the Time-Sharing System, the major bottleneck to system expansion proved to be lack of drum storage, because only two auxiliary storage drums delivered with the system could be used for swapping.

To increase drum space, the DATOR drum was modified to allow its use as an auxiliary storage drum and in 1965 two more drums were placed in operation, adding another quarter-million words of drum space. This additional capacity increased the maximum number of potential users from 18 to 30.

The Q-32 has an extensive built-in error-checking capability, with about 30 percent of the computer devoted to this type of circuitry. To permit effective use of this capability, an automatic recovery program called FIX was developed. This program, which operates whenever a machine malfunction occurs, has significantly improved system reliability.

COMPONENT	NUMBER	CAPACITY/SPEED	TOTAL
		AN/FSQ-32 COMPUTER	
Main Core Memory	4	16,384 words	65,536 words
. Cycle time 2.5 μsec.			
. 48-bit word			
Input Core Memory (Buffer)	1	16,384 words	16,384 words
. Cycle time 2.5 μsec.			
Drum	5	139,264 words	417,792 words
. Access time 11.5 ms.			
. Word transfer rate 2.75 μsec.			
Disc File	16 discs	262,144 words	4,194,304 words
. Access time 225 ms.			
. Word transfer rate 12 μsec.			
Tape Drives (729-IV)	12	112½ ips	
Card Reader (714)	1	250 cpm	
Card Punch	1	100 cpm	
Typewriter	2	100 wpm	
		ASSOCIATED COMPUTERS (ON/OFF-LINE)	
PDP-1			
. Shares input core memory of Q-32			32K words
. Cycle time 5 μsec.			
. 18-bit word main core memory	1	4K words	4K words
1401-D			
. Core memory	1	4K char.	4K char.
. Printer	1	600 lpm	
. Tape drives (729-IV)	1	112½ ips	
. Card reader (Uptime)	1	850 cpm	
		I/O DEVICES	
Teletypes and Typewriters			
. Model 33 Teletypes	33	100 wpm	
. TWX data sets (remote users)	8	100 wpm	
. Soroban typewriters	2	100 wpm	
. Telex data sets	1	100 wpm	
. Data-Phone sets	5	100 wpm	
. TX-2 Computer Link			
. IBM 1051/1052	1	150 wpm	
. Model 35 Teletype	1	100 wpm	
Communications Testboard	1		
Display Consoles	6	2K char. max. (per console)	
. Light pens			
. Vector-generator capability			
. Graphic tablet	1	5K points/sec.	
. Keyboard	2	100 wpm	
Telephones			
. Links for simultaneous conversations	12	100 wpm	
. Phones	35		

FIGURE 9-5. CRL EQUIPMENT CHARACTERISTICS

An extensive library of utility programs for general laboratory usage has been developed. The Utility Library, called UTLB, has over 40 programs available, consisting primarily of programmer tools such as tape-manipulation capabilities and disc-file manipulation and repacking.

The laboratory's six display consoles, which are driven directly from the Q-32 DATOR drum, are capable of displaying 2000 flicker-free characters. Each console is provided with a light pen from which signals are sent to the PDP-1. Two of the consoles have been modified for the General Purpose Display System. These two consoles each have a bank of pushbuttons and a special keyboard, and graphic input tablets. The graphic input tablet permits entering into the computer and/or onto the display scope an image of any line traced on the tablet by its associated stylus.

RAND Tablet Modification Project

A project was initiated to develop a new display technique in using the RAND Tablet. The display incorporates a computer-driven cathode ray tube projected on the surface of the RAND Tablet. The net effect is a coupling of display and tablet so that the viewing and writing areas are essentially the same. A breadboard version of this technique was successfully demonstrated on the Philco 2000, and the work is currently being converted to the Q-32 under time-sharing. This conversion is nearly complete from both hardware and software standpoints. The goal of this project is to investigate the advantages of a projected CRT/Tablet display in the areas of character recognition, graphics, and general application of high-resolution inputs to a computer. (This work is being done by Lou Gallenson of the Technology Directorate's Special Projects staff.)

Applications

Although a wide variety of research efforts have been conducted in the R&T Laboratory, the primary application during the past year continued to be time-sharing. The result has been that conventional batch processing has utilized only a very small portion of the available operating time.

Approximately 450 users were authorized access to the Q-32 until ARPA support ended on 17 November 1966. Users in the SDC Santa Monica facilities primarily used Model 33 Teletypes for communication. The remote users made use of the laboratory communication facilities including leased lines, TWX, and Data Phones.

Since 18 November, 1966, the following users are supporting the Q-32:

Outside Users

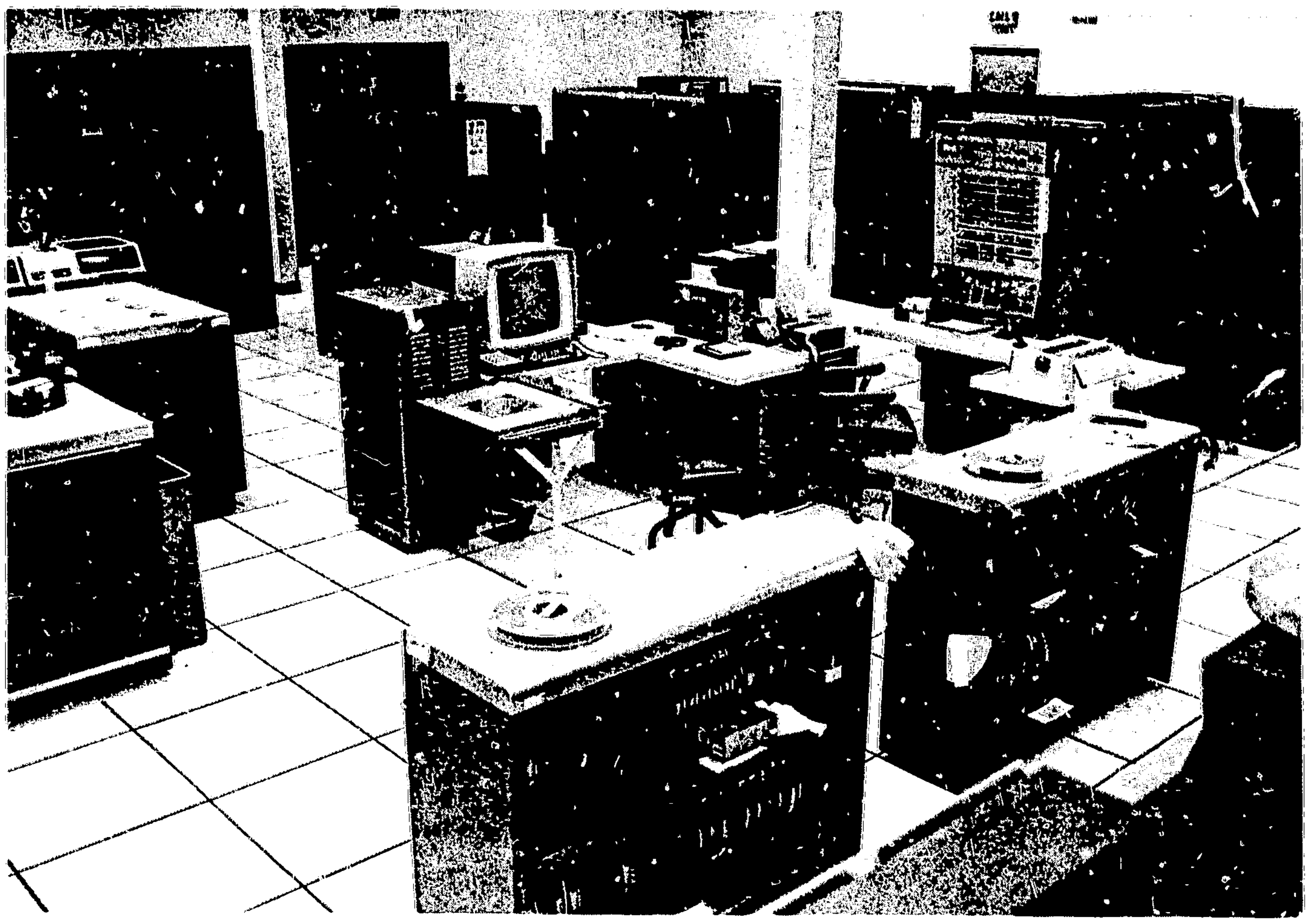
Advanced Research Projects Agency
A. F. Cambridge Research Laboratory
A. F. Personnel Research (AFPR2)
AFRRPC (DDS), Hq., U. S. Air Force
Defense Communications Agency
Hill Air Force Base
Kelley Air Force Base
National Bureau of Standards
National Institutes of Health (DCRT)
Navy Personnel Research
Richfield Oil Company
Southwest Regional Laboratory
Stanford Research Institute
University of Southern California
U. S. Department of Interior, Geological Survey

SDC Users

Research Directorate
Technology Directorate
Advanced Systems Division
Defense Systems Division

Demonstrations

The Time-Sharing System used on the Q-32 has been the object of considerable interest in the computing community, and therefore has been used extensively for demonstration purposes both at SDC and remote locations. One shortcoming in



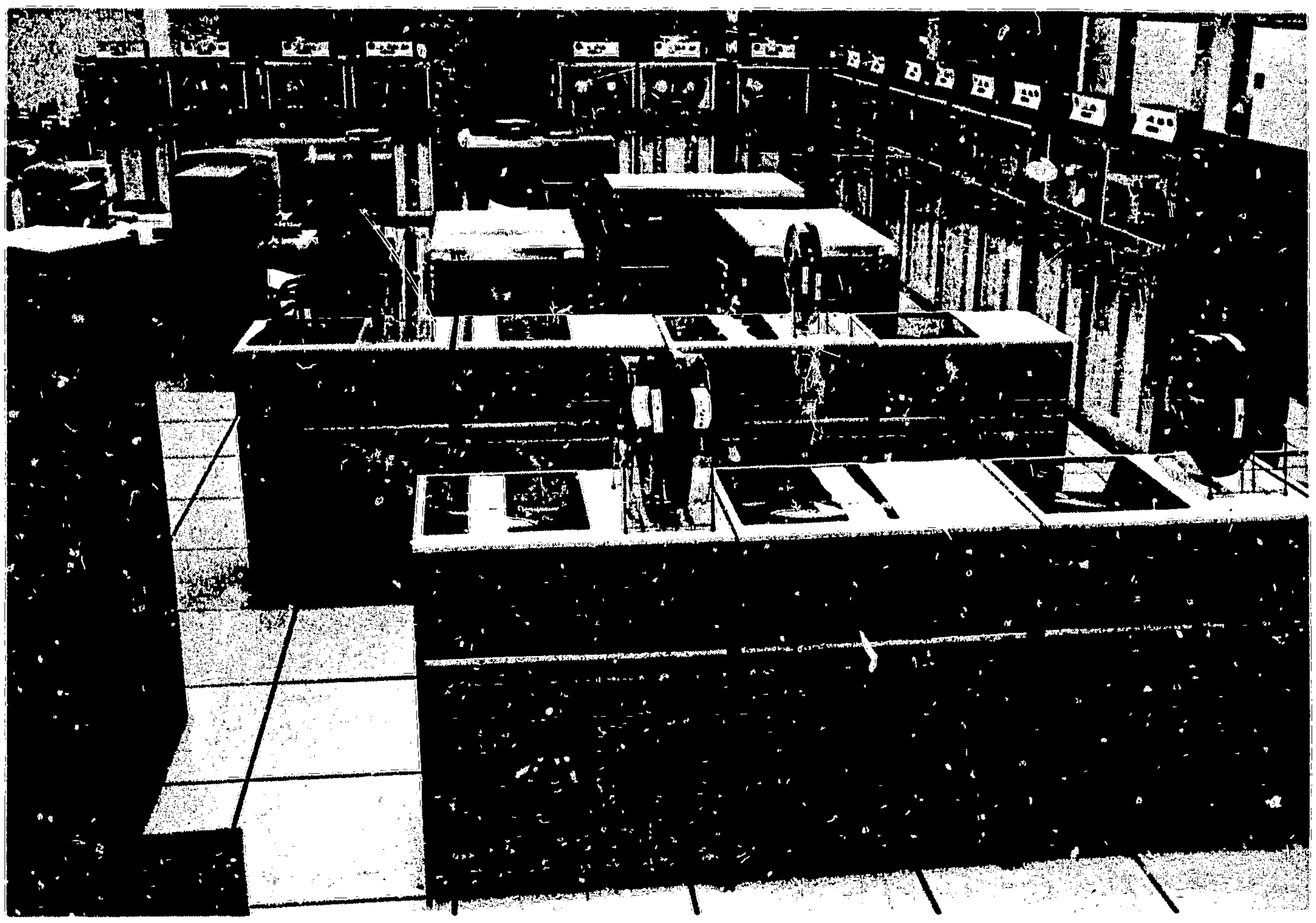


FIGURE 9-6. TWO VIEWS OF IBM S/360 MODEL 65 COMPUTER IN R&T LABORATORY

demonstrating time-sharing has been that it is difficult for more than two or three people at a time to see teletype printouts. To overcome this disadvantage, a system using closed-circuit television cameras and monitors was developed in 1966; the system allows up to 80 persons at a time to view the dynamic demonstration of TSS. An investigation is under way to find a suitable large-screen projection TV to expand this capability.

Systems Simulation Research Laboratory

The SSRL portion of the R&T Laboratory was phased out according to plan; the last day of operation was 30 September 1966. The equipment, including the Philco 2000 computer and associated input/output devices, was sold to Pratt Institute of Brooklyn, N. Y., and delivered to them in October 1966.

The first-floor space formally used by this system will be an expansion area for the S/360. The second-floor space will be used for laboratory areas for display and other equipment attached to the S/360.

Computer Center Operations

The Computer Center Department's S/360 equipment configuration continued to expand during 1966. The major change occurred on 1 August 1966 when the IBM S/360 Model 50I was replaced with an S/360 Model 65I. The equipment presently installed for the Model 65 is indicated in Figure 9-7 and the configuration is illustrated in Figure 9-8.

In addition to the IBM S/360 equipment, three Sanders Display Consoles are being interfaced to the Model 65. These devices will provide a small-scale tabular display capability for research activities. A RAND Graphic Input Tablet was also interfaced to the S/360 in 1966. Used in conjunction with the operator's IBM Model 2250 Display Console, the tablet provides an interim capability until the CCD laboratory is fully operational.

To provide better service and increased support for the varied user requirements of the CCD, the R&T Laboratory staff was integrated with CCD Operations. The CCD Operations staff now comprises: (1) the Equipment Operations activity, which is responsible for operating all equipment under CCD's jurisdiction; (2) Statistical Services (described on p. 9-7); and (3) Customer Liaison and Support, which provides liaison between the CCD and both vendors and customers, performs specialized operational programming services, and oversees the maintenance and installation of computer and laboratory equipment.

Plans

During 1967 it is planned that the Q-32 will be phased out and a new R&T Laboratory will be developed from equipment additions to the Computer Center Department's S/360.

The present area occupied by the S/360 Model 65 will be expanded to approximately 9,000 square feet to accommodate an S/360 Model 67 duplex computer, and an S/360 Model 50H computer to be operated by CCD for a special project in the Technology Directorate. The Model 50 installation will include two Sanders displays, two IBM 2250 display consoles, and eight IBM 2741 communications terminals. The dual 67 installation will replace the Model 65 presently installed.

Project Documentation

1. Butler, M. H. An automatic error-recovery system for time-sharing. SDC document SP-2436. April 20, 1966. 12 pp.
2. Chaney, P. E. and Linde, R. R. Operational management of time-sharing systems. SDC document SP-2392. June 6, 1966. 30 pp.
3. Fiala, F. T. Time-sharing operations and management. SDC document SP-2417. March 30, 1966. 15 pp.

COMPONENT	NUMBER	CAPACITY/SPEED	TOTAL BYTES
Main Core Memory 2365-2 . Cycle time .75 μsec. . Accesses 8 bytes (double word) in parallel	2	262,144 bytes	524,288
Drum 2301 . Average access time 8.6 ms . Byte transfer rate .8 μsec. . Rotates @ 3500 rpm	1	4,000,000 bytes	
Disc 2311 . Average access time 75 ms (25 ms min. to 135 ms max.) . Byte transfer rate 6.4 μsec. . Maximum number of bytes/track 20624 . Number of tracks = 200	8	7,250,000 bytes	58,000,000
Tape Drives 2402-3 . Tape Speed 112.5 ips . Density 800 bytes/in. . Data rate 90,000 bytes/sec.	16	40,000,000 bytes (@ 4800 bytes/block)	
Printer 1403-N1	2	1100 lpm	
2540 Reader 1000 cards/min.	1	2.67 KB/sec.	
2540 Punch 300 cards/min.		.80 KB/sec.	
2250 Display 8192 bytes buffer size . Grid area 1024 x 1024 points . Buffer to CPU transfer rate	1	240 KB/sec.	
2321 Data Cell Drive . 600 msec. maximum to any record . Transfer rate 55,000 bytes/sec.	1	418,000,000 bytes	
1052 Communications Terminals	9	14.8 char./sec.	
2741 Communications Terminals	25	14.8 char./sec.	

FIGURE 9-7. S/360 MODEL 65 EQUIPMENT

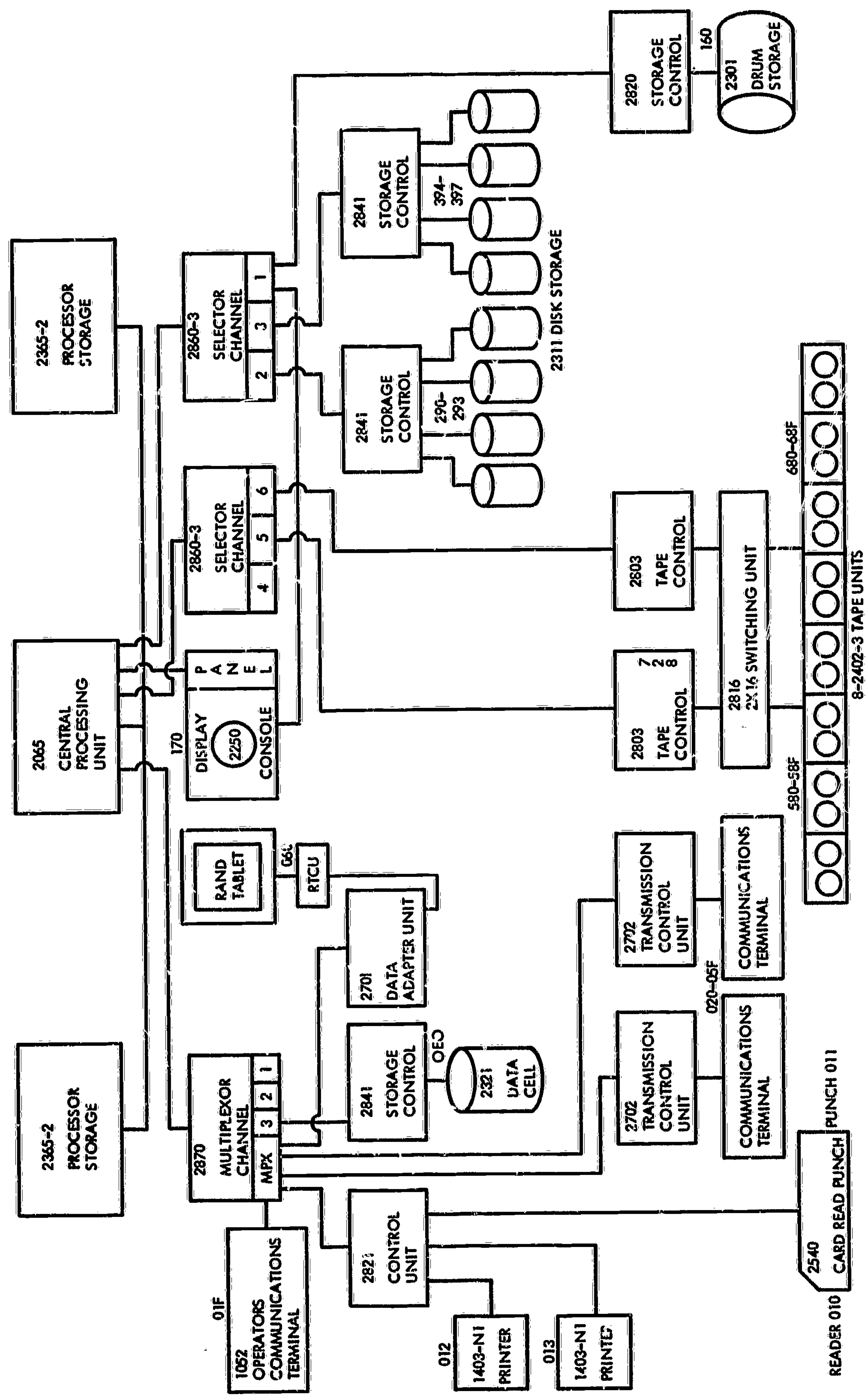


FIGURE 9-8. S/360 MODEL 65 CONFIGURATION

SPECIAL SERVICE OPERATIONS

The Research & Technology Division provides a number of special services in support of its own activities as well as those of the entire Corporation and appropriate outside agencies. In addition to the services described below, a major activity during the year was the more than 500 briefings and demonstrations of R&TD projects that were given to the external community. A large number of our technical personnel have contributed to this effort. A more detailed account of the demonstration activity is given on p. 3-6; an inventory of the more frequently demonstrated programs is provided on p. A-1.

INFORMATION PROCESSING INFORMATION CENTER

C. J. Shaw, Project Leader
H. J. Ilger
P. L. Eddy

Description

The Information Processing Information Center (IPIC) monitors, evaluates, and promotes the dissemination throughout SDC of outside technical knowledge relating to automatic information processing. With the cooperation of the SDC Technical Library, the IPIC collects, catalogs, displays, announces, files, distributes, references, searches, and reviews such things as technical reports, manufacturer's manuals, etc. It subscribes to several, regularly updated, looseleaf information services, covering such things as commercially available U. S. computers, display equipment, and data communication equipment, and the information processing literature. The IPIC also gives technical briefings and arranges technical briefings by outside experts.

The IPIC maintains a growing file of over 6,000 reports and several thousand miscellaneous (uncataloged) documents--sales brochures, computer manuals, etc. These are cataloged and filed under the following major topic headings:

- Information processing in general
- Logic, switching theory, automata and formal languages
- Automation, cybernetics, and information theory
- Digital computers
- Peripheral equipment
- Computer programming
- Nonnumerical information processing
- Mathematical and numerical computation
- On-line and real-time systems
- Applications of information processing

This literature may be borrowed or copied by Corporate personnel.

A major service provided by the IPIC is answering technical inquiries, which may range in scope from a request for comments to a brief essay on the state-of-the-art in natural language processing. Almost half of the inquiries come from outside the Research & Technology Division, and many come from outside the Corporation. The IPIC staff of experienced computer professionals is often requested to participate in various projects and studies throughout SDC.

Additional IPIC responsibilities include:

1. Monitoring, cataloging, and assisting in the exploitation of the Corporation's own inventory of technical resources by conducting ad hoc surveys of specific resource types (e.g., computing facilities), by assisting in proposal efforts, by answering specific inquiries, etc.

2. Answering requests for information and literature on JOVIAL and, in particular, the 7090/94 JOVIAL compiler for the Technology Directorate.

3. Serving as a focal point for the dissemination throughout the Corporation of IBM documents.

4. Performing as SHARE program librarian for the Corporation.

5. Providing individual, on-line guidance in using the facilities of selected time-sharing systems, particularly the SDC S/360 Time-Sharing System currently being developed (see p. 9-1).

6. Serving as a control center for running PL/I programs.

Progress

During 1966, the IPIC has, with the use of the SURF system (see p. 5-25), partly automated and recataloged its entire report collection. Over 600 new reports have also been cataloged. In addition, the IPIC has responded to more than 900 separate inquiries, including about 80 requests for JOVIAL compilers or literature. The IPIC also arranged or gave a small number of technical briefings during 1966.

IPIC staff members conducted and participated in a variety of projects during 1966, including a survey of JOVIAL usage, and studies of the PL/I programming language, computer programming and command and control, and the state-of-the-art and trends in computer programming languages. IPIC staff members also coordinated and designed the curricula for 360 programming courses for the Computer Center Department, and supported various SDC activities by reporting on SDC experience and capability in such areas as command and control computer programming, and programming language selection.

In August 1966, the IPIC took over from the Computer Center Department the functions of distributing IBM documents and maintaining the SHARE program library. The IPIC also initiated the practice of republishing, for SDC distribution, IBM's document announcements for the 360, 7090/94, and 1401 computers. The IPIC is currently ordering and distributing IBM documents at the rate of about 1,500 per month. Some time was spent learning the details of managing the SHARE program library. Several requests for SHARE programs are handled per month.

In anticipation of the installation of a tutorial teleterminal, initially tied to the SDC S/360 Time-Sharing System, the IPIC staff has been collecting and studying the system documentation and writing test programs to operate under the current version of the system.

During the summer, the IPIC took on the job of handling PL/I program decks and arranging for them to be run at the IBM Data Center. Currently, some 20 to 25 PL/I jobs are handled per month.

Plans

The IPIC plans to continue its current work of providing a central fund of knowledge about automatic information technology and rendering its present services. Much organizational and technical work remains to be done before the IPIC will be able to provide on-line guidance to prospective time-sharing users, especially if the service is to be extended to non-SDC time-sharing systems. In addition, the IPIC intends to more widely publicize the availability and contents of the SHARE program library.

Project Documentation

1. Shaw, C. J. Theory, practice, and trend in business programming. SDC document SP-2030/001/02. July 1, 1965. 18 pp.

2. Shaw, C. J. Computer programming and command and control. SDC document TM-2857. April 25, 1966. 62 pp.

3. Shaw, C. J. A graphic notation for describing multicomputer nets: Some preliminary notions. SDC document TM-3042. June 30, 1966. 21 pp.

4. Shaw, C. J. Computer software: Developments and recent trends in programming and operating systems. SDC document SP-2582. August 26, 1966. 12 pp.

5. Shaw, C. J. The gist of GIS (generalized information system). SDC document TM-3217. October 17, 1966. 12 pp.

6. Shaw, C. J. Selected SDC documents on applications and observations of time sharing. SDC document TM-1372/001/00. June 28, 1966. 21 pp.

INFORMATION CENTER ON INFORMATION SCIENCE AND TECHNOLOGY

Y. F. Neeland

Description

This Center was established in 1965 to help SDC personnel in maintaining an awareness of the state-of-the-art and of current research and developments in information retrieval, language processing, documentation, and related areas. The Center identifies and, where appropriate, acquires literature in these selected areas. It covers all types of publications (technical reports, printed articles, conference papers, books, dissertations, and a few patent disclosures). It also maintains a card catalog of published abstracts, which serves as a tool for retrieving information on authors, sources, and subject matter of the publications.

Progress

The following were the major activities in 1966:

1. Continuation of the identification and acquisition of literature, references, and abstracts. Approximately 2,000 published items were processed in 1966, and the number of catalog cards grew to 14,000.

2. Development of a series of bibliographies. The series of bibliographies on items processed by the Center, initiated in 1965, was continued. The last of a four-part bibliography of 1965 literature was published early in 1966. Three of the four parts of the bibliography of 1966 literature have been published in 1966, with the fourth part scheduled for release early in 1967. The bibliography serves SDC personnel, authors of chapters in the Annual Review of Information Science and Technology (see p. 11-1) and also other persons who can obtain copies from the Defense Documentation Center or from the Clearinghouse for Federal Scientific and Technical Information.

3. Continuation of answering inquiries and provision of information support, both to SDC personnel and to Annual Review authors.

Plans

In 1967 the Center will continue to improve and expand its coverage and provide a more efficient organization of its holdings. The Center will also explore the use of the SURF system (see p. 5-25) to provide better access to the literature.

Project Documentation

1. Neeland, Y. F. A bibliography on information science and technology for 1965, Part IV. SDC document TM-2625/004/00. January 15, 1966. 94 pp.

2. Neeland, Y. F. A bibliography on information science and technology for 1966, Part I. SDC document TM-3008/001/00. July 1, 1966. 51 pp.

3. Neeland, Y. F. A bibliography on information science and technology for 1966, Part II. SDC document TM-3008/002/00. September 15, 1966. 87 pp.

4. Neeland, Y. F. A bibliography on information science and technology for 1966, Part III. SDC document TM-3008/003/00. November 25, 1966. 94 pp.

PROGRAMMING DOCUMENTATION

M. H. Perstein, Project Leader
G. Campbell
R. P. Esada
J. S. Hopkins
P. R. Kennedy
H. S. Schwimmer

Description

This project aims to develop standards and guidelines for programming documentation. It is concerned with style and format of all phases of programming documentation; addresses questions of style as related to programming documents; and develops standard formats for programming documentation. The project is developing guidelines for the content of programming documents ranging from initial formulation and design to users' manuals and maintenance documents.

The programming documentation project also assists authors in applying the standards and guidelines and often takes an active role in producing programming documents.

Progress

Work has continued to refine standards for programming documentation both by publication of additional guides and revision of existing ones. The existing literature on documentation standards and guidelines has been studied in order to distill a relevant and coherent policy for SDC. The literature search included unpublished material obtained through contact with people involved in committee work for SHARE and for the United States of America Standards Institute.

Plans

Work will continue on the development and publication of standards and guidelines for programming documentation. Project members will continue to assist in the preparation of programming documents.

Project Documentation

1. Kennedy, P. R. Programming documentation standards - introduction. SDC document TM-3125. August 30, 1966. 14 pp.
2. Kennedy, P. R. Programming documentation standards - guideline for the user's guide. SDC document TM-3125/003/00. August 26, 1966. 15 pp.
3. Kennedy, P. R. Programming documentation standards - guidelines for the preliminary system description. SDC document TM-3125/001/00. November 1, 1966. 12 pp.
4. Kennedy, P. R. An unfinished chapter in the history of programming standards. SDC document SP-2645. November 3, 1966. 12 pp.
5. Perstein, M. H., and Herold, V. P. Style guide for computer program documentation for R&TD. SDC document PPG-6050/001/00. November 1, 1965. 31 pp.
6. Perstein, M. H. Numbered-line syntactic description of JOVIAL (J3). SDC document SP-2311. January 1, 1966. 10 pp.
7. Perstein, M. H. Condensed syntactic description of JOVIAL (J3). SDC document SP-2311/001/00. July 4, 1966. 4 pp.
8. Perstein, M. H. Grammar and lexicon for basic JOVIAL. SDC document TM-555/005/00. May 10, 1966. 102 pp.

RESEARCH & TECHNOLOGY DIVISION BOOKS

ANNUAL REVIEW OF INFORMATION SCIENCE AND TECHNOLOGY*

C. A. Cuadra, Editor
E. M. Wallace
Y. F. Neeland

This project involves the development of a series of Annual Reviews of work in the general area of information retrieval, language processing, and documentation. Each volume describes, relates, and critically analyzes the most significant journal and report literature for each calendar year. The potential audience includes instructors and students of information science and library science courses, managers and information specialists in information centers and libraries, and research workers in all areas related to information handling.

The Annual Review series is sponsored by the American Documentation Institute, under a grant from the National Science Foundation. SDC is providing the main technical support to the ADI and is sharing the cost of preparation of the first two volumes.

Volume I, published in September 1966, was formally introduced at the 29th Annual Meeting of the American Documentation Institute, where the reviews served as a primary basis for a series of state-of-the-art reviews. The contents and authors for Volume I are as follows:

Introduction to the ADI Annual Review
  Carlos A. Cuadra, SDC

Professional Aspects of Information Science and Technology
  Robert S. Taylor, Lehigh University

Information Needs and Uses in Science and Technology
  Herbert Menzel, N. Y. Univ.

Content Analysis, Specification and Control
  Phyllis Baxendale, IBM Research Laboratory

File Organization and Search Techniques
  W. Douglas Climenson, U. S. Government

Automated Language Processing
  Robert F. Simmons, SDC

Evaluation of Indexing Systems
  Charles P. Bourne, Programming Services, Inc.

New Hardware Developments
  Annual Review Staff

Man-Machine Communication
  Ruth M. Davis, Department of Defense

Information System Applications
  Jordan J. Baruch, General Electric Co.

Library Automation
  Donald V. Black, Univ. of Calif.
  Earl A. Farley, Univ. of Kansas

Information Centers and Services
  G. S. Simpson, Jr., Battelle Memorial Institute
  Carolyn Flanagan, Engineering Index

National Information Issues and Trends
  John Sherrod, U. S. Atomic Energy Commission

Indexes
  Pauline Atherton and Stella Keenan, American Institute of Physics

On the basis of an analysis of the problems encountered in Volume I, an improved "Authors Guide and Style Manual" was prepared, in 1966, for use by the authors of subsequent chapters.

During 1965 and early 1966 the Volume I authors were provided with extensive selected bibliographies, as an aid in their literature collection and analysis (see p. 10-3). Feedback on the use of these bibliographies was used to improve the format for the 1966 bibliographies.

During the spring of 1966, the contents of Volume II were decided upon and qualified authors were found. The work for Volume II will be completed by June, 1967. Because of improvements based on experience with Volume I, the publication cycle for Volume II will be one month shorter. Publication is set for September 1, 1967.

*Supported in part by the American Documentation Institute, under a grant from the National Science Foundation.

AUTOMATED LANGUAGE PROCESSING

H. Borko, Editor

This book is a natural outgrowth of the efforts by SDC and other personnel concerned with linguistics and language processing to keep informed in their fields and to communicate their research findings. The book will review, integrate, and draw conclusions from the available research in language data processing by computer. The potential audience is the investigators presently working in such fields as information retrieval, machine translation, mechanolinguistics, stylistics, and content analysis. It is expected that the book will also be used to orient newcomers to these fields and to serve as a text in university courses on computer applications.

The preparation of the book involves a number of unique features. The manuscript was typed on a flexowriter and the paper tapes were converted to magnetic tapes. These tapes were used as input to a photocomposition device for final typesetting. They were also used for computer processing to produce an index to the book. Unfortunately, these innovative features caused a considerable delay in the production of the book, but these obstacles have now been overcome and the book should be available early in 1967.

The following is a list of the chapter headings:

Chapter	Author
Language Data Processing	
1. Introduction	H. Borko
2. Language and the Computer	L. Schultz
3. Mathematical Models in Linguistics and Language Processing	H. P. Edmundson
Statistical Analysis	
4. Indexing and Classification	H. Borko
5. Extracting and Abstracting	R. E. Wyllys
6. Stylistic Analysis	S. Y. Sedelow
Syntactic Analysis	
7. Syntactic Theories in Computer Implementation	D. G. Bobrow
8. Answering English Questions by Computer	R. F. Simmons
9. Translating Languages	E. D. Pendergraft
10. Designing Artificial Languages for Information Storage and Retrieval	C. H. Kellogg
An Experiment in Language Processing	
11. Book Indexing and Preparation by Computer	H. Borko

HANDBOOK OF NONPARAMETRIC STATISTICS

J. E. Walsh, Author

This book represents a developmental research project, with the original intent of containing, in a reasonably complete and usable fashion, statements of nearly all experimental types of nonparametric procedures based on material occurring prior to 1958.

The first volume, subtitled Investigation of Randomness, Moments, Percentiles, and Distributions, consists of 575 pages and was published by D. Van Nostrand Co., Inc., in August 1962. The second volume, subtitled Results for Two and Several Sample Problems, Symmetry, and Extremes, consists of 712 pages and was published by Van Nostrand in February 1965. The third volume, subtitled Analysis of Variance, will contain approximately 550 pages and the first draft has been completed. It is scheduled to be published by Van Nostrand in the spring of 1968.

TABLE OF RANDOM NUMBERS

C. E. Clark, Author

A table of uniform and normally distributed random numbers has been prepared. This table contains a novel feature not found in currently available tables of random numbers, namely, the tabulation of certain statistics of subsets of the numbers. The usefulness of these statistics lies in increasing the sampling efficiencies of a Monte Carlo procedure. The book, entitled

Random Numbers in Uniform and Normal Distribution, was published in the fall of 1966 by the Chandler Publishing Company.

MAN-MACHINE DIGITAL SYSTEMS

H. Sackman, Author
(part-time in Research & Technology Division)

This book is concerned with the historical origins, status and prospects of computer-assisted experimental methodology for design and development of computer-supported systems. Detailed descriptions of such systems and the impact of new experimental techniques are described for the advent of computer-serviced societies, with special emphasis on a systems approach. A humanistic philosophy is developed toward democratic and experimentally regulated evolution of such societies. The book is currently in production and will be published by Wiley & Sons in the spring of 1967.

Chapter headings are as follows:

PART I:	THE EMERGENCE AND NATURE OF MAN-MACHINE DIGITAL SYSTEMS
1	Human Evolution Toward Computer-Serviced Societies
2	Introduction to Man-Machine Digital Systems
3	Evolution of SAGE as a Prototype Man-Machine Digital System
4	Division of Labor Between Men and the Computer in SAGE
PART II:	EXPERIMENTAL METHOD IN MAN-MACHINE DIGITAL SYSTEMS
5	System Development and System Testing
6	Real Time
7	Experimental Investigation of Realtime System Performance
8	Simulation and Training
9	Man-Computer Dialogue
10	Collection and Analysis of Digital Data
PART III:	THEORY, PHILOSOPHY AND SOCIAL PROSPECTS OF MAN-MACHINE DIGITAL SYSTEMS
11	Theory and Philosophy of Man-Machine Digital Systems
12	Computers and Society

COMPUTER APPLICATIONS IN THE BEHAVIORAL SCIENCES

H. Borko, Editor

This book, published by Prentice-Hall, Inc., in February 1962, provides authoritative information on the role of computers in behavioral science research.

MANPOWER DEVELOPMENT

The System Training Concept: A New Management Tool

E. H. Porter, Author

This book, published by Harper and Row in April 1964, describes and interprets the story of the RAND/SDC System Training Program, and indicates how system training principles may be applied to a wide variety of organizations.

INVENTORY OF DEMONSTRABLE R&TD PROGRAMS

During 1966, over 500 briefings and demonstrations on the programs developed in R&TD were provided to several thousand visitors to SDC by Division personnel. Below are brief descriptions of the more frequently demonstrated R&TD programs. (This list does not include the on-line programs produced and demonstrated by other organizations in SDC.) Page references indicate where in the Report more detailed information can be found on the particular program.

ADROIT (page 2-11)
A computer-programmed organism which responds to commands in accord with theories of animal behavior. The robot's actions are visualized on a display scope.

Automated Counseling (page 7-11)
An experimental, interactive, computer-based system that provides automatic appraisal of students' cumulative folders and conducts on-line educational-planning interviews.

Bargaining and Negotiation Behavior (page 6-7)
A computer-based laboratory simulation vehicle for studying the dynamic interaction process that occurs in mixed-motive bargaining situations.

BOLD: Bibliographic On-Line Display (page 5-20)
A highly automated document storage and retrieval system enabling nonprogrammer users to conduct on-line searches of large volumes of data for selective retrieval and display of information.

DGAQ: Datebook Generator and Querier
Allows a nonprogrammer to interact with a computer in establishing times and schedules for meetings.

GOCI
A tree-structured language facilitating person-to-person and person-to-computer communication via display scopes.

GPDS: General Purpose Display System (page 4-5)
A system providing an on-line capability to design, generate, manipulate, save and recall display formats.

Leviathan (page 6-12)
A computer-based laboratory simulation vehicle for studying large organizations. Simulates essential features of bureaucratic organizations and their working forces; live subjects enact the roles of executives in a pyramid of command.

LISP (page 1-8)
An advanced list processing language and system designed primarily for symbol manipulation.

On-Line Character Recognition (page 1-18)
A program that recognizes hand-written characters in an on-line, real-time environment and will enable information to be presented to the computer by simply writing the inputs on a tablet.

PLANIT (page 7-8)
A general-purpose, on-line lesson design and teaching program with built-in interactive algebraic and statistical capabilities.

Protosynthex (page 5-7)
An experimental natural-language-processing program for locating, in computer-stored text, answers to questions posed in English.

STAT (page 7-5)
An interactive program that provides the laboratory portion of a first course in applied statistics. Presents statistical exercises, generates pseudorandom samples, provides computational assistance, and evaluates students' answers.

A DEMONSTRATION OF TSS/LUCID

TABS: Time-sharing Automated Budgeting System
Enables a budget analyst to input raw data for a budget into the computer and have the computer output the budget cost on a month-by-month basis.

Time-Sharing Systems (pp. 3-3 and 9-3)
Permit simultaneous use of a single computer by multiple users. Both remote and local users, working concurrently, can produce and debug new programs, conduct laboratory experimentation, retrieve information from data bases, and perform many other functions. Operational on the Q-32 and the IBM S/360.

TINT: Teletype INTerpreter
An on-line time-sharing program for programmers that facilitates construction, compilation, debugging, and operation of JOVIAL-like programs.

TRACE: Time-shared Routines for Analysis, Classification and Evaluation (page 6-16)
An interactive, time-shared, display-oriented programming package designed to assist in the prestatistical and statistical analyses of large volumes of data, such as experimental and census data.

TSS/LUCID (page 4-7)
An interactive data base system that enables the nonprogrammer to use the computer to perform complex data management functions (e.g., data description, data base construction, on-line query).

TSS Manager Program (page 3-6)
A set of display programs that enables operators of the Q-32 Time-Sharing System to monitor the utilization of the system and to manage the allocation of time and space.

SYMPOSIA

The following symposia were hosted by the Research & Technology Division during 1966:

REGIONAL REVIEW CONFERENCE OF SOUTHWEST REGIONAL LABORATORY

SDC organized a Regional Review Conference of the Southwest Regional Laboratory for Educational Research and Development, held in El Segundo, California, on May 3, 1966. Ninety-one professional and lay persons representing the various areas of the Southwest Region participated in the conference, which was designed to review and improve the initial statement of Laboratory program requirements and specifications. SDC's Harry Silberman, Development Program Director, reviewed the history of the Laboratory's development, described the region's communication network, and outlined the procedures by which the initial Laboratory program requirements were specified. Participants indicated their reactions to the program plans and suggested possible specific improvements for the first year of Laboratory operations. John Coulson of SDC chaired one of the discussions.

REGIONAL LABORATORY CONFERENCE ON INFORMATION PROCESSING

On July 6 and 7, 1966, SDC and the Southwest Regional Laboratory for Educational Research and Development hosted a national conference of all Regional Laboratory project directors and accompanying staff members to discuss common information processing problems. The presentations provided a general orientation on the present state-of-the-art. Discussions were directed at the development of cooperation among Regional Laboratories in solving data processing problems.

Presentations by representatives of the U. S. Office of Education on the Education Research Information Center and on the National Center for Educational Statistics, along with summary reports from representatives of various regional laboratories, occupied the morning of the first day of the conference. In the afternoon, project directors and data processing specialists held separate group sessions. On the second day, Donald Drukey, SDC, spoke on time-sharing and data-base systems. James Miller, University of Michigan, spoke on multimedia networks; and William Ramage, University of Pittsburgh, made a presentation on man-machine interface problems. In the final session, Wade Robinson, Central Midwestern Regional Laboratory, and John Coulson, SDC, summarized the group sessions held on the previous day; and Dick Schutz, Southwest Regional Laboratory, summarized the conference and solicited recommendations for further cooperative activities from the Regional Laboratory directors.

One outcome of the conference was the establishment of a committee on educational data processing for Regional Laboratories. Harry Silberman, SDC, serves as chairman of that committee which has subsequently met in Denver, Colorado.

ONR-CAI INTEREST GROUP

The third meeting of the Office of Naval Research Computer-Assisted Instruction Interest Group was hosted at SDC on September 26-27, 1966. The attendees (approximately 80) represented a cross-section of the major users of CAI in the leading universities and private businesses across the country. The group meets semiannually to discuss problems and to exchange information with respect to the latest state-of-the-art in the area of computers used in education.

The first day of the two-day meeting was devoted to talks and demonstrations by the Education and Training staff, on the subjects of

TRANSNATIONAL WORKING GROUP ON DYNAMICS OF CONFLICT

computer-assisted instruction at SDC, and the use of computers in the field of education.

The second day, the ONR members divided into three groups to discuss the following problems: (1) What procedures should be used to develop quantities of quality instructional materials for CAI? (2) How should CAI be integrated with existing education functions? (3) What procedures should be used to facilitate acceptance of CAI in schools? (4) How will problems of CAI cost be solved?

Among SDC participants were: Harry Silberman, meeting chairman; Sam Feingold, program director; Jack Bratten, John Cogswell, Don Estavan, Ted Krebs, and Jules Schwartz, featured speakers.

The proceedings of the meeting are being prepared and will be published during the first quarter of 1967. The next ONR meeting will be held at the University of Pittsburgh where work is being done on continuous progress instruction.

DYNAMICS OF CONFLICT CONFERENCE

SDC played host to a distinguished group of American and European social psychologists on November 17-23 in Santa Monica when it cosponsored the Third Transnational Working Group on Dynamics of Conflict.

The 13-member working group includes Gerald Shure, head of SDC's Behavioral Gaming and Simulation Program, and colleagues from 11 U. S. and European educational institutions. Group members are specialists in applying the techniques of experimental social psychology to problems in intergroup and interpersonal conflict, conflict resolution, and negotiation and bargaining.

Participants took advantage of computer facilities at SDC--particularly the TRACE system (p. 6-16)--and met social scientists from SDC, UCLA and RAND who are expert in the study of the dynamics of conflict.

Major attention at the meeting focused on reviewing and synthesizing completed studies, planning further research and discussing a number of general topics. Special emphasis was devoted to a discussion of the relevance of laboratory studies to problems in bargaining and negotiation, particularly in the arena of international conflict.

A special highlight of the conference centered around the discussion and evaluation of a 12-man, noncomputerized bargaining game that was carried out in nine European and American centers. The game, for which Shure and Harold H. Kelley of UCLA assumed major responsibility, afforded opportunity not only for the collection of hard bargaining data, but also for recording and observing free verbal interaction within or between bargaining parties.

ACM SIGPLAN MEETING

Over 40 senior information processing specialists attended an open meeting of an ACM SIGPLAN working group at SDC on October 5, 1966. The group--known officially as "SIGPLAN Working Group Number 1 on Syntax-Directed Compilers"--heard two talks on metacompiling and saw demonstrations of two SDC-developed metacompilers.

Val Schorre, SDC, presented a tutorial review of metacompiling techniques, followed by Lee Schmidt from Beckman Instruments, whose talk "Inside the Big C" dealt with the same subject.

SIGPLAN members also saw demonstrations of META and META5, programs that are used to compile compilers (see p. 1-3). The programs were run on the Q-32 computer under the SDC Time-Sharing System.

SDC's Erwin Book was cohost of the meeting.

RESEARCH COLLOQUIA

During 1966 the following colloquia were held under the sponsorship of the Research & Technology Division.

DATE	SPEAKER	TOPIC
2/9	S. Ginsburg, SDC	Stack Automata and Compiling
3/1	J. G. Miller, Mental Health Research Institute, University of Michigan	EDUCOM (the Inter-University Communication Council)
3/16	Y. Bar-Hillel, The Hebrew University of Jerusalem	Universal Semantics and Logic
4/12	M. Juncosa, The RAND Corporation	Recent Activities in Numerical Analysis
4/20	R. E. Bleier, SDC	Time-Shared Data Management System
5/4	R. P. Abelson, Yale University	The Ideology Machine
5/23	J. Staudhammer, Arizona State University	Teaching Applied Mathematics with Computer Assistance
6/15	R. J. Newman, SDC	Computer and Display Aids to Human Problem Solving
6/22	L. Friedman, SDC	Computer Synthesis of Behavior
6/29	T. Saaty, U. S. Arms Control & Disarmament Agency, Department of State	Basic Techniques in Arms Control Research
7/14	T. Dalenius, Stockholm University	Intuition vs. Analysis
7/15	C. H. Kellogg, SDC	Improving User Communication With Structured Data Files
7/19	H. Solomon, Stanford University	Cluster Analysis
8/10	E. R. Clark, SDC	On the Automatic Simplification of Source-Language Programs
8/17	J. T. Gullahorn & J. E. Gullahorn, Michigan State University	Some Problems in the Empirical Validation of a Computer Model of Social Behavior
10/13	J. Dyer, SDC	Generalized Multistep Methods for the Numerical Solution of Differential Equations
10/18	E. Parzen, Stanford University	Analysis and Synthesis of Models by Time Series Methods
11/2	P. W. Abrahams, Information International, Inc.	The LISP 2 Programming Language and System
11/11	W. M. Newman, Imperial College, London	Computer-Aided Architectural Design
11/16	K. Needham, Cambridge Language Research Unit, and Newnham College, Cambridge, England	Semantic Analysis

PROFESSIONAL ACTIVITIES

The following is a listing of the professional activities of the staff for 1966. Included are presentations of papers, appointments to offices, and any publications not directly related to the projects in this report.

Abrahams, P. W. (III), Barnett, J. A., Book, E., Firth, D. C., Kameny, S. L., Weissman, C., et al.
The LISP 2 programming language and system. Presented at the AFIPS Fall Joint Computer Conference, San Francisco, November 8-10. (Available as SDC document TM-3163.)

Ancker, C. J., Jr.
Some further results in the theory of stochastic duels. Invited presentation at the 29th National Meeting of the Operations Society of America, Santa Monica, May 19-20. Also Chairman, Military Applications Section V: General Developments.

The status of developments in stochastic duels--II. Presented at the 30th National Meeting of the Operations Research Society of America, Durham, North Carolina, October 17-19. Also Chairman, Military Applications Section I: Mathematical Theories of Combat--II: Micro Versus Macro Representations.

Chairman, Session VA, 5th U. S. Army Operations Research Symposium, Ft. Monmouth, New Jersey, March 29-30.

Vice-Chairman, Visiting Committee for Industrial Engineering, University of Southern California.

Member, Planning Committee, Operations Research Lecture Series, University of California, Los Angeles.

Vice-Chairman and Chairman-Elect, Military Applications Section, Operations Research Society of America; Member, Membership Committee; Chairman, Professional Liaison Committee of the Military Applications Section; Screener, Lanchester Prize.

Reviewer, Operations Research and Opsearch. Abstractor, International Abstracts in Operations Research.

Barnett, J. A.
LISP structure and software memory associations. Presented at the Semiannual SHARE International Meeting, Toronto, Canada, August 10.

LISP 2 programming examples. Presented to the Special Interest Group on Artificial Intelligence of the ACM, Santa Monica, September 20.

Vice-Chairman, Special Interest Group on Artificial Intelligence of the ACM.

See also Abrahams, P. W.

Barrar, R. B.
Convergence of the Poincaré-von Zeipel procedure in celestial mechanics. Presented at Celestial Mechanics Seminar, Yale University, New Haven, January 27.

Reviewer, Mathematical Reviews.

Bernstein, M. I.
Manager, SHARE Graphic Hardware Project.

Blankenship, D. A.
See Doyle, L. B.

Book, E.
Chairman, Special Interest Group on Programming Languages of the ACM.

See also Abrahams, P. W.

Borko, H.
President, American Documentation Institute. General Chairman, 29th Annual Meeting of the ADI, Santa Monica, October 3-7.

Automatic information and question answering. Presented at the 50th Annual Meeting of the American Educational Research Association, Chicago, February 17-19.

The conceptual foundations of information systems. Presented at the Technical Publications Seminar, sponsored by the Society of Technical Writers and Publishers and the Department of Communication, Stanford University, Palo Alto, March 19.

On information system analysis and design. Presented at the 3rd Technical Information Center Administration Conference (TICA) Philadelphia, August 29-30.

Research in automated classification. Presented at the American Documentation Institute Tutorial Session, New York, February 23.

Chairman, Panel on Future Computer Applications and Implications, Institute on Computers and the Policy Making Community, sponsored by the Carnegie Endowment for International Peace and the United States Atomic Energy Commission, Lawrence Radiation Laboratories, Berkeley, California, April 18.

Invited Participant, International Symposium on Syntactic-Relational Factors in Classification, sponsored by the University of Maryland and the National Science Foundation, Washington, D. C., June 8.

Borko, H. (cont.)
Visiting Professor, Linguistic Institute of the Linguistic Society of America, University of California, Los Angeles, June 17-August 12; Visiting Professor, University of California, Los Angeles, School of Library Science.

Member of official United States delegation to the 32nd Congress of the International Federation for Documentation, The Hague, Netherlands, September 19-24.

Invited Participant in Symposium on Mechanized Abstracting and Indexing, sponsored by UNESCO and VINITI, Moscow, USSR, September 25-October 1.

Bratten, J. E.
System development and innovation in education. In S. C. Eboch (Ed.), Novel strategies and tactics for field studies of new educational media demonstrations. The Ohio State University Research Foundation RF Project 2001.

What's new in educational research at SDC. Presented to the Community College Workshop at Chabot College, San Leandro, California, November 17.

Invited Participant, Conference to Develop a Rationale for the Division of Plans and Supplementary Centers, sponsored by the U. S. Department of Health, Education, and Welfare, Washington, D. C., May 16-17.

Consultant, Bureau of Elementary and Secondary Education, Office of Education, Washington, D. C.

Burnaugh, H. P.
The BOLD (Bibliographic On-Line Display) system. Presented at the 3rd Annual National Colloquium on Information Retrieval, University of Pennsylvania, Philadelphia, May 12-13. (Available as SDC document SP-2338/000/01.)

Chaney, P. E.
See Linde, R. R.

Clark, E. R.
On the automatic simplification of source-language programs. Presented at the 21st National Conference of the ACM, Los Angeles, August 30-September 1. (Available as SDC document SP-2389.)

Cogswell, J. F.
Computers in student appraisal and educational planning; Instructional management information systems; and Systems technology in education. In J. Loughary (Ed.) Man machine systems in education. New York: Harper & Row.

Cogswell, J. F., Donahoe, C. P., Estavan, D. P. and Rosenquist, B. A.
The design of a man-machine counseling system. Presented at the Annual Convention of the American Psychological Association, New York City, September 2-6. Also presented at the Western Regional Conference of California Project TALENT, San Francisco, November 15-16, and to the Los Angeles Psychological Counseling Association, December 7. (Available as SDC document SP-2576/001/01.)

Cogswell, J. F. and Estavan, D. P.
Explorations in computer-assisted counseling. Presented at a conference of Project CAUSE, San Diego State College, January 6. Also presented at the Center for Vocational and Technical Education Seminar at Ohio State University, Columbus, January 12-14, and to the Superintendent of Schools, County of Los Angeles, June 21. (Available as SDC document TM-2582.)

Coulson, J. E.
Automation, electronic computers, and education. Phi Delta Kappan, March 1966, XLVII (7), 340-344.

Computers in programmed instruction and educational data processing. Invited lecture presented at Santa Monica City College, December 13.

Computers in research and development on automated instruction. Presented to the 4th International Congress of Cybernetic Medicine, Nice, France, September 19-23. Also Panel Chairman.

Current development in instructional technology. Presented at the Annual Convention of the Greater Victoria Teachers' Association, Victoria, B. C., March 4-5.

Educational applications of information-processing technology. Educational Technology, October 1965, VI (19), 1-10.

An inductive approach to instructional research. Presented to a seminar at Los Angeles State College, School of Education, October 27.

Programmed instruction and computer-aided instruction in mathematics: A progress report of research at SDC. Presented at the Fall Conference of the California Mathematics Council, Northern Section, Asilomar, California, December 3.

Session Chairman, Regional Review Conference, Southwest Regional Laboratory for Educational Research and Development, El Segundo, May 3.

Panelist, Computers in Educational Research Panel, California Educational Data Processing Association Meeting, San Francisco, November 15.

Coulson, J. E. (cont.)
Chairman, Division 15, Committee on Programmed Instruction, American Psychological Association.

Consultant, Air Training Command (Programmed Instruction Task Force) and Office of Economic Opportunity.

Member, Industry-Communications Advisory Committee, Title III Program, Los Angeles City Schools; Education Committee, Santa Monica Chamber of Commerce.

Cramer, P. A.
Manager, Systems Division, SHARE.

Cuadra, C. A.
Automation and fingerprint retrieval. Presented to the Los Angeles Chapter, Institute of Electrical and Electronics Engineers, Santa Monica, March 21. (Available as SDC document SP-2347.)

Program Chairman, 29th Annual Meeting of the American Documentation Institute, Santa Monica, October 3-7. Chairman, Panel on Whither Information Sciences?

Dear, R. E.
Referee, Journal of Mathematical Psychology.

Dobbs, G. H.
Computer time-sharing. Presented to the Los Angeles Chapter, Financial Executives' Institute, January 20.

Industry/science education and industry/community relations. Presented at Community Science Workshops, Inc., Los Angeles Trade-Technical College, April 16.

Man-machine interaction--instant access to information. Presented at the 1966 International Data Processing Conference and Business Exposition, Chicago, June 21-24.

Research and development problems in time-sharing. Presented at the CDC 1604 Users' Group Conference, Los Angeles, May 5. Also presented as the Luncheon Address at the 21st National Conference of the ACM, Los Angeles, August 30-September 1.

SDC progress and plans for IBM 360 time-sharing. Presented to the Special Interest Group on Time-sharing of the ACM, Santa Monica, March 14.

State of the art survey of data base systems and time-sharing systems. Presented to the California Commission for the County Tomorrow, Anaheim, January 27.

Invited Discussant, Seminar on Applications of Time-sharing Technology in Industry, sponsored by the Diebold Research Group, Los Angeles, March 2.

Invited Participant, Panel on Impact of Real-time Use of Computers on Management, AFIPS Fall Joint Computer Conference, San Francisco, November 8-10.

Donahoe, C. P.
See Cogswell, J. F.

Doyle, L. B. and Blankenship, D. A.
Technical advances in automatic classification. Prize-winning paper presented at the 29th Annual Meeting of the ADI, Santa Monica, October 3-7.

Drukey, D. L.
A general-purpose display system for command and control. Presented at the 11th Technical Symposium of the Avionics Panel of NATO AGARD, Munich, Germany, November 7-10.

Language processing and retrieval research at SDC. Presented at the 29th Annual Meeting of the ADI, Santa Monica, October 3-7.

Time-sharing applications for management decision making. Presented at the 12th Annual Electronic Data Processing Conference, sponsored by the American Management Association, New York City, February 28-March 2.

Du Bois, E. M.
Development of the SDC system 360 time-sharing system. Presented at Conference with a 50, sponsored by the Applied Mathematics Division of Argonne National Laboratories, University of Chicago, Argonne, Illinois, October 31-November 1. (Available as SDC document SP-2691.)

Eddy, P. L.
Consultant, Programming Fundamentals for 9th Grade Mathematics Students Project, Lincoln Junior High School, Santa Monica.

Estavan, D. P.
A CAI language for a time-sharing system. Presented at the Office of Naval Research Conference on CAI Languages, Cambridge, Massachusetts, March 2-3.

See also Cogswell, J. F.

Feingold, S. L.
A flexible interactive CAI language. Presented at the Office of Naval Research Conference on CAI Languages, Cambridge, Massachusetts, March 2-3.

Participant in the Task Force on Educational Systems and Technology at the University of Colorado, Boulder, August 16-17.

Feingold, S. L. (cont.)
Program Coordinator, CAI Group Interest Conference, sponsored by the Office of Naval Research, Santa Monica, September 26.

Feingold, S. L. and Frye, C. H.
PLANIT (Programming LANguage for Interactive Teaching). Presented at the National Conference on Curricular and Instructional Innovations in Large Colleges and Universities, Michigan State University, East Lansing, November 8. Also presented to the San Fernando Valley Chapter of the ACM, November 16.

Firth, D. C.
See Abrahams, P. W.

Fleishman, T.
The development of estimating equations for computer programming costs. Presented at the 30th National Meeting of the Operations Research Society of America, Durham, North Carolina, October 17-19.

Franks, E. W.
A time-shared data management system using a cross-index file and self-defining entries. Presented at the AFIPS Spring Joint Computer Conference, Boston, April 26-28. (Available as SDC document SP-2248.)

Friedman, L.
Computer synthesis of behavior. Presented at the Brain Research Center, University of Rochester, New York, August 8.

A programming strategy for robot control. Presented to the Artificial Intelligence Group, Stanford Research Institute, August 29-30.

A theory of instinctive behavioral mechanisms. Presented at the Annual Meeting of the American Association for the Advancement of Science, Washington, D. C., December 26-31.

Invited participant, Computers and Psychobiology Workshop, sponsored by ONR Physiological Psychology Branch, U. S. Naval Postgraduate School, Monterey, California, May 16-17.

Frye, C. H.
Participant, CAI Group Interest Conference, sponsored by the Office of Naval Research, Santa Monica, September 26.

See also Feingold, S. L.

Gafarian, A. V.
Digital computer simulation at diamond interchanges. Presented at the Bureau of Public Roads Program Review Meeting, Washington, D. C., December 6-8.

Chairman, Institute of Mathematical Statistics, Regional Section Meeting of the American Statistical Association, Los Angeles, August 15-19.

Member, Editorial Advisory Board, Transportation Research. Reviewer, International Abstracts in Operations Research and Transportation Research. Referee, American Statistical Association, Operations Research Society of America, and Highway Research Board.

Gafarian, A. V., Hayes, E. and Mosher, W. W.
The development and validation of a digital simulation model for design of freeway diamond interchanges. Presented at the 29th National Meeting of the Operations Research Society of America, Santa Monica, May 19-20.

Gafarian, A. V. and Walsh, J. E.
Statistical approach for validation of simulation models by comparison with operational systems--illustrated for traffic flow. Presented at the 4th Conference of the International Federation of Operational Research Societies, Cambridge, Massachusetts, August 29-September 2.

Ginsburg, S.
Automata, phrase-structure languages, and logic. Presented at the University of Southern California, Los Angeles, March 30.

Context free languages. A series of five talks presented at the Conference on the Algebraic Theory of Machines, Languages, and Semigroups, Pacific Grove, California, August 29-September 8.

The mathematical theory of context free languages. New York, N. Y.: McGraw Hill.

Stack automata and compiling. Presented at the Case Institute of Technology, Cleveland, Ohio, April 12.

Ginsburg, S., Greibach, S. A. and Harrison, M.
One-way stack automata. Presented at the 7th Annual Symposium on Switching Circuit Theory and Logical Design, Berkeley, California, October 26-28.

Grant, E. E. and Sackman, H.
An exploratory investigation of programmer performance under on-line and off-line conditions. Presented to the Special Interest Group on Time-sharing of the ACM, Santa Monica, September 19. (Available as SDC document SP-2581.)

Haggerty, D. P. and Oppenheim, D. K.
META5: A tool to manipulate strings of data. Presented at the 21st National Conference of the ACM, Los Angeles, August 30-September 1.

Hanna, W. J.
Vice-Chairman, San Fernando Valley Chapter of the ACM.

Harman, H. H.
The minres method of factor analysis. Presented at the Annual Convention of the American Psychological Association, New York City, September 2-6.

Session Chairman, Workshop on Formulating the Research Attack, National Institutes of Health Conference, Santa Monica, April 17-18. Member, Planning Committee on Techniques of Accident Research, NIH. Consultant, Accident Prevention Research Study Section, NIH.

Reviewer, Computing Reviews. Screener, Lanchester Prize, Operations Research Society of America. Editorial Consultant, Journal of Mathematical Psychology and Journal of the Human Factors Society.

Hayes, E.
Programming a digital simulation model of a freeway diamond interchange. Presented at the 21st National Conference of the ACM, Los Angeles, August 30-September 1.

Invited Participant, Workshop on Simulation Languages, University of Pennsylvania, Philadelphia, May 17-18.

See also Gafarian, A. V.

Hibbard, T. N.
Scan limited automata and context limited grammars. Presented at the Algebraic Languages and Machines and Semigroups Conference, Pacific Grove, California, August 29-September 8.

Referee, Journal of the ACM.

Hormann, A. M.
Problem-solving and learning by man-machine teams. Presented to the Naval Intelligence Research Advisory Group, Office of Naval Research, Washington, D. C., June 21.

President, SDC Branch, Scientific Research Society of America (RESA).

Irvine, C. A.
Program Chairman, Los Angeles Chapter of the ACM.

Jacobs, E. H.
Moderator, Panel on PL/I programming language for command and control systems. San Fernando Valley Chapter of the ACM, January 19.

Chairman, Awards Committee, Los Angeles Chapter of the ACM.

Kameny, S. L.
Chairman, Communications Sessions, Special Interest Committee on Symbolic and Algebraic Manipulation, Washington, D. C., March 29-31.

Invited Participant and Panelist, IFIP TC-2 Working Conference on Symbol Manipulation Languages, Pisa, Italy, September 4-9.

See also Abrahams, P. W.

Karush, W.
Screener, Lanchester Prize, Operations Research Society of America. Referee, Operations Research and Management Science. Reviewer, International Abstracts in Operations Research.

Katter, R. V.
Validating a test of document representations. Prize-winning paper presented at the 29th Annual Meeting of the ADI, Santa Monica, October 3-7.

Visiting Professor, Linguistic Institute of the Linguistic Society of America, University of California, Los Angeles, June 17-August 12.

See also Weis, R.

Kellogg, C. H.
Reviewer, Computing Reviews.

Krebs, L. T. and Yett, F. A.
Brief description of SPLAN (School Organization Budget Planning System). Presented at the Annual Review of Regional Laboratory Activities, U. S. Office of Education, Washington, October 18-19. Also presented at the California Project TALENT Conference, San Francisco, November 15-16.

LaBolle, V.
Development of aids for the management of computer programming. Presented at the Spaceborne Computer Software Workshop, El Segundo, California, September 20-22.

Statistical analysis of computer programming costs. Presented at the 4th Annual Computer Personnel Research Conference, University of California, Los Angeles, June 27.

Assistant Chairman, Special Interest Group on Programming Management of the ACM; Secretary, San Fernando Valley Chapter of the ACM.

Linde, R. R. and Chaney, P. E.
Operational management of time-sharing systems. Presented at the 21st National Conference of the ACM, Los Angeles, August 30-September 1. (Available as SDC document SP-2392.)

Meeker, R. J.
The pacifist game. Presented at a seminar on the Anatomy of Peace and Conflict, Monterey Peninsula College, Monterey, California, March 7.

TRACE: Lecture and demonstration. Presented to the American Statistical Association, Los Angeles, August 13.

User interaction with the TRACE system. Presented at the TRACE Symposium, sponsored by APA Divisions 5 and 21, American Psychological Association Meeting, New York City, September 3.

See also Shure, G. H.

Meyer, W. E.
Member, SHARE Symbolic Mathematical Computations Committee.

Myer, E. P.
A display story. Presented to the Department of Public Health, University of California, Los Angeles, May 9.

SDC Technology Directorate activities. Presented at the NASA Computational Center, Huntsville, Alabama, July 27. Also presented at Manned Spacecraft Center, NASA, Houston, July 28.

Needham, K. (Sparke-Jones)
Invited Participant, Panel on Automated Language Processing, Annual Meeting of the ADI, Santa Monica, October 3-7.

Newman, J. R.
Extension of human capability through information and display systems. Presented at the Annual Convention of the American Psychological Association, New York City, September 2-6.

Newmark, G.
New approaches to improving foreign language instruction. Presented at the U. S. Air Force in Europe (USAFE) Educational Conference, Toul, France, April 10-13.

Evaluation techniques and instruments for use in foreign language programs. Presented to the California Association for Supervision and Curriculum Development, Foreign Language Special Interest Committee, Riverside, May 6. Also presented to the San Diego County Foreign Language Coordinating Council, San Diego, May 23.

New approaches to improving foreign language instruction. Presented to the Foreign Language Conference, sponsored by the Far West Laboratory for Educational Research and Development, Berkeley, California, December 1.

Programmed learning. Presented to the National Defense Education Act Summer Institute, University of California, Los Angeles, July 6.

A study of programmed self-instruction for seventh-grade learners. In G. Mathieu (Ed.), Advances in the teaching of modern languages. London: Pergamon Press, Inc., II (3), 47-75.

Techniques for evaluating foreign language instruction. Presented to the Foreign Language Association of Santa Clara County at Palo Alto, California, January 22. Also presented to the Conference of Alameda County Schools at San Leandro, January 22.

Olney, J. C.
English discourse structure. Presented at the Linguistic Colloquium, University of California, Los Angeles, March 29.

An inventory of syntactic features and word classes proposed in formal grammars of English. Presented to the National Meeting of the Association of Machine Translation and Computational Linguistics, University of California, Los Angeles, August 1.

Oppenheim, D. K.
See Haggerty, D. P.

Perstein, M. H.
Member, Sectional Committee X3.4.2 on Language Specifications, United States of America Standards Institute; Committee on Preparation of Examination Questions, DPMA.

Rome, B. K. and Rome, S. C.
Positive feedback in large organizations. Presented to the American Society for Public Administration, China Lake Chapter, China Lake, California, February 11.

Rosenbaum, J.
Computer-based instruction. Presented to the Claremont Chapter, Phi Delta Kappa, Claremont, California, October 1.

Four university-based systems. In The computer in physics instruction. (A Report on the Conference on the Uses of the Computer in Undergraduate Physics Instruction, at the University of California, Irvine, November 4-6, 1965). Published by the Commission on College Physics, 1966, 49-50.

Rosenquist, B. A.
The systems approach for planning and evaluation. Presented to the National Seminar for Research in Vocational Education, University of Illinois, Urbana, May 16-20.

See also Cogswell, J. F.

Sackman, H.
Toward democratic and experimentally regulated evolution of computer-serviced societies. Presented at Fresno State College, California, May 20.

See also Grant, E. C.

Schwartz, J. I.
On-line programming. Communications of the ACM, 9 (3), 199-202.

Problems in time-sharing. Presented to the Cincinnati Chapter of the ACM, January 6; the Chicago Chapter, January 7; the Rochester Chapter, April 11; the Syracuse Chapter, April 12; the Pittsburgh Chapter, April 13; the Kentucky Chapter, Lexington, April 14; and the Orange County Chapter, Anaheim, October 11.

Invited Participant, EDUCOM Summer Study Conference on Information Networks, Boulder, Colorado, July 5-29.

Sedelow, S. Y.
The computer and verbal data processing. Presented at the University of North Carolina, Chapel Hill, February 25. Also presented at the University of Connecticut, Storrs, April 22.

Learning the new methodologies. Presented at the American Education Research Association, Chicago, February 18.

Literary pattern and the computer. Presented at Duke University, Durham, North Carolina, November 16.

Patterns in natural language. Presented at the California Institute of Technology, Pasadena, March 17.

Teaching style with the aid of the computer. Presented at the National Council of Teachers of English, Houston, November 25.

Shaffer, S. S.
Chairman, National Constitution and Bylaws Committee, ACM. Chairman, Educational Committee, San Fernando Valley Chapter, ACM.

Shaw, C. J.
Assemble or compile? Datamation, 1966, 12 (9), 59-62.

Computer software: Developments and recent trends in programming and operating systems. Presented to the EDP Committee of the League of California Cities, Santa Monica, August 26. (Available as SDC document SP-2582.)

Practical aspects of computer programming languages. Presented at the Computer Installation Management Course, sponsored by the California State Personnel Board, Sacramento, October 4.

Specifying object-code efficiency. Datamation, 1966, 12 (4), 70.

Member, Technical Program Committee, 1966 National ACM Conference, Los Angeles, August 30-September 1. Chairman, Session on Programming Languages for Symbolic, Numeric, and Hybrid Computation.

Panel Moderator, Session on The Spaceborne Common Programming Language, 1966 IEEE Aerospace Computer Symposium, Santa Monica, October 24.

Chairman, ACM Special Interest Committee on Programming Languages; Editor, SICPLAN Notices. Vice-Chairman, Los Angeles Chapter of the ACM. Recipient, Outstanding Member Award for 1966, Los Angeles Chapter of the ACM.

Member, Editorial Board, Annual Review in Automatic Programming.

Shure, G. H.
Experimental bargaining and negotiation. Presented at the Social Psychology Colloquium, Columbia University, Teachers College, New York City, December 1.

Methodological issues in the study of threat in the laboratory. Presented at the 30th National Meeting of the Operations Research Society of America, Durham, North Carolina, October 17-19.

New computer developments in data processing. Presented to the International Relations Staff, Northwestern University, Evanston, Illinois, September 1.

An overview of TRACE and some research implications. Presented at the Annual Convention of the American Psychological Association, New York City, September 2-6.

TRACE: Lecture and demonstration. Presented to the American Statistical Association, Los Angeles, August 12.

TRACE: Timeshared routines for analysis, classification and evaluation. Presented at the TRACE Symposium, sponsored by APA Divisions 5 and 21, American Psychological Association, New York City, September 2-6.

Invited Participant, ONR Transnational Meeting of Social Psychologists on the Dynamics of Conflict, Nice, France, January 11-16.

Shure, G. H. (cont.)
Co-chairman, Third Transnational Conference on the Dynamics of Conflict, Santa Monica, November 17-23.

Associate Editor, Sociometry; Editorial Consultant, Behavioral Science and Canadian Journal of Psychology.

Shure, G. H. and Meeker, R. J.
Some observations on cooperation in laboratory studies of bargaining behavior. Presented at the West Coast Conference for Small Group Research, Long Beach, California, April 28.

Silberman, H. F.
Learning theories and instructional systems. Presented at the National Conference on Systems Approaches to Curriculum and Instruction in the Open Door College, University of California, Los Angeles, July 18.

Opportunities for technical writers and artists in programmed instruction. Presented to the Los Angeles Chapter of the Society of Technical Writers and Publishers Seminar on New Job Opportunities for Technical Writers and Artists, Los Angeles, January 22.

Problems with using computers in education. Presented at the USAFE Education Conference, Wiesbaden, Germany, October 4.

Southwest regional laboratory for educational research and development. Presented at the California Association of School Psychologists and Psychometrists, Los Angeles, March 18. Also presented at the Spring Conference of the California Association for Supervision and Curriculum Development (Southern Section), Riverside, May 5.

The systems approach, technology, and the school. Presented to the San Fernando Valley Chapter, Systems and Procedures Association, Los Angeles, January 19. (Available as SDC document SP-2025.)

Using computers in education: Some problems and solutions. Presented to the Valley Chapter, Systems and Procedures Association, Woodland Hills, California, December 21. Also presented at the Symposium on Educational Technology and Evaluation, sponsored by the Research and Development Center, University of California, Los Angeles, December 15. (Available as SDC document SP-2545/002/00.)

Discussant, Symposium on Laboratory Simulation: New Developments in Instruction and Research, American Educational Research Association Meeting, Chicago, January 17-19.

Invited Participant, Workshop on Communications in the New Media, sponsored by California State Department of Education, El Camino College, Los Angeles, November 10.

Invited Participant, Seminar on the Impact of Technology on Public School Education, sponsored by the Charles F. Kettering Foundation, New York City, November 16.

Chairman, Joint Committee on Data Processing for the Regional Laboratories for Educational Research and Development. Appointed Member, Research Commission, National Association for Supervision and Curriculum Development. Member, Southern California Advisory Council for Educational Television; Advisory Committee, California State Department of Education; EDUCOM Task Force on Educational Systems and Technology.

Member, Committee on Automation in Reading and National Council on Measurement in Education, International Reading Association.

Member, Programmed Instruction Committee, Division of Educational Psychology, American Psychological Association.

Member, Standards Committee on Teaching Machines, American Educational Research Association.

Member, National Joint AERA-APA-DAVI Committee on Standards for Auto-Instructional Programs and Devices.

Consultant and Proposal Reviewer, U. S. Office of Education Bureau of Research. Consultant on Curriculum Development, American Society for Engineering Education. Board Member, Doctor of Education Association, University of California, Los Angeles.

Editor, A symposium on proposals for increasing the effect of educational research on classroom instruction. Harvard Educational Review, 1966, 36 (3), 295.

Simmons, R. F.
On-line interactive displays in application to linguistic analysis and information processing retrieval. Presented at the Man/Machine Interactions Symposium, Paris, France, October 10-15. (Available as SDC document SP-2432/001/00.)

Verbal inference in question-answering systems. Presented at the Linguistic Institute of the Linguistic Society of America, University of California, Los Angeles, July 14.

Principal Reviewer, Progress Review Panel on Automated Language Processing, Annual Meeting of the American Documentation Institute, Santa Monica, October 3-7.

Simmons, R. F. (cont.)
Visiting Professor, Linguistic Institute of the Linguistic Society of America, University of California, Los Angeles, June 17-August 12.

Staudhammer, J. and Ash, M.
A sufficiency solution of the traveling salesman problem. Presented at the 29th National Meeting of the Operations Research Society of America, Santa Monica, May 20.

Steel, T. B., Jr.
Informal summary of the proceedings of the conference. Dinner Address, IFIP TC-2 Working Conference on Symbol Manipulation Languages, Pisa, Italy, September 4-9. Also Chairman, Panel on Basic Concepts in String Manipulation.

Promising avenues for future research--software. Presented at the IEEE Computer Group Chapter Meeting, Palo Alto, California, March 22.

Software state of the art. Presented at the Spaceborne Computer Software Workshop, El Segundo, California, September 21.

Time-sharing--not this August. Presented to the Computer Association of Minnesota, St. Paul, October 10.

Invited Participant, Panel on Advances in Programming Languages. AFIPS Fall Joint Computer Conference, San Francisco, November 8-10.

Secretariat, Advisory Council, ACM Standards Committee; Charter Member, ACM Special Interest Group for Programming Languages; Member, ACM National Program Committee, Special Interest Group on Business Data Processing, and Standards Committee Steering Committee.

Chairman, ASA X3.4 Common Programming Languages Subcommittee; Member, X3.4 Membership Committee and X3.4.6 Glossary Working Group.

Referee, AFIPS Fall Joint Computer Conference, San Francisco, November 8-10; Member, AFIPS International Relations Committee.

Organizer, IFIP TC-2 Working Group 2.2 on Description Languages; USA Representative to IFIP Technical Committee-2, Programming Languages; Member, Organizing Committee, 1966 IFIP TC-2 Working Conference on List Processing Languages, and Organizing Committee, 1967 IFIP TC-2 Working Conference on Simulation Languages.

Member, SHARE Executive Board; SHARE Representative to JUC; Manager, SHARE Advanced Planning Division.

Editor, Journal of Formal Languages; Referee, Journal of the ACM and Computing Reviews.

Editor, Formal language description languages for computer programming. Proceedings of the IFIP TC-2 Vienna Conference, September 15-18, 1964. Amsterdam, Holland: North-Holland Publishing Company, 1966.

Summerfield, A. B.
Automated data processing in hospitals. Presented at the Preconference Symposium on Hospital Administration, Elsinore, Denmark, April 20-23.

Computers in medicine--the broad view. Presented to the Madison County Medical Society, Anderson, Indiana, March 21.

Information management systems and medical information; and Real-time information systems for medicine. Invited lectures presented at University of California, Los Angeles, July 18-29.

Introduction to systems analysis. Presented at the International Advanced Symposium on Data Processing, Elsinore, Denmark, April 25-May 3. Also presented to the Finnish Hospital Association, Helsinki, Finland, May 9.

Totschek, R. A.
Invited Participant, Workshop on Simulation Languages, University of Pennsylvania, Philadelphia, May 17-18.

Vorhaus, A. H.
A general-purpose display system for manipulating a data base. Presented to the AMA Continuing Seminar on Management Information Systems, New York City, September 27.

Invited Participant, ARPA Contractors' Meeting on Graphics, Lincoln Laboratory, Cambridge, Massachusetts, April 7-8.

Invited Participant, Panel on Data Base Systems, sponsored by the Diebold Research Group, Minneapolis, June 6-7.

Walker, A. C.
ADI 1966--some new concepts in conference design. Los Angeles Engineer/Scientist, 1966, 5 (10), 6.

The role of the ADI in information science and technology. Los Angeles Engineer/Scientist, 1966, 5 (9), 13.

Walker, A. C. (cont.)
Program Design Chairman, 29th Annual Meeting of the ADI, Santa Monica, October 3-7.

Wallace, E. M.
Rank order patterns of common words as discriminators of subject content in scientific and technical prose. Statistical Association Methods for Mechanized Documentation, Symposium Proceedings, Washington 1964. National Bureau of Standards Miscellaneous Publication 269, December 15, 1965, 225-229.

User requirements, personal indexes and computer support. Presented at the 29th Annual Meeting of the ADI, Santa Monica, October 3-7.

Vice-Chairman, Los Angeles Chapter of the ADI; Chairman, Paper Selection Committee, 1966 Annual ADI Meeting; Member, Editorial Committee, ADI Proceedings; Chairman, Ad Hoc Committee Supervising the Institute's Exhibit.

Walsh, J. E.
Vice-President and President-Elect, Operations Research Society of America.

Lectures in operations research. Presented in Tokyo to the Chiefs of Staff of the Japan Air Self Defense Force, July 11 and 15; Chiefs of Staff, Japan Maritime Self Defense Force, July 13; Chiefs of Staff, Japan Ground Self Defense Force, July 14; Japan Defense Agency personnel, July 18; Defense Academy, July 25; Air Staff College, July 26; Japanese Union of Scientists and Engineers, July 26; and Operations Research Society of Japan, August 2.

Statistical validation of simulation by comparison with operating systems--illustrated for traffic problems. Presented to the North Texas Chapter of the American Statistical Association, Dallas, October 6.

Two problems in biostatistics. Invited symposium lecture at the University of Florida, Gainsville, February 28. Also presented to the Medical Department, Japan Air Self Defense Force, Tokyo, July 12.

Uses and theory of linearized nonlinear regression. Presented to the Chinese Society for Quality Control, Taipei, Formosa, July 7. Also presented to the Institute of Statistical Mathematics, Tokyo, July 12.

Panelist, Sessions on Data Sources and on Educational Considerations, The Future of Management Sciences Symposium, The Institute of Management Sciences, Miami, October 10-11.

Member, Council for International Federation of Operational Research Societies; Coordinator, Operations Research Around the World Meetings. Chairman, 29th Meeting, ORSA. Session Chairman, 12th Annual Meeting of Western Section, Operations Research Society of America, Stateline, Nevada, June 13-14.

Visiting Professor of Statistics, Southern Methodist University, Dallas, Texas.

Adjunct Professor of Business Administration, University of Southern California, Los Angeles.

Reviewer, Zentralblatt für Mathematik and International Abstracts in Operations Research. Referee, American Mathematical Monthly and Journal of the American Statistical Association.

See also Gafarian, A. V.

Weis, R. and Katter R. V.
Multidimensional scaling of documents and surrogates. Presented at the Annual Meeting of The American Statistical Association, Los Angeles, August 15-18.

Weissman, C.
Cyberculture, bane or boon? Presented at The Leisure Time Symposium, sponsored by the Council of Jewish Women, Los Angeles, March 22.

Input-output in LISP 2--an approach to device and machine independent I-O. Presented at Bolt, Beranek and Newman, Cambridge, Massachusetts, January 10-11.

Management of system resources. Presented to the Special Interest Group on Program Management of the ACM, Los Angeles, July 19.

Member, Technical Program Committee, 21st National Conference of the ACM, Los Angeles, August 30-September 1. Also Panelist, Session on Program Organization in a Multi-processing Environment. Publications Chairman, Los Angeles Chapter, Special Interest Group on Time-sharing.

See also Abrahams, P. W.

Yarnold, K. W.
Chairman, Membership Committee, Operations Research Society of America.