AD-752 797

Computer-Assisted Planning

System Development Corporation

prepared for

Advanced Research Projects Agency

SEPTEMBER 1972

Distributed By:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

UNCLASSIFIED
Security Classification

AD-752-797

DOCUMENT CONTROL DATA - R & D

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

1. ORIGINATING ACTIVITY (Corporate author)
SYSTEM DEVELOPMENT CORPORATION
2500 Colorado Avenue
Santa Monica, California 90406

2a. REPORT SECURITY CLASSIFICATION

2b. GROUP

3. REPORT TITLE
COMPUTER-ASSISTED PLANNING: Final Technical Summary Report to the Director, Advanced Research Projects Agency

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Technical - 16 September 1971 to 15 September 1972

5. AUTHOR(S) (First name, middle initial, last name)
Morton I. Bernstein

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
15 September 1972	iii, 48	26

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
DAHC15-67-C-0149	TM-3628/010/00
b. PROJECT NO. 2D30	
c. Program Code No. 2P10	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d. ARPA Order No. 1327	

10. DISTRIBUTION STATEMENT
APPROVED FOR PUBLIC RELEASE - DISTRIBUTION UNLIMITED

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Advanced Research Projects Agency Information Processing Techniques Office

13. ABSTRACT

This document reports results of applied research in four task areas: (1) Computation and Communication Trade-Off Studies (CACTOS); (2) Vocal English Data Management; (3) On-Line Graphic Computation; and (4) Systems Research and Networking. Work in these areas has been directed toward the development of prototype facilities for a computer-assisted planning system. These facilities include a computer-network modeling system (the CACTOS model); systems for man-machine communication and data management via speech, ordinary English, and data-tablet graphics; and an initial facility for data-sharing among existing data management systems within a computer network. The work has also involved the continuing evolution of a sophisticated time-sharing system (ADEPT/ICOS) and the implementation of protocols for communication with other Hosts over the ARPA Network.

Ia

DD FORM 1 NOV 68 1473

UNCLASSIFIED
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
acoustics research						
Advanced Development Prototype (ADEPT)						
ARPA Network						
AUTODIN						
CACTOS						
computer graphics						
computer networks						
CONVERSE						
cost-effectiveness, computer network						
data sharing						
data structures						
DEC PDP-11/05						
deductive inference						
DS/2						
English recognition						
FORTRAN						
Honeywell DDP-516						
IBM 370/145						
ICOS						
image processing						
JUMPS/MMS						
language processing						
linguistics						
LISP						
logic						
man-machine communication						
man-machine interface						
modeling						
query languages						
Raytheon 704						
signal processing						
speech understanding						
TAM						
TELNET						
time-sharing						

Ib

SYSTEM DEVELOPMENT CORPORATION

COMPUTER-ASSISTED PLANNING

FINAL TECHNICAL SUMMARY REPORT TO THE DIRECTOR, ADVANCED RESEARCH PROJECTS AGENCY

16 SEPTEMBER 1971 TO 15 SEPTEMBER 1972

Sponsored by the Advanced Research Projects Agency
ARPA Order No. 1327

TM-3628/010/00

SYSTEM DEVELOPMENT CORPORATION

COMPUTER-ASSISTED PLANNING

FINAL TECHNICAL SUMMARY REPORT TO THE DIRECTOR, ADVANCED RESEARCH PROJECTS AGENCY

THE WORK REPORTED HEREIN WAS SUPPORTED BY THE ADVANCED RESEARCH PROJECTS AGENCY OF THE DEPARTMENT OF DEFENSE UNDER CONTRACT DAHC15-67-C-0149, ARPA ORDER NO. 1327, PROGRAM CODE NO. 2P10.

M. I. BERNSTEIN

Tel. (213) 393-9411

16 SEPTEMBER 1971 TO 15 SEPTEMBER 1972

TM-3628/010/00

(page iv blank)

TABLE OF CONTENTS

LIST OF FIGURES

1. SUMMARY

This Technical Memorandum is the final report to the Advanced Research Projects Agency (ARPA) on System Development Corporation's (SDC's) research in Computer-Assisted Planning (CAP) for the period 16 September 1971 through 15 September 1972. The CAP program, which has been ongoing at SDC since 1967, derives from earlier ARPA-sponsored research in Computer-Aided Command utilizing the Advanced Development Prototype (ADEPT), a time-sharing system particularly well suited to problems of the magnitude of CAP.

The CAP program has been directed toward the long-range goal of clarifying and systematizing the strategic planning process, with the ultimate objective of constructing a prototype computer-based planning system. Such a system would have a major impact on the Department of Defense. In addition to providing the basis for more effective and efficient planning, it would help to regularize the planning process, presumably enhancing the understanding and acceptability of both the process and its product. Most tangibly, the system would permit rapid redirection of resources in response to shifting demands and with clearer prior knowledge of implications and side effects. In the five years since the program's inception, the major research emphasis has been on developing prototypes of some of the components of such a system, and particularly the communications components--the facilities at the man/machine interface. Improved man-machine communication, in and of itself, contains the potential for improving every computer-based function upon which military and other governmental planners and analysts are becoming increasingly dependent. Putting the ultimate user of information in direct contact with the information base, in a mode that is normal and natural to his training and discipline, can enhance productivity and creativity at all operational levels. Consequently, prototype systems for communicating with computers in ordinary English, by voice, and by means of hand-drawn images and mathematical notations have been or are being developed.

The other emphasis of this research program has been on the development and refinement of basic computer time-sharing and networking technology. A model and procedures to assist planners in the design, development, and improvement of networks of intercommunicating computer systems has been constructed and is operational, and work on time-sharing, both locally and through the ARPA Network, has been steadily progressing.

The following paragraphs summarize the direction and accomplishments of these activities during the past year.

Computation and Communication Trade-off Studies (CACTOS)

The CACTOS project, which was completed this year, had two related objectives. The first was to examine the 1975-80 Department of Defense (DoD) requirements for computation and communication resources, with particular attention to the trade-offs that are feasible between concentrated and distributed computation power. The second was to build a network modeling tool with which to perform sufficient trade-off analyses to validate both the assumptions upon which DoD computation/communication requirements are based and to validate the model itself, so that it may be used by others as a practical planning tool. The results of these efforts are reported in summary in this report and in detail in a companion report (SDC document TM-4743/013/00, in press).

Vocal English Data Management

This task comprises two separately identifiable projects--Voice Input/Output and CONVERSE--with separate short-range goals but a single long-range goal. The long-range goal is to provide the facility for a man to query a computer-based information management system by spoken English sentences; the shorter-range goals are to develop separate systems for processing English and human speech. The Voice Input/Output Project, one of a group of ARPA-supported speech understanding research efforts, has focused on developing the various components of a projected Vocal Data Management System that will extract meaningful information from a speaker's input utterance, using both highly complex signal-processing techniques and prediction constraints that narrow the range of likely utterances within the bounds of a reasonable processing capability. Ultimately, a full-scale speech understanding system would probably be required to respond appropriately to the normal speech of any person, in acoustic environments that might range from highly controlled sound studios to telephone or radio channels. Determining whether such a capability is feasible in the next three to five years will be one purpose of this research; in the meanwhile, the project is progressing toward a more limited capability for accepting input from a few speakers for whom the system has speech profiles and who are speaking the highly constrained subset of English phrases that make up the DS/2 data management system language. During this year, the project devoted most of its effort to implementing several large signal-processing and word-prediction systems on its Raytheon 704 computer.

The CONVERSE (English Input/Output) Project's goal is to develop powerful language-processing techniques that will permit communication with a data management system by keyboarded English sentences. Improvements have been made this year in both the variety and the complexity of linguistic constructs that the CONVERSE system can handle. In addition, attention has been given to the problem of diversifying the kinds of data bases to which queries can be addressed. Specifically, a data base of computer programs has been implemented that contains sufficient richness that, by means of a few simple English queries,

a user may readily determine which, if any, of the programs cataloged are useful to him. In the course of exploring ways of simplifying some of the highly complex syntactic and semantic extraction processes involved, a relatively simple yet potentially powerful English-like keyword technique was discovered; further research into the boundaries of its applicability and its shortcomings is being carried out in order to assess its full potential. Finally, the full-scale implementation of a deductive inference component was begun. This component will provide the system with the crucial capability of inferring the meaning of sentences that--like the vast majority of sentences used by speakers of English--are truncations or paraphrases of the formally complete propositions they represent.

On-Line Graphic Computation

Research in prior years led to the development of an interactive data-tablet system for inputting hand-printed mathematical expressions as the principle means of directing computations. During this past year, that system, The Assistant Mathematician (TAM), was refined and improved. More importantly, the scope of the effort was enlarged; TAM and its ancillary graphic-processing technology will now form the basis of an interactive image and signal processing system. Much of the effort this year went toward (1) defining the problem and (2) defining and designing the overall approach and the individual components required by the user of such a system--including the hardware, the operating system environment, and the actual tools and techniques that the user needs or would find useful. The guiding philosophy underlying the design is that user-defined processes should be created in the most natural way and that storage management (image files are large and multitudinous) and historical (or audit) trails should impose no burden on the experimentor. Experimental implementation of some image and signal processing has begun and the overall system definition and design are nearing completion.

Systems Research and Networking

The ADEPT time-sharing system, first implemented in 1965, has moved from the development computer, an IBM 360 Model 50, through several computers and is now resident on an IBM 370 Model 145 in an expanded and enhanced form, called the Interactive Common Operating System (ICOS). Many modifications have been made to accommodate the needs of researchers using it as a base. It now handles larger, more complex programs and a greater number of users at one time, and it supports a wide diversity of interactive consoles and devices. The system's mobility has had a delaying effect on stabilizing SDC's ARPANET Network Control Program (NCP) and Host-to-Host protocols, but the ARPANET work has progressed and will be enlarged during the coming year.

The Computer Network Data Sharing Project is attempting to define and test the feasibility of sharing data distributed among the diverse and largely incompatible data management systems resident in a computer network. The basic

2. COMPUTATION AND COMMUNICATION TRADE-OFF STUDIES (CACTOS)

2.1 PROGRESS

The objective of the two-year Computation and Communication Trade-Off Studies (CACTOS) Project, which was completed as an ARPA-supported project this year, was to develop a means for assessing the performance of computer networks, especially those operated or being planned by the Department of Defense. While it has become clear that computer networks will be a significant factor in meeting the increasing demand for world-wide computation and communication, it has not been correspondingly clear how best to design such networks from a computation viewpoint in order to achieve improved cost-effectiveness. In particular, there has been a need for a clearer understanding of the operational and fiscal trade-offs between communications resources and computational resources. The CACTOS Project addressed this problem.

During the first year of the project, a comprehensive analysis was made of the anticipated future (1975-1980) Department of Defense requirements for computational and communication resources and of likely developments in computer and communciations technology. An analysis was also made of the relationships between the many parameters that characterize a computer network and the network's performance of various kinds of computational tasks. These parameters include, among other things, the size, speed, and location of the computers, the characteristics of the communication channels connecting them, the number and types of messages sent over the channels and the manner in which messages are routed and distributed. The performance measures encompass throughput, cost, response time, reliability, and accuracy. A computer-based network modeling and analysis system that can simulate the performance of a network with specified parameters was developed in prototype form, and was used to conduct initial analysis of the Joint Uniform Military Pay System/Marine Manpower Management System (JUMPS/MMS), which operates on several computers interconnected via DoD's Automatic Digital Network (AUTODIN). The results of the first year's work were published in an interium report.[1]

The major tasks for this year were to expand, refine, and validate the modeling system, conduct experiments with several existing and planned computer networks in order to arrive at conclusions with respect to computation resources, and obtain guidelines for use in the design, construction, and modification of computer networks. The results of this year's work are reported in detail in a separate final report (TM-4743/013/00, in press) and are briefly summarized here.

---

[1] SDC TM-4743/012/01.

In its prototype form, the network modeling system was oriented toward analysis of communications performance and was relatively insensitive to modifications in network computational resources. This limitation was removed by the incorporation of capabilities for detailed analysis of computation characteristics, including memory size, the balance between input/output (I/O), and central processing unit (CPU) speeds, instruction rates, word sizes, and other important machine characteristics. Refinements were also made in the system's sensitivity to different types of messages.

Two major sets of experiments were conducted. The first set, which was run using a hypothetical 40-node network with centers located in 26 cities across the U. S., was designed to examine performance variations associated with centralized versus decentralized computation. The results indicated that large computers distributed in a semi-centralized configuration were more cost-effective than either fully decentralized processing using more, smaller machines or a fully centralized large computer concentration at a single urban complex. This preference held for almost all realistic parameter values. Cost-effectiveness was defined as the ratio of workload to the product of monthly total cost and average response time. The superior performance of a CPU-oriented (e.g., scientific) network was also demonstrated.

The second set of experiments was oriented toward obtaining guidelines for use in the design, construction, and modification of computer networks. These experiments were run using a completely connected eight-center network with equal-capacity lines. The lines were removed in stepwise fashion, based on the criteria of least loaded or least cost-effective, and the results of this method of altering network configuration were compared to the results of *a priori* specification of standard configuration types, such as ring and star configurations. The general conclusion was that a link-removal method is superior to the specification of standard configurations because it is more sensitive to the message traffic requirements of the network. It was also concluded that several methods of link removal should be tried in order to achieve the most cost-effective configuration.

Finally, the network modeling system was applied to the operational analysis of one existing and one proposed network and to the requirements analysis for a second proposed network. The existing network was the centralized JUMPS/MMS referred to earlier, and the evaluation had two main objectives. The first was to validate the modeling system itself by comparing its simulated results with actual AUTODIN statistics; the variance was less than 5%. A second objective was to evaluate JUMPS/MMS in order to reduce its response time, which was ranging from 5 to 10 days. It was found that some decentralization of data bases and computer power would result in a 5-10% reduction in time delay. It was also found that a dedicated message-switching system or higher message priorities in AUTODIN would further shorten single-message response times to less than half their present duration.

The second system analyzed was the proposed Air Force Advanced Logistics System (ALS), a decentralized network of six computer centers. The analysis produced performance measures for several alternative configurations of these six centers.

The third analysis was oriented toward determining the most cost-effective dedicated configuration for a projected General Services Administration (GSA) computer network with nodes in 13 cities across the U. S. The analysis, which was based on a modification of the GSA Request for Proposal, indicated the available trade-offs between computation and communications resources. A fiscal ceiling was assumed for system operation. The preferred configuration was large computers distributed at two locations (semi-centralized).

2.2 SUMMARY

The CACTOS Project has developed a computer-based modeling system for analyzing and evaluating computer networks in order to establish optimum balances between computation and communications resources in meeting cost-effectiveness and performance requirements. The system is written in FORTRAN and can be adapted to run on most available medium or large computers. It incorporates most of the relevant parameters for both the computation and the communications characteristics of a computer network. It has been validated and used to analyze several existing and proposed networks. Development was based on a comprehensive study of the technological and economic factors in the design and construction of military computer networks, particularly as they can be anticipated for the 1975-1980 time period.

The primary issue investigated by the system was the question of whether, in general, centralized or decentralized (distributed) computational power offers the best potential performance and cost-effectiveness for present and future computer network configurations. The conclusion was that partial decentralization, using large computers to achieve economies of scale, provides optimum results for the types of computer networks that will be constructed to meet the needs of the Department of Defense during the 1975-1980 time period. The detailed results of the CACTOS investigations are reported in two separate documents, listed below in Section 2.4.

2.3 STAFF

CACTOS

Dr. B. P. Lientz, Principal Investigator
T. A. Brotherton
G. M. Cady
D. Lashier (part-time)
Marilyn Pillon
Luanne Waul
Dr. N. E. Willmorth

2.4 DOCUMENTATION

Cady, G. M. Computation and Communication Trade-Off Studies: An Analytical Model of Computer Networks. SDC SP-3656. (Presented at the Western Electronic Show and Convention, Los Angeles, September 1972. Also included as Appendix 1 to SDC TM-4743/013/00, cited below.)

Lientz, B. P.; Cady, G. M.; Pillon, M; Willmorth, N. E. Final Report of the CACTOS Project: Investigation of Computation and Communication Trade-Offs in Military Command and Control Systems. SDC TM-4743/013/00, 15 October 1972 (in press).

Willmorth, N. E. Report of the CACTOS Project: A Preliminary Investigation of Computation and Communication Trade-Offs in Military Command and Control Systems. SDC TM-4743/012/01, 1 April 1972.

3. VOCAL ENGLISH DATA MANAGEMENT

This task, whose long-range goal is the development of a vocal CONVERSE--a data management system in which ordinary English speech is the data management language--comprises two projects representing separate but related technologies: the CONVERSE Project, representing the technology of computer processing of natural language (English); and the Voice Input/Output Project, representing the technology of speech understanding by computer. The CONVERSE system embodies capabilities for processing natural English within the context of general-purpose data-base interrogation and manipulation. It handles both large grammars and large domains of discourse, using a natural-language compiler, a formal Intermediate Language, and a concept net and fact file that organize items of conceptual and factual information.

The Voice Input/Output Project, now completing its second year, is pursuing speech understanding by computer on the basis of the hypothesis that both the human's "inquiry state" and the words and phrases he uses can be dynamically modeled and predicted to greatly increase the likelihood that the computer will recognize and interpret his spoken utterances.

3.1 VOICE INPUT/OUTPUT

The long-term goal of the Voice I/O Project is the construction of a vocal CONVERSE--that is, a speech system that accepts a reasonable subset of English. A major milestone along that path will be the implementation of a Vocal Data Management System (VDMS), which resembles DS/2 or TDMS and accepts a more austere, constrained language. Two distinct data bases have been selected to provide domains of discourse for VDMS. These are: (1) data relevant to a subset of computer programs in the SHARE library, and (2) information pertinent to the submarine fleets of the United States, USSR, and United Kingdom. With reference to the computer program data base, the user may make the following sample requests:

(1) Print program name where class equals quadrature
(2) Repeat where category equals numerical analysis
(3) Count where type equals linear equations and runtime less than five six
(4) Print entries where key word equals root squaring

In addition, sample requests of the submarine data base are:

(1) Total missiles where country equals USSR
(2) Print category where class equals attack and country equals USA
(3) Repeat where class equals training
(4) Count where country equals USSR and torpedo tubes greater than or equal to six

VDMS was chosen as the experimental test bed for several reasons. First, the constrained nature of the language makes it practical. Second, the vocabulary size, breadth of the language, and kinds of non-acoustic sources of knowledge may be varied easily, allowing good instrumentation to determine how much of the system works and which parts contribute what (and how much) to the recognition process. Also, the implementation of VDMS will require research (and hopefully, progress) in many of the outstanding problem areas in speech understanding. Among these are:

- recognition of continuous speech
- vocabulary size
- syntax support
- semantic support
- predictive grammars
- user modeling
- reliability
- system test procedures
- system organization

The following reports progress made during the past year and plans for the coming year.

3.1.1 Progress

During the past year, the Voice I/O Project implemented several large systems and programs. Among these are VOICEBOX, SYNTHBOX, and TOPS.

The VOICEBOX system includes an interactive interpreter and a group of overlays comprising more than 100,000 machine instructions operating on the Raytheon 704 computer. The overlay sets are GRAPHBOX, MATHBOX, LISTBOX, EDITBOX, and VRBOX. (Soon to be incorporated in the system is SYNTHBOX, described below.)

MATHBOX computes Fast Fourier Transforms, LPC[1] spectra and cepstrally smoothed log spectra for formant analysis, and cepstra for pitch analysis of speech files. All of the functions operate on a 25.6-millisecond time slice, which is equivalent to 512 samples per frame at the present sampling rate of 20,000 samples/second. Before processing, the data are passed through a Hamming window for spectral shaping.

The GRAPHBOX overlay plots portions of speech files, MATHBOX output, and filter data on the TEKTRONIX. Segment boundaries may be displayed. LISTBOX provides tabular, symbolic output of any of the more than 20 types of data

[1] Linear predictive coefficients

files managed by the VOICEBOX system. Library and backup tapes may be created and used through EDITBOX. EDITBOX provides the capability of copying segments of speech files into a new file.

VRBOX is a reimplementation with modifications of the Speech Recognition System developed by P. J. Vicens and D. R. Reddy at Stanford University. It consists of the SMOOTHBOX, SEGBOX, RECBOX, MAPBOX, and LEXBOX overlays. SMOOTHBOX normalizes and/or smooths filtered parameter data. SEGBOX does an initial partition of the signal into roughly defined acoustic groups called segments. RECBOX gives pseudo-phonetic labels to the segments. MAPBOX attempts to match the labeled utterance to the members of a lexicon and recognize the input. LEXBOX places the labeled segmented utterances in the lexicon and performs various lexicon-maintenance functions.

The VOICEBOX interactive language, VBL, has commands to pass control to any of the overlays, as well as functions to assist in all aspects of recording, playback, and acoustic control. In all, there are more than 70 available functions and more than a dozen infix and prefix operators. The following are examples of language usage (the % character introduces a comment):

```
RECORD+PLAY.                          %RECORD A SAMPLE AND PLAY IT BACK

LPC(321);GRAPH(SPEECH,LPCFILE).       %COMPUTE LPC AROUND CENTRAL
                                      %MILLISECOND 321 AND GRAPH IT ALONG
                                      %WITH A PORTION OF THE SPEECH FILE

LOOP WHILE ASK                        %ASK USER WHETHER TO RECORD ANOTHER SAMPLE
  SPEECH=SPEECH+1;                    %INCREMENT TO NEXT FILE NUMBER
  RECORD.                             %RECORD SAMPLE IN SPEECH FILE

FUNCTION FROLIC(SPEECH);PLAY.         %PLAYBACK THE CONTENTS OF THE
                                      %FILE PASSED AS ARGUMENT

FROLIC.                               %PLAY CONTENTS OF FILE SPEECH

FROLIC(700).                          %PLAY CONTENTS OF FILE 700
```

VBL allows abbreviations and has an optional audit-trail capability. The latter feature records the interaction between man and machine on a

selectable I/O device for later perusal. Besides interactive use, VBL may function as a job-control language for off-line, batch operations; for example:

```
LOOP WHILE RESTORE(SPEECH)              %RESTORE SPEECH FILES FROM LIBRARY
                                        %TAPE

   TAKEVR;QUSMOOTH;QUSEG;QUREC;EXIT;    %FILTER, SMOOTH, AND RUN SAMPLE
                                        %THROUGH VR SYSTEM

   GRAPH(QMATRIX,RMATRIX);              %GRAPH FILTER DATA "LINED" WITH
                                        %RECOGNITION BOUNDARIES THEN
   XEROX;                               %HARDCOPY GRAPH

   SAVE(QMATRIX)$                       %SAVE FILTER DATA ON NEW LIBRARY
                                        %TAPE

REWIND(BININ);                          %REWIND OLD TAPE AND CLOSE

REWIND(WEOF(BINOUT));                   %NEW ONE

XIT.                                    %ALL DONE; RETURN TO 704 MONITOR
```

SYNTHBOX is a four-pass speech synthesis-by-rule program based upon work done by Dennis Klatt of MIT.[1] The first pass accepts the input phonetic string spelled in the phonetic alphabet adopted by the ARPA speech contractors. The string is modified by a set of coarticulation rules for English and a duration is assigned to each phase, based upon stress information and neighboring constituents. The second pass determines the fundamental frequency (pitch) contour. The third pass draws smooth formant frequency and amplitude contours for the utterance. The fourth pass is a digital vocal-tract model that computes the actual points comprising the signal. Figure 3-1 is a graph of 50 milliseconds of synthesized speech and an LPC run over the central 25.6 milliseconds. The synthesized phone is OY in the word "boy."

---

[1] "Acoustic Theory of Terminal Analog Speech Synthesis." Conference Record, 1972 Conference on Speech Communication and Processing, Air Force Cambridge Research Laboratory AFCRL-72-0120, Special Reports No. 131, 22 February 1972, pp 131-135.

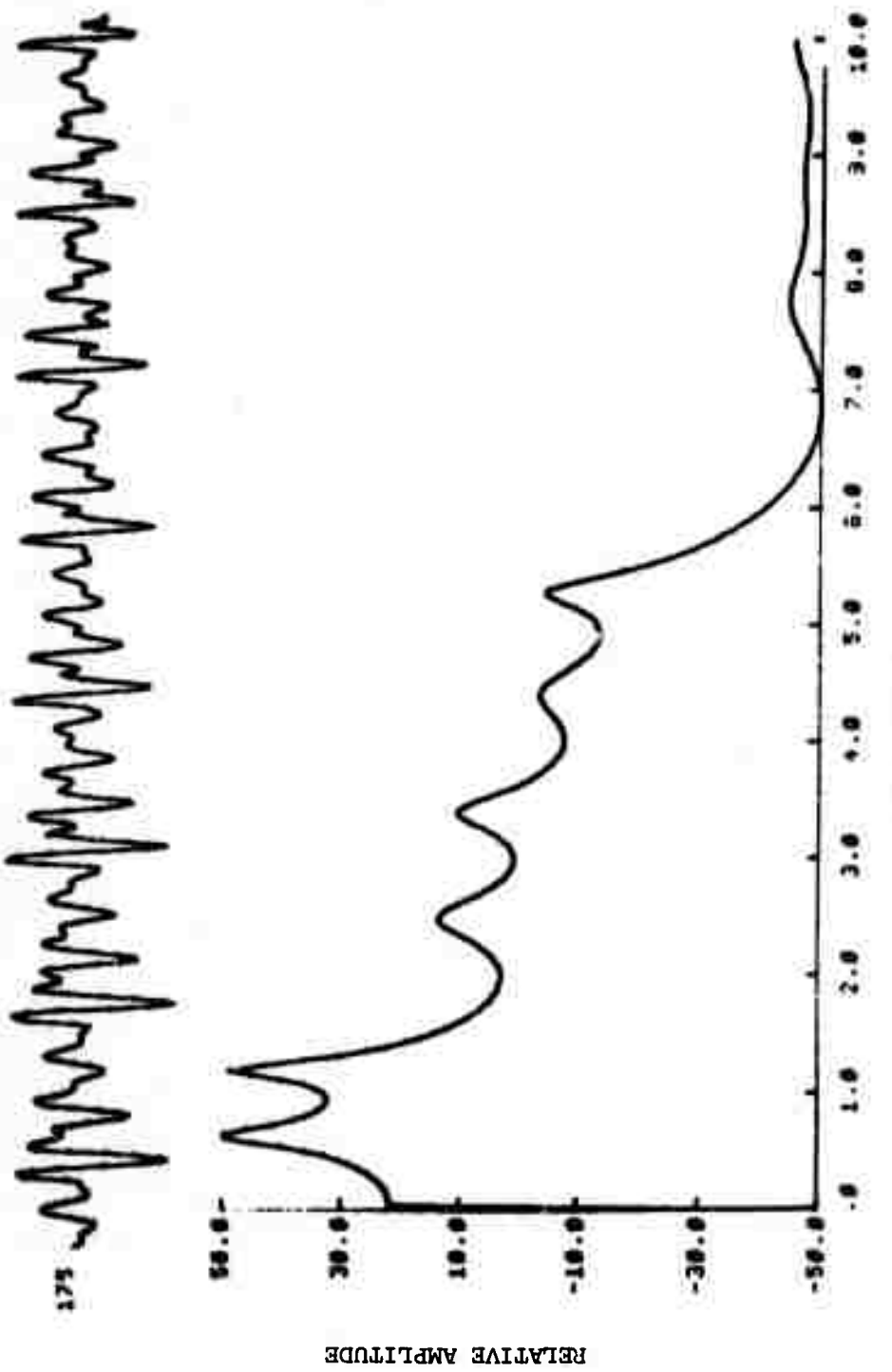

Figure 3-1. Synthesized Speech and LPC Run

Another system developed during the last year is TOPS, a top-end parser to be used for checking out the acoustic processing portions of the system. TOPS is written in LISP and operates on the Raytheon 704. It embodies several unusual capabilities and options. For example, the parser can be made to run either left to right or right to left, can accept either data management or chess-notation language (syntax developed at Carnegie-Mellon), and the bottom (acoustic end) may be approached in a variety of ways. The system is organized as five autonomous modules: EXEC, SYNTAX, SEMANTICS, ACOUSTICS, and DMS (see Figure 3-2). EXEC invokes each of the modules and controls data flow among them. At each stage of the parse, there is a context that represents the contiguous string of words found to date. SYNTAX takes the context and computes the set of possible next words or syntactic categories; e.g., the word WHERE or the class item-name. This determination is based solely on constraints arising from a context-free set of grammar rules.

The SEMANTICS module comprises two sub-components, LOCAL and GLOBAL. The LOCAL component allows or disallows various words in the proposed syntactic class, based upon the local context. The context rules were derived by a gross inspection of the context-free grammar rules, but the LOCAL module does not use the grammar rules directly. Examples are that no item-name may be used adjacent to itself or that a symbolic item-value may not appear after the relational-operator greater than. If the LOCAL semantics are turned off, then the module allows all members of the proposed syntax categories.

The GLOBAL component simply ranks by priority the set of alternatives allowed by the LOCAL component. The basic priority, given to each member of the allowable set, is a function of the context length and the size of the allowable set of next alternatives. GLOBAL, if the option is selected, augments the basic priority, using information gained from the last user interaction. As an example, if the system was not able to recognize the last input utterance, GLOBAL gives a high plausibility score to the key words introducing the "Debugging" commands: DESCRIBE, EXPLAIN, and SHOW.

The ACOUSTIC module is now simulated by a set of symbolic processing functions. In the near future, these will be replaced by the acoustic processing overlays, CWIPER (Contextual Word In Phrase Extraction Routine) and CWIPER*. The input to CWIPER is a list of predicted words and time constraints and a sureness of prediction (computed by LOCAL and GLOBAL) for each word. A search for each predicted word is performed, and an acoustic "merit" is computed. The merit score and the sureness-of-prediction value are combined to yield an overall "goodness-of-match." In the normal case, CWIPER returns all found words whose "goodness-of-match" scores are reasonable. The established time boundaries and "goodness-of-match" scores are returned with the words.

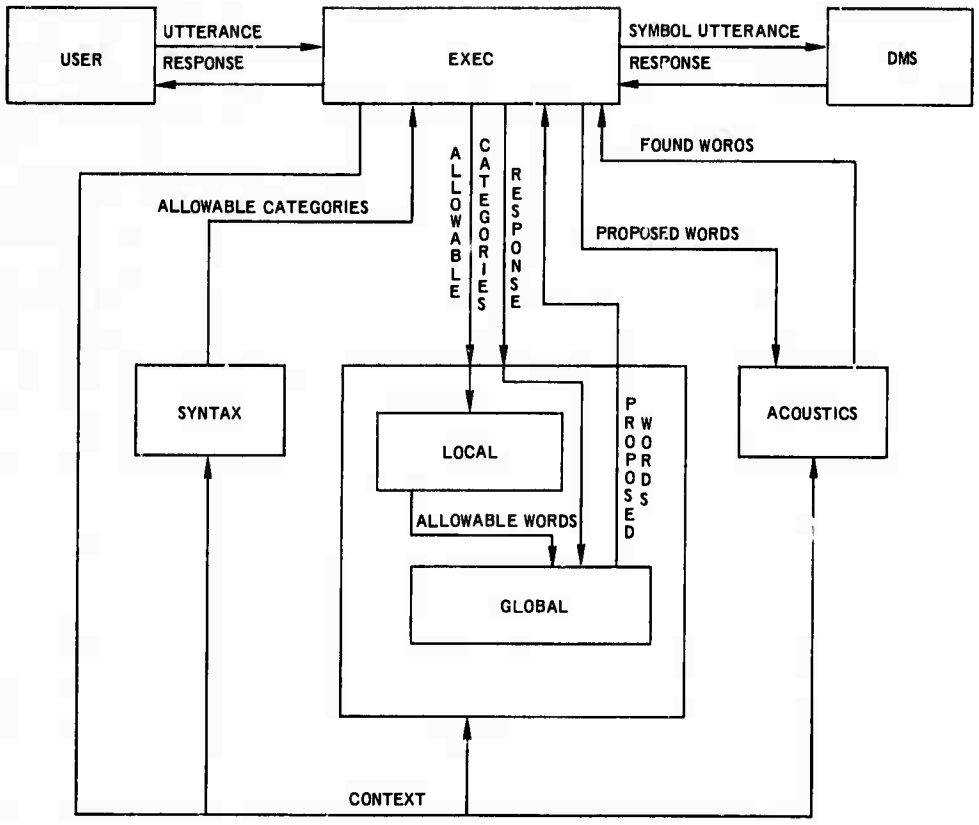

Figure 3-2. TOPS System Flow

```
SYNTAX PREDICTS
((*INAME* B 3 L 14))

SEMANTICS ALLOWS
((NAME B 3 L 14 69)
 ((KEY WORD) B 3 L 14 66)
 (SIZE B 3 L 14 66)
 ((EXECUTION TIME) B 3 L 14 66))

CWIPER FINDS
((NAME 3 4 79)(SIZE 3 5 70))

IN THE CONTEXT OF
((PRINT 0 1 65)(PROGRAMMER 1 2 75)(AND 2 3 68))
```

Figure 3-3. EXEC Output

The input to CWIPER* is a list of predicted words that share time boundaries and a sureness-of-prediction level. CWIPER* returns only the best match (assuming it exceeds some threshold). This logic is useful in picking one of a mutually exclusive class of words, say the digits. Figure 3-3 shows the trace output from one cycle of EXEC. TOPS is operating left to right, with both global and local options selected. In this case, CWIPER, rather than CWIPER*, is the simulated bottom end. The SYNTAX allows only an item-name (*NAME*) to follow in the context of the phrase PRINT PROGRAMMER AND . . . . (The B and L symbols denote time restrictions; B states that a boundary must occur at the specified time 3, and L states that a boundary may occur anywhere before--or, in right-to-left operation, after--the specified time 14.) From the list of possible item-names (NAME, KEY WORD, SIZE, EXECUTION TIME, PROGRAMMER), the LOCAL component has allowed all except PROGRAMMER, not liking the phrase PRINT PROGRAMMER AND PROGRAMMER. GLOBAL has added sureness-of-prediction scores ranging from 66 to 69, based upon the last utterance. The simulated CWIPER has found the two item-names NAME and SIZE, with NAME having a better score.

In addition to TOPS, ENQUIRE, a program developed by A. Leal of the CONVERSE Project, has been moved to the 704 LISP system. ENQUIRE (see Section 3.2) is a keyword-driven question-answering program that requires almost no order relationships among the tokens. As a top end for a speech understanding system, it provides a framework for exercising the bottom-end modules, where a direct neighbor context may not be available. CWIPER will be integrated with ENQUIRE in much the same way as it will be with TOPS.

During the year, several other activities of an experimental nature were undertaken. Sample speech data have been collected for each member of the project in an attempt to discover speaker normalization algorithms. Research has begun in the area of formant transitions, using an LPC formant tracker developed by the project. In the past, investigators have been tied to very inexact techniques involving analog chart output from a sonograph machine. New numerical techniques evolved over the past five years have opened up the possibility of vastly improving the state of knowledge in acoustic phonetics. Other areas of acoustic research underway at SDC are the investigation of correlates of syllabic stress and pause patterns in spoken data management language. Knowledge gained from these experiments will prove highly useful for improving overall system performance.

3.1.2 Summary

During the entire year, a need for more basic research then was originally envisioned has become apparent. In order to correct this, there has been a deviation from the original plan of moving directly into the construction of the Vocal Data Management System with a more extensive top end that would integrate a variety of components over the network. The parser was to include a control structure language for a parallel, co-routine executive. Only a limited amount of progress has been made in this area.

VOICEBOX was created as the central component of an acoustic research laboratory. SYNTHBOX was implemented to allow critical-feature-intelligibility testing. Another use for SYNTHBOX is the future incorporation of the rules pass of SYNTHBOX in CWIPER.

During the coming year, work will proceed in three areas. In the area of acoustics research, experiments will be run on various types of articulatory phenomena. The data and conclusions drawn will be documented and distributed. Besides the benefit to the SDC project, this activity will provide a much-needed source of knowledge to the remainder of the speech community and, in particular, the other members of the ARPA speech group. CWIPER construction will be another area of major activity for next year. To date, techniques for phonetic classification based upon rough filter-bank data have been developed. It remains for us to integrate the synthesis rule pass, formant tracker, and top-end logic. Of course, the data gathered by the acoustic research experiments are vital. The third major area is continuation and completion of the sophisticated top end we believe is necessary for computer understanding of continuous speech over a large vocabulary. Work in this area is progressing with the completion of the TOPS and ENQUIRE systems on the Raytheon 704.

3.2 CONVERSE: AN ENGLISH DATA MANAGEMENT SYSTEM

The goal of the CONVERSE (English Input/Output) project is to construct a conversational on-line English data management system that will permit users to manage, maintain, and query data bases with ordinary English sentences. Much of the recent work toward this goal has been oriented toward the more ambitious objective of allowing the user to speak his sentences, in addition to, or instead of, typing them into the system on a teletypewriter terminal.

An essential component of such a system is a language processor capable of carrying out sophisticated syntactic, semantic, and deductive interpretation procedures so that the user may draw on a virtually unconstrained subset of English sentences to carry out all data management tasks. The construction of such a processor has been the primary focus for the project's research.

3.2.1 Progress

The short-range objectives that have served as a focus for this year's activities include extending the prototype CONVERSE system's sentence recognition and interpretation capability, developing a detailed plan for achieving deductive inference and starting implementation in this area, and expanding experimental work with CONVERSE to include two new data base domains, one of which will serve as a common data base for the CONVERSE and Voice I/O Projects. In addition, design and study efforts aimed toward the construction of more substantial conceptual networks and syntactic/semantic and semantic/deductive interfaces have also been undertaken to ensure orderly progress toward the long-range objectives.

3.2.1.1 User Interface

Improvement at the user/machine interface has been made by continuous extension of the range of English sentences that are interpretable by the system and by the addition of new facilities for morphological analysis of inflected word endings; recognition of a much wider class of English punctuation marks, hyphenated words, abbreviations, numerical and temporal sequences, etc.; and by new routines that apply spelling-correction heuristics and initiate user feedback dialogs where appropriate.

The CONVERSE spelling-correction heuristics are an extension of the work Warren Titleman carried out in this area as part of his effort toward user orientation in BBN LISP. In the past, if a CONVERSE user mistyped a character or the I/O device misread or "doubled" characters, retyping of the entire sentence was usually required. Now, when dictionary lookup fails, the spelling corrector is invoked, and, where possible, spelling corrections and/or morphological analysis will find or create the correct word. Where this fails, a user can abort the entire sentence, reenter terms correctly, or define and substitute terms as appropriate. Input

character sequences acceptable to CONVERSE are discussed in detail in TM-4885, "A Guide to the CONVERSE Input Conventions."

Morphology

The morphological analyzer as presently implemented handles inflectional endings, including plurals for nouns, all of the regular inflections for verbs, the -ly suffix for adverbs, and the comparative and superlative suffixes for adjectives. It attempts to find a word stem that is a dictionary entry by stripping off suffixes and making various adjustments to the spelling. When a stem is found, the lexical "signature" for that word is retrieved from the dictionary and modified to reflect the semantic and part-of-speech shifts indicated by the suffix that was removed. Additions to morphological capability in the future will include treatment of more cases of currently handled inflections and suffixes, processing of multiple suffixes, and extension of the anlyzer to identify some of the frequently occurring nominalizing, verbalizing, and adjectivalizing suffixes for adjectives, verbs, and nouns.

3.2.1.2 English Recognition

The versatility of an English language processor depends upon at least two considerations: (1) the variety and complexity of the grammatical constructions for which the syntactic component can produce correct deep structures, and (2) the ability to make a semantic link between sentences or phrases that are paraphrases of each other but differ in lexical items or syntactic deep structure or both. When English is being used to query a data base, the second of these considerations assumes great importance. Whereas CONVERSE incorporates an extensive grammatical capability, paraphrasal relations have just begun to receive emphasis. Although an adequate formal linguistic theory of paraphrase still seems far off, work is proceeding in several areas that offer promise of achieving substantial improvement in this respect.

Case

The deep structures generated by CONVERSE incorporate a version of Fillmore's case grammar.[1] The case framework is particularly well suited to CONVERSE because it provides the semantic component with a link between elements of the deep structure and specific entries in the relationally organized data base. In English, grammatical case shows up primarily in specific prepositions, in the types of question words (who, what, how, etc.) and in the relative order of sentence constituents. In general, each English verb is idiosyncratic as to what prepositions, if any, it uses to signal each case relationship. Verbs also differ with respect to the case relations that may be associated with them.

---

[1] Fillmore, Charles J., "The Case for Case," in Bach and Harms (eds.), Universals in Linguistic Theory. New York: Holt, Rinehart and Winston, 1968.

While a fairly general framework of cases has been worked out in which some 30 distinctive case relationships have been identified, only those needed for current data bases are presently implemented. They are the following:

Case Name	Principal Relationship
1) Domain	logical subject
2) Neutral	direct object
3) Goal	location toward which action is directed
4) Benefactive	entity that benefits from a performed action
5) Location	where an action is performed
6) Time	when an action is performed
7) Manner	the way in which an action is performed
8) Means	the instrument or agent with which an action is performed
9) Source	location where the action originates

For example, in "Bill bought flowers for Mary", "Bill" is the Agent (Domain in CONVERSE terms), "flowers" is the Nuetral, and "for Mary" is the Benefactive of the verb "buy". Agent and Neutral cases are obligatory for "buy", while a number of other cases are optional. For instance, "from the florist" could have been part of the sentence as an instance of the Source case. If the question, "For whom did Bill buy flowers?" is asked, "For whom" is an interrogation of the Benefactive entry in the data base relation defined by the verb "buy". In general, the question words identify what case entry in a verb relation of the data base is being queried. Case analysis on nominalized forms of verbs is parallel, except that nominalizations occur in a different set of syntactic structures and the sets of prepositions for each case relation are different.

The Relational and Correspondive cases are being added because of their occurrence in the computer program data base (see Section 3.2.1.4) in phrases such as "programs relating to Matrix Inversion" and "the abstract for HICLUSTER"--Relational and Correspondive, respectively. Additional capability in case assignment is also being implemented to handle predicate nominals and nominalized forms of verbs (e.g., "Where is Fresno located?" vs. "What is the location of Fresno?").

Anaphora and Thematic Development

A comprehensive survey of the various types of anaphoric (pronomial) expressions in English was completed several months ago. For each type, the project (1) noted the regularities discovered thus far by linquists that could be exploited by CONVERSE for processing such expressions, and (2) estimated the likelihood that anaphoric expressions of that type would occur in questions or statements addressed to CONVERSE. On the basis of (1) and (2), a higher priority was assigned to certain types of anaphoric expressions, e.g., to personal, possessive, and demonstrative pronouns, possessive adjectives, and indefinite pronouns and quantifiers.

There is already a working implementation of relative and interrogative pronouns in CONVERSE. The rules currently implemented in CONVERSE for handling a few pronouns and possessive adjectives use data base responses as antecedents of following anaphoric expressions. New rules will take as input one or more semantic representations of a given anaphoric expression and test possible antecedents for semantic compatibility with it. In evaluating possible antecedents, these rules will take account of surface syntax as well as deep structure. One important use of surface structures is to enable CONVERSE to prefer a possible antecedent that exhibits surface-structure parallelism with the anaphoric expression over one that does not.

For example, in the dialog

> "Is Los Angeles larger in population than New York?"
>
> "No."
>
> "Is it larger in land area than San Francisco?"

the probability is high that 'Los Angeles' is the antecdent of 'it.' Using a computational adaptation of Harris-type discourse analysis for recognizing anaphoric and thematic relations, CONVERSE should be able to exploit syntactic (and semantic) parallelism for antecedent evaluation. Woods et al.[1] have already implemented several anaphoric rules of the type being developed for CONVERSE, and it is expected that some of them can be incorporated into CONVERSE with little change.

In contrast to the large volume of publications on anaphora in recent years, linguists have devoted little attention to thematic development, although the primary function of anaphora is to make thematic development possible. Thematic analysis exposes the communicative strategy underlying a discourse or dialog by (1) segmenting a text into passages that develop certain other passages (and that may be developed by still others in turn), and (2) labeling the type and direction of development in each case. Among the types of thematic development recognized in the as yet unpublished thematic analyses of Scientific American articles prepared by Olney et al.[2] are comparison, identification, justification, generalization, and exemplification; this very sentence is a development by exemplification of the phrase "the types of thematic development." Inasmuch as an expository strategy can usually be analyzed as anticipation by the writer of a certain sequence of questions likely to occur to his intended reader as the writer develops his theme(s), one may expect that interrogative strategies used in CONVERSE dialogs will turn out to be closely parallel to expository ones. Recent work with Olney's thematic analysis technique has confirmed this expectation.

---

[1] BBN Report Ac. 2378, p. 420.

[2] Semantic Foundations for Question Answering Project at UCLA, hereinafter referred to as the Semantic Foundations Project. This group is working closely with the CONVERSE Project.

An algorithm for thematic analysis is now being developed. Here, the Harris-type computational discourse analysis mentioned above provides one point of departure: when two clauses that are adjacent in a text exhibit a high degree of syntactic and/or semantic parallelism, it is likely that either one of them develops the other by one or a small number of development types (of which comparison is one) or that both develop a (usually) preceding passage by the same development type. Sentential adverbs and subordinating conjunctions provide more specific clues as to the type and direction of thematic development, and a semantic and distributional analysis of such words is now being undertaken. This analysis will also provide information for refining the cases assigned to embeddings by the CONVERSE grammar. Perhaps the most important aspect of the thematic analysis work is the promise it offers for enabling CONVERSE to grasp the user's interrogative strategy, at least in part, and on that basis to formulate expectations usable as clues by a speech recognition device.

Semantics

Semantics is one of the major areas of interest and research in natural language systems, now that syntactic analysis techniques are reasonably well understood and widely implemented. Either directly or indirectly, the greater part of the CONVERSE effort is directed toward improving and expanding the system's semantic capability. The fact that the CONVERSE facility presently queries structured data bases allows for some simplification of the semantics over what will eventually be required in a truly general-purpose English-language facility. Nevertheless, every effort is being made to arrive at general solutions to problems of semantics whenever possible and feasible.

A number of practical difficulties relating to semantics have already been attacked. One of these is the fact that users of natural language produce sentences having a wide variety of constituents deleted in the surface string. Restoring these deletions may involve considerable processing, so that a practical question arises as to whether it is more economical to expand the syntax rules to produce the theoretically correct deep structure or to expand the semantic rules to obtain the desired semantic information from incomplete deep structures. For example, take the nominal paraphrases of a verbal sentence:

(1) What cities are located in California?
(2) What cities are in California?
(3) What are the cities in California?
(4) What are the California cities?

Sentences (2), (3), and (4) are paraphrases of (1) but do not contain the verb "located". Routines have been implemented in the CONVERSE semantic rules that determine that "cities" and "California" qualify as entries in

the "located" relation and construct the correct Intermediate Language expression to query this relation in the data base. Also, in the case of incomplete comparative structures, the semantic solution has been largely adopted. For example, if the user asks, "What city is larger than Downey in population?", the semantic rules reconstruct the full meaning of the query, whose syntactic deep structure would correspond to "What city is larger in population than Downey is large in population?"

Another area of semantic difficulty is the lack of complete correspondence between the manner in which English correlates of logical operators (or connectives) are used in English sentences and the manner in which they are used in formal logic, especially with regard to quantification and negation. The practice of the speaker is much looser and less structured in these areas than the corresponding rigorous formalization. Rules on the syntactic level for handling "quantifier floating" are an example of progress in these areas. For example, English speakers tend to move "all" around in sentences:

(1) Do all the cities in California have smog?
(2) Do the cities in California all have smog?

In both sentences, the speaker intends for the "all" to quantify "the cities in California." The syntactic rules transform the linear order and constituent structure of the second example so that it is identical to the constituent structure assigned to the first example.

One more area having implications for semantics relates to a system user's presuppositions about the data base as he types in his queries. The semantics of the CONVERSE English processor must anticipate such presuppositions whenever possible and provide defaults for essential referential information that has been left unspecified. As a simple example, in the city data base (see Section 3.2.1.4), the user may have framed his query using the superlatives "largest" or "smallest" to modify "city" without specifying whether population or land area is the intended unit of measurement. Here, the system may either "default" to one or the other measurement or may ask the user to specify a selection from among the two.

In the course of implementing semantics for Intermediate Language-1, the project has attempted to identify significant semantic constructs that occur frequently during semantic processing at higher nodes in a syntax tree. Such constructs become important pieces of semantic information in the target (i.e., Intermediate) language expression that eventually results at the top node of the tree. This experience is proving to be of great value in the task of writing new semantic rules in terms of the new Intermediate Language (IL-2).

3.2.1.3 Concept Network and Inference

Concept Network

Since early in its development, CONVERSE has had a concept network, a network of the important concepts underlying its data base. This network provides a formal representation of the possible relations and hierarchical structures discernible among the data items. It has always been specific to a given data base, and it contains nearly all of the knowledge about its universe of discourse (the data base) that the system uses during parsing.

It is normally most economical to store data in just one way; a goal of an English question-answerer, however, is to access the same data by a varied set of queries. Thus, a network of concepts contributes to a system's ability to "understand" English queries by providing conceptual paths between the most direct and exact phrases by which the data base may be accessed, on the basis of which paraphrasal relations among these phrases can be recognized.

This approach is now being extended to general concepts as well as those for specific data bases. The underlying assumption is that there are a number of important, highly general notions that should form part of the conceptual framework of English-language systems. Such a framework would contain explications of notions such as "cause", "action", "reason", "purpose", etc. This makes the concept network intimately related to the system's dictionary. Just as the dictionary contains a core of closed-class function words, so there will be an analogous basic concept net of highly general notions that are needed for virtually all universes of discourse.[1] The CONVERSE research has revealed versions of these notions embedded in the structure of English. For example, "reason" is both a thematic relation and a case relation (of embeddings); so are "cause" and "purpose". "Cause" is directly involved in an affixal relation expressed by several suffixes, perhaps most characteristically by '-ize'. "Action" is directly involved in an affixal relation expressed by '-ion'. The version of this notation that turns up as a case relation is "agent"; its thematic counterpart is "procedure."

---

[1] Winograd notes that "saying that a structure is a 'network' is not much more information than saying that it is represented by bits in a computer memory." ("Procedures as a Representation for Data in a Computer Program for Understanding Natural Language," MIT Project MAC Report TR-84, p. 407.) We should therefore point out that by 'concept net(work)' we mean a structure in which the nodes represent concepts (i.e., terms specific with the respect to sense) and the labeled links connecting them stand for axioms taking the form of atomic sentences and falling into some small number of basic types. Additional axioms, more complex in form, are attached at the nodes to which they directly pertain. The primary use of the concept network is to facilitate logical inference.

In addition to these highly general notions and the specific terms needed for a particular data base, the concept network must contain terms of intermediate generality so that the specific terms can be characterized appropriately, e.g., to facilitate the deduction of sentences in which they occur. Thus for the new data base of programs (see Section 3.2.1.4), the specific terms include "compiler" and "random-access storage," with respect to which "convert" and "equipment" are terms of intermediate generality. The following paragraphs review progress made during the year toward constructing a concept network of terms at all three levels for the computer program data base.

Highly general terms. For several years, the Semantic Foundations Project has been collecting the recent literature of analytic philosophy for conceptual analyses of the keywords of the major defining formulas in Webster's Seventh (W7), e.g., "cause" in the formula "to cause to", "action" in the formula "the action of", etc. Since these analyses are usually given in the context of particular philosophical problems, it is necessary to generalize each analysis and adjust it so as to eliminate mutual incompatibilities with others before entering nodes and links in the CONVERSE concept network that amount to partial representations of it. These steps have been completed for 10 highly general notions.

Specific terms. Definitions for several hundred technical terms relating to computers and computer programs have been selected from technical glossaries and dictionaries.

Terms of intermediate generality. In the course of selecting the specific terms, the project staff noted all terms of intermediate generality occurring in their definitions; 300 such terms were identified. At the same time, the Semantic Foundations Project was about to obtain all W7 definition contexts for each keyword of the major defining formulas, together with its inflected forms and most of its affixal derivatives. Because such contexts can contribute significantly to the semantic analyses of the 300 terms of intermediate generality, we arranged for them to be included in the context searching run. This run yielded nearly 150,000 definition contexts for a little over 1,800 word forms arranged in the style of a KWIC index. Inspection of the definition contexts for a sample of the 300 terms of intermediate generality indicates:

(1) These contexts provide a useful supplement to the semantic-field data that can be obtained for each of the 300 terms from its definitions in Webster's Unabridged, consulting thesauri and dictionaries of synonyms, etc.
(2) These contexts provide a start toward identifying the relevant portions of the definitional scope of each of the 300 terms, separately for the several senses in which it is used in W7 definitions. The contexts also aid in the process of identifying the relevant portions of the definitional scope of each of the highly general terms included in the search (in virtue of being keywords of major defining formulas).

In both cases, the contexts indicate the words for which definitional contexts should be obtained on the next pass through the W7 transcript, so that the full definitional scope of each original term can eventually be ascertained. In most cases, straightforward machine processing of the contexts, coupled with a quick visual scan, should suffice to identify the relevant portions of the definitional scope of a given term.

Inference

CONVERSE currently employs several kinds of built-in inferential capabilities to facilitate semantic interpretation and question-answering. For example, CONVERSE uses class-inclusion logic to block interpretations that would be syntactically correct but meaningless; for example, "Who is mayor of Denver?" will be blocked for the county of Denver but not for the city. In answering complex questions, CONVERSE must generate procedures that compute new set extensions from existing set extensions, as in the question, "What cities are in states that have a population less than that of the city of Boston?" Here it is necessary for the system to infer which states have a certain property (viz., the property of having a population smaller than that of Boston). However useful and powerful such inference is, it is strictly special purpose and in some cases ad hoc. The need is to provide a more general deductive inferential capability, one which can be brought to bear in many different contexts and for different purposes in semantic interpretation and question-answering.

The important concerns here are (1) selection of relevant general premises from among a very large number of possible general premises, most of which are completely irrelevant to any particular problem at hand; and (2) efficiently using a large mass of particular assertions, as opposed to the highly general statements with which mathematical inference is almost exclusively concerned. In the question-answering case, the derivations that are constructed once relevant premises have been discovered are likely to be relatively straightforward and direct. Consequently, the first task is to provide CONVERSE with the capability of discovering relevant premises within a reasonable amount of time and at a reasonable cost.

The inference system being designed for CONVERSE first constructs preliminary, skeletal derivation proposals. The purpose of these proposals is to find possible proofs (deductions) before any attempt is made to verify the proposals. Verification is thus delayed until overall proof plans have been established. Later phases of processing examine the variable flows within a proposal to detect possible collisions and search the fact file for compatible sets of values for instantiation.

The construction of proof proposals will be accomplished by a proof proposal generator (PPG). The PPG is the main driver of the CONVERSE inference system. It examines the input question to determine the assumptions and goals involved;

calls upon the two main components that indicate possible middle-term predicates and premises--a chain generator and an advice file; integrates this information and constructs a proposal using it; and decides whether a proposal is complete or further deduction is needed.

The chain generator is now being programmed. It attempts to find middle-term paths linking the assumption predicates and the goal predicates. It uses a predicate connection graph, which is constructed initially before any questions are input to the system, to find chains of predicates indicating a possible implication chain. The predicate connection graph exhibits the conditional interactions between predicates within the premises in a premise graph (the file of all general and molecular empirical statements). Associated with each link within the PCG is such information as what predicates are involved, the premises in which they occur, and the type and degree of conditionality among them. The PPG will extract the premises with which links in the chain are associated and start constructing a proof proposal. It will then examine these premises to find those predicates not involved in the chain and decide whether they should be deduced or await later searching of the fact file for possible instantiations. The decision will be based on the extent to which a predicate's extension is complete, the difficulty of deducing a particular predicate, etc.

The advice file is the second component used by the PPG to find relevant middle-term predicates and premises. The file will contain statements specifying the conditions under which the advice should be [illegible] and the actions to be taken by the PPG if the conditions are met. The actions will mainly indicate the use of predicates or premises as middle terms when certain conditions, such as the occurrence of a particular predicate as a goal or the occurrence of a particular constant in the problem, are met. For example, an advice statement may look like this--"if connecting the predicate NORTH to the predicate SOUTHWEST, use the premise NORTH (X,Y)⊃SOUTH (X,Y)." It will be the function of the PPG to use this advice in conjunction with the chains output by the chain generator to construct the proof proposals.

Although the deductive module is being built to interface with CONVERSE, its initial development will be somewhat independent of the existing CONVERSE system. Interfacing will depend on the development of IL-2, on the programming of a preprocessor that constructs the data structures (premise graph, predicate connection graph, etc.) needed for deduction, and on further development of the concept network. Further discussion of the present approach to deduction can be found in SP-3679, "Inferential Question-Answering: Extending CONVERSE."

3.2.1.4 Data Base Implementation/Experimentation

A critical area of activity during the past year has been to enhance the adaptability of CONVERSE to significantly different domains of discourse. Several demonstrable English question-answering systems have worked well for a single, small, tightly circumscribed domain; CONVERSE, on the other hand, has been designed from its inception to adapt to varying domains of discourse, much as operational data management systems are designed to do. During the year, the existing data base of California and New York cities was extended, and two new data bases--one of English sentences and one of computer program descriptions--were implemented and used for experimentation.

City Data Base

During the year, extension and exercising of the city data base was completed. This data base has grown to encompass some 14,000 facts concerning the cities in California and New York. A fair idea of the question-answering scope for this data base is illustrated in the following dialog:

```
HOW MANY CITIES ARE THERE?
    123
HOW MANY ARE THE EASTERN CITIES?
     37
GIVE ME SOME OF THE EASTERN CITIES!
 NAME
 ALBANY
 AMSTERDAM
 AUBURN
 BINGHAMTON
 BUFFALO
     MORE?
YES
 ELMIRA
 FREEPORT
 HEMPSTEAD
 ITHACA
 JAMESTOWN
     MORE?
NO
```

```
FOR THE WESTERN CITIES, WHAT IS THE POUPLATION AND MEDIAN FAMILY INCOME?
     DID YOU MEAN POPULATION ? (spelling correction routine at work)
YES
 NAME POP-60 MEDIAN-FAMILY-INCOME-59
 ALAMEDA 63855 6188
 ALHAMBRA 54807 7185
 ANAHEIM 104134 7625
 ARCADIA 41005 9526
 BAKERSFIELD 56848 6541
     MORE?
NO
IN WHICH STATE IS FRESNO LOCATED?
     CALIFORNIA
DO ANY CITIES HAVE A POP GREATER THAN LA'S POP?
     YES.
WHICH CITY HAS THE LARGEST POP?
     NEW YORK
WHAT IS NY'S POOP?
     DID YOU MEAN POOR?
NO
    MORE?
     POOP?
?    (user's call for help)
     IF YOU TYPE ZERO OR MORE WORDS FOLLOWED BY A "←" I WILL REPLACE TH
E WORD ABOVE WITH WHAT YOU TYPED BEFORE THE "←".  IF YOU TYPE "!" I
WILL ABORT THE SENTENCE.
     POOP?
POP
     DO YOU MEAN 1) NEW-YORK-CITY OR 2) NEW-YORK-STATE?
     TYPE "1!" OR "2!".
1!
     THE POP-60 FOR NEW-YORK-CITY IS 7781984
```

```
WHAT IS THE TOTAL LAND AREA OF THE SMOGGY WESTERN CITIES?
      790.70
WHICH CITY HAS THE LARGEST MEDIAN FAMILY INCOME?
      BEVERLY HILLS
WHICH WESTERN CITIES AREN'T SMOGGY?
 NAME
 ALAMEDA
 ALHAMBRA
 ANAHEIM
 ARCADIA
 BAKERSFIELD
      MORE?
YES
 BALDWIN PARK
 BELLFLOWER
 BERKELEY
 BEVERLY HILLS
 BUENA PARK
       MORE?
NO
WHAT ARE THE SMOGGY CITIES?
 NAME
 LONG BEACH CAL
 LOS ANGELES
 OAKLAND
 SAN DIEGO
 SAN FRANCISCO
       MORE?
NO
```

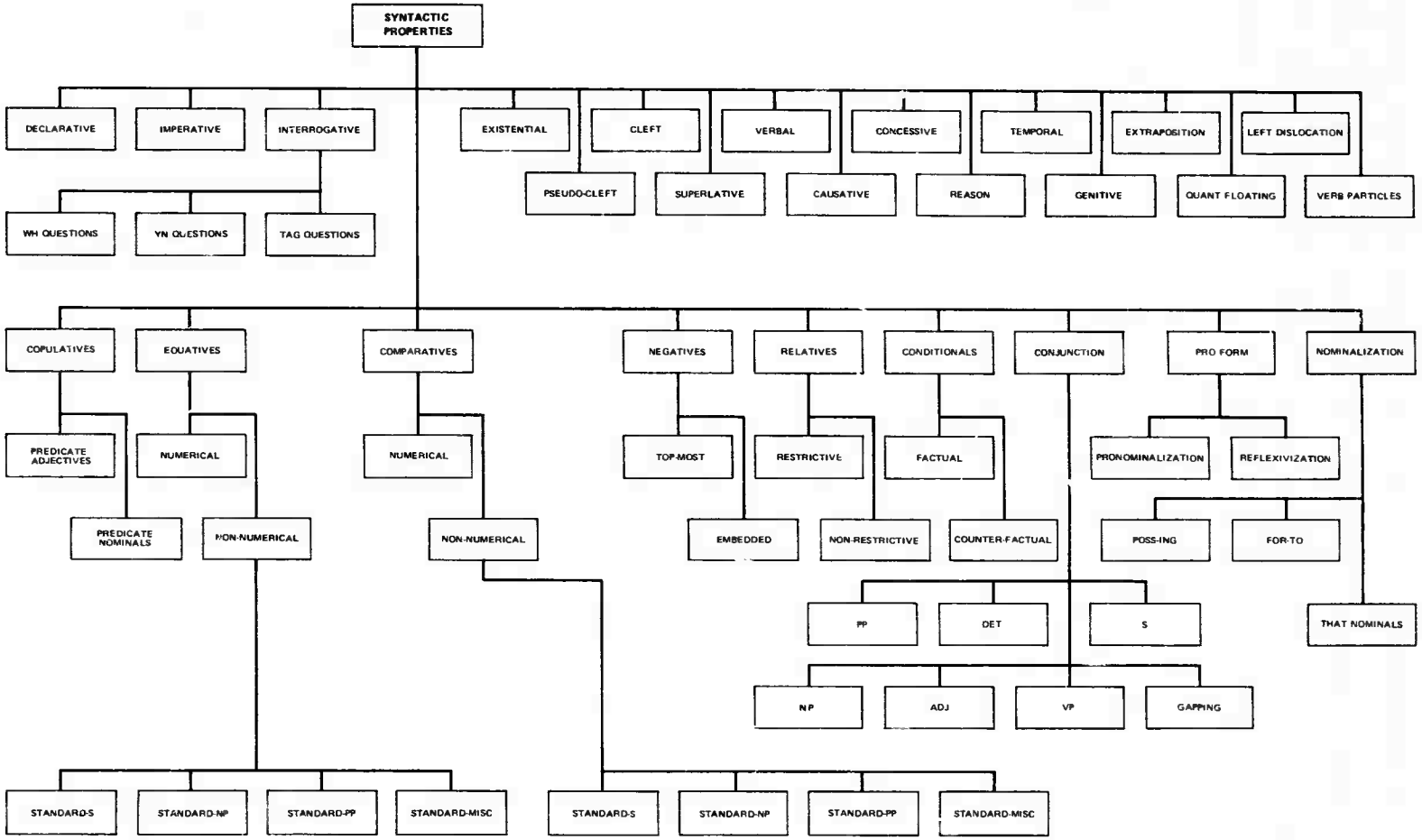

Figure 3-4. Syntactic Properties for Sentence Data Base Indexing

Sentence Data Base

The sentence data base consists of about 4,000 facts characterizing the syntactic structure of some 700 sample English sentences that are cross indexed according to the system of syntactic properties illustrated in Figure 3-4. There were two reasons for implementing such a data base: (1) to aid in the checkout of the CONVERSE English subset grammar, and (2) to yield experience in attacking a new and distinctively different data base domain.

Simple requests to the sentence data base, such as "What are the yes/no questions?" or "How many sentences contain neither restrictive relatives nor for-to nominals?" lead to tallies or printouts of the requested subset. Alternatively, in an imperative mode of operation, the system accepts commands such as "Parse the negative tag questions!" or "Answer the WH questions that are copulative!", derives the appropriate subset of sentences, then proceeds to pass each sentence as a character string to the parser. In the case of a "parse" command, the surface and deep structures are printed out for each sentence. The "answer" command causes each semantic interpretation in intermediate language (IL) to be evaluated against the data base, thus leading to a printout of all relevant question-answer pairs. This data base has proved helpful in the checkout and validation of the current CONVERSE grammar.

Computer Program Data Base

The computer program data base contains information characterizing a set of 500 computer programs. This data base was obtained from the National Program Library and Central Program Inventory Service for the Social Sciences (NPL/CPIS) at the University of Wisconsin. The information it contains includes descriptions of programs, routines, and systems used in statistics and social science data processing, as well as the programming languages and computers they employ. In addition, there is a large class of keywords that have been used to characterize the programs. The contents and use of this data base are detailed in TM-5015, "Question Answering in the CONVERSE System."

Data Base Experimentation

Implementing and experimenting with these three data bases has revealed both weaknesses and strengths in CONVERSE's ability to adapt to new domains of discourse. In sentence recognition, for example, only relatively minor changes have been necessary in the surface and deep structure rules. Changes at the semantic interpretation rule level, however, have been much more extensive, and there is clearly some distance to be covered before a system configuration in which semantic rules remain largely invariant with respect to changes in data base domain can be achieved. In the data management area, fact retrieval continues to operate quite fast--on the order of 100-200 milliseconds for

elementary requests from data bases of 10,000 to 20,000 facts. Evidence continues to suggest that the CONVERSE set-theoretic data structure has a number of advantages for natural language data management. However, the current implementation does suffer from some constraints in LISP (maximum array size, maximum number of floating-point numbers allowed, etc.) and in the expressive power of Intermediate Language-1. These limitations will be overcome in the implementation of IL-2 and the reimplementation of the basic data management routines in a lower-level programming language.

3.2.1.5 Speech Understanding Support

The CONVERSE Project's efforts to support the ultimate construction of a speech understanding system range from near-term linguistic support to the long-range design and construction of advanced language processing modules. During the year, the CONVERSE staff provided the Voice I/O Project with a copy of a simplified keyword question-answerer, ENQUIRE, ...that makes use of semantic clues in answering questions of a set of properties defined over a domain of objects. The Voice I/O Project is currently using ENQUIRE in support of its work on predictive linguistic constraints. ENQUIRE is described in a forthcoming document, "Question-Answering by Keyword Search."

After several discussions about possible common semantic domains of discourse, the two projects decided upon the domain of computer program descriptions as a common data base for joint development and research. This domain will permit experimentation with semantic/syntactic constraints that range from the fairly simple--for sets of keywords describing a collection of programs--to the quite complex program domains that include complex relations between programs and their subroutines, types of acceptable input data, space and timing information, etc.

For the longer term, almost all of the current work in syntax, semantics, and deduction is aimed at the ultimate achievement of a speech understanding system. Such a system will require sophisticated semantic, deductive, and syntactic support of the kinds now being developed. A collaborative effort in morphology will begin to shed some light on syllable-juncture phenomena and provide a basis for affixally signaled word-sense extension.

3.2.2 Summary

The CONVERSE Project is constructing an English Data Management system with the eventual goal of a vocal as well as a teletypewriter interface. In order to provide maximum utility to a user, such a system must ultimately offer an English processing and sentence understanding capability that is considerably in advance of present-day language processing technology. To achieve this, the project has created a research prototype natural-language data management system and implemented various data bases in it in order to find out what factors limit its performance, and is now introducing new language processing modules or improving existing modules in order to extend the system's versatility, question-answering capability, and efficiency.

During this year, the project advanced in a number of areas, from the practical to the conceptual. At the user/machine/problem-domain interface, techniques were implemented to facilitate the user's composition and communication of requests. This permits a user to type in his request freely and quickly, realizing that the system will, wherever possible, detect and correct his errors in typing and spelling and, when necessary, allow him to replace or define new or grossly misspelled terms. The new input-character sequence-recognition conventions implemented this year greatly widen the scope of interpretable symbols to include abbreviations, hyphenated words, dates, times, quotations, arithmetic operators, and other special symbols. User dialog has been enhanced in numerous small but important ways to provide increased user guidance and direction in cases where ambiguities must be resolved or help is requested. Many new messages have been added to the system to indicate, when possible, just how the scope of the English subset or range of the data base domain has been exceeded.

Another area of important experimental activity this year has involved the implementation of two new data base domains and the continued exercising of the system with these new domains. This activity has led to a better understanding of the components of CONVERSE that currently limit or make difficult the system's adaptability to new data base domains. For example, while only minor changes were required in the sentence structure recognition rules in moving from one domain to another, extensive changes were required in the semantic interpretation rules, and considerable time was necessary to construct new dictionary entries and concept network linkages. Also, while fact retrieval remains fast even for fairly substantial data bases, the current routines for data base generation and creation need to be improved both in speed and in range of application.

In English syntax, a number of new rules were added, and the operation of many others was improved. In addition, the project added a morphological analyzer, implemented a first-version anaphoric (pronoun or pro-verb) processor, and developed an approach to further work in this area and in the related area of thematic analysis. Efforts to extend question-answering power and improve vertical communication between syntactic and semantic processors produced significant progress in firming up specifications for IL-2, in generating a comprehensive design specification for a deductive component, and in beginning the implementation of deduction and premise selection and retrieval components.

During the coming year, the project will shift the emphasis of its efforts in intrasentential syntax from the addition of new rules to consolidation and increased parsing efficiency and improved syntax/semantics interaction. Work on intersentential syntax (anaphora, thematics) will expand, as will work on productive word-sense extension via affixal processes. These efforts are directed towards providing essential expanded syntax/semantic support for improved speech and language understanding processors.

3.3 STAFF

Voice Input/Output

J. A. Barnett, Principal Investigator
W. A. Brackenridge (part-time)
R. DeCrescent
R. A. Gillmann
Iris Kameny
L. M. Molho
D. Pintar (part-time)
H. B. Ritea

CONVERSE

C. H. Kellogg, Principal Investigator
J. H. Burger
T. C. Diller
K. J. Fogt
A. Leal
R. V. Weeks

3.4 DOCUMENTATION

Barnett, J. A. "A Vocal Data Management System." Conference Record, 1972 Conference on Speech Communication and Processing. Air Force Cambridge Research Laboratory AFCRL-72-0120, Special Reports No. 131, 22 February 1972. pp. 340-343.

Barnett, J. A. The VOICEBOX Interactive System. SDC TM-5002, 19 September 1972.

DeCrescent, Ron. The Necessity for Standardizing Speech Recording Environments. SUR Note 29, NIC 9441. 3 April 1972. ARPA Network Information Center, Augmentation Research Center, Stanford Research Institute.

Kellogg, C. H. Question Answering in the CONVERSE System. SDC TM-5015, 17 October 1972.

Leal, A. A Guide to the CONVERSE Input Conventions. SDC TM-4885, 15 March 1972.

Ritea, H. Barry. A Comparison of FFT Algorithms. SDC TM-4857/100/00, 5 January 1972.

Ritea, H. Barry. A Survey of Digital Signal Processing Techniques for Speech Analysis: Table of Contents for Document Series TM-4857. SDC TM-4857/000/00, 5 January 1972.

Ritea, H. Barry. SDC Speech Bibliography. SUR Note 18, NIC 9081. 16 February 1972. ARPA Network Information Center, Augmentation Research Center, Stanford Research Institute.

Ritea, H. Barry. System Description and Data Representations for the SDC Speech Understanding System. SUR Note 7, NIC 8261. 9 December 1971. ARPA Network Information Center, Augmentation Research Center, Stanford Research Institute.

Ritea, H. Barry. The Cepstrum, The Cepstrally Smoothed Log Spectrum and the Chirp Z-Transform. SDC TM-4857/200/00, 5 January 1972.

Ritea, H. Barry. The Folding Quefrency of the Cepstrum. SDC SP-3632, 5 January 1972.

Travis, L., Kellogg, C. H., and Klahr, P. Inferential Question Answering: Extending CONVERSE. SDC SP-3679, in press.

4. ON-LINE GRAPHIC COMPUTATION

Enhancing communication between man and computer by providing facilities compatible with techniques used in visual man-to-man communication is the principal goal of the Graphic Input/Output (I/O) Project. Our concern has been to develop methods of graphical interaction that permit a user to carry on a dialog with a computer in the language and notation of his discipline or problem domain. The functional entities required for such a dialog are a data-input tablet (e.g., sonic pen), an interactive display operable as a terminal in a time-sharing system, and a programmed facility to accept and interpret graphic input and to create a meaningful graphic output.[1]

4.1 PROGRESS

In this reporting period, work has been focused on three areas: extension of the existing graphic input/output facility and programming system, TAM (The Assistant Mathematician);[2] identification and definition of a specific problem application area in which adaptation and extension of TAM would provide a means for (1) relaxing expressional constraints and (2) performing computations over a data base, improving the ease and flexibility with which a researcher communicates with the computer; and conduct of an ARPANET graphics experiment with Massachusetts Institute of Technology as a step in making the system and its components more widely available.

TAM Extensions

The capabilities of TAM were extended by including a dynamic dictionary-building facility that allows the user to add new entries and make corrections to his dictionary while running the TAM system. Previously, he had to do his dictionary construction and updating outside the TAM environment. The interface between the dictionary builder and the parser/unparser forced extensive reorganization of the data buffers, and some existing routines had to be modified to be compatible with both tasks.

As a result of continued use of and experimentation with the TAM system, extensive changes were made in the parser logic to reduce the frequency of errors. The matrix processor was rewritten to handle nested fractions within matrices. The modules that examine the positional relationships between a character and potential subscripts and superscripts were changed to reduce the number of subscripts erroneously identified. Other relationship changes

---

[1] Bernstein, M. I. Hand-Printed Input for On-Line Systems. SDC document TM-3937. April 1968.

[2] Bebb, Joan. The Assistant Mathematician (TAM). SDC document TM-4790/000/00. October 1971.

improved the handling of characters that normally extend below the line of printing, such as 'g'. The matrix analyzer was rewritten to avoid confusion between a matrix and a fraction within brackets that has a fractional numerator and/or fractional denominator. These changes have greatly improved the reliability of the parser.

The main need in making these extensions to TAM was to accommodate larger matrices and to extend TAM's functional capabilities. Since operating system capacities relative to a single overlay program had already been reached, TAM was split into two overlayed segments--the interpreter and the parser/unparser. This necessitated several major changes to the driver and loader programs to implement the interface between overlays; at the same time, the loader was upgraded to be compatible with an improved version of CWIC output. To minimize swapping time, multiple outputs (from iterated output statements) are now accumulated in the interpreter before going to the parser/unparser segment.

Application Area Selection

The criteria for selecting an application area for a TAM-like system included the applicability of interactive man-machine techniques to some of the computational processes done and the usefulness of the area as a basis for the data base computation and constraint-relaxation effort. To this end, the general area of signal and image processing was selected. Another area considered and rejected was analysis of weather data. Weather data provide an extensive data base, but little experimental or interactive computation is required. On the other hand, because image and signal processing involves extensive computation over large data sets, it can serve as a base for both the data base computation and constraint relaxation study. At present, in most research centers, computational work in signal and image processing is oriented to batch processing. The application of interactive techniques to the tasks required to process image data will, we feel, introduce a synergism to image and signal processing experiment design by allowing human experimenters to dynamically determine the points in the processing at which either they or the computer system should assume control of the processing strategy. In addition, the human can select processing strategies according to known characteristics of the image data and can terminate a strategy when it is clear to him, although not necessarily to the computer system, that the strategy has failed.

It was also determined that the field of image processing is the proper context (in that it is notationally rich) in which to investigate the relaxation of constraints on expressional variety and data handling by introducing a domain-definition capability that frees the image-processing researcher from the need to keep track of historical and definitional characteristics of images and their computational products (such as image transformations). Our initial program in this area will be oriented toward two-dimensional filtering and convolution techniques. The problem definition led to a system specification for an interactive image processing system that will incorporate facilities necessary to achieve the anticipated synergism.

The Image Processing System

Image processing research is an endeavor in which the computational environment allows a proper combination of mathematical description of analytic processes and pictorial representation of entities and functions best described graphically. Image processing is a combination of graphic and mathematical processing techniques. Some of these techniques are well-defined and accepted, such as convolution or Fourier transformation; these, in a proper system, should be basic system operators. Other mathematical processes, such as alternative transformation techniques, are being developed by researchers in the field; these are amenable to description in two-dimensional mathematical notation. Most importantly, some processes currently defined analytically are better done graphically, filter specification being a primary example. Specification of a symmetric filter as an analytic function is, in some cases, an approximation to achieve a desired curve shape. Certainly, drawing the curve desired and allowing the computer to generate the function values is a superior technique. Currently, specification of asymmetric filters of any great size is a grueling task because values must be input tabularly; graphically, bounded areas with given values may be described rapidly and accurately.

Interactive manual assistance facilitates pattern detection and phenomena isolation. Picture thresholding using analytic specification is a more flexible technique than the traditional subroutine call with circumscribed parameters; a contrast-enhancing function input as a graph allows, again, curve description not generally easy to specify either analytically or tabularly.

The image processing system specified in rough form during this year provides the facilities necessary to accomplish step-by-step convolution, contrast enhancement, and filtering of images. It will maintain a catalog of images and their computational products and provide an automatic historical trace of processing done to any catalog member. All images and their computational products are uniformly regarded as real or complex valued matrices; any system facility is equally applicable to any entity known to the system. The system is directed using two-dimensional mathematical notation incorporating a powerful set of operators and the matrices as operands. Standard image processing techniques such as Fourier transformation, spatial convolution, and point-by-point multiplication, division, addition, and subtraction are included as basic system operators. A typical system directive might be

$$D \leftarrow A \circledast B + C$$

which means convolve, in the space domain, the image A with the filter B; add C to the result and create the output entity D. All information relative to the physical descriptions of the entities A, B, and C is contained within them; the new entity, D, is created with derived appropriate physical characteristics

and the historical information that it is the convolution of A and B summed with C. This history, in fact, is created to include the genesis information from A, B, and C.

The system includes a set of analytic tools that allow the image-processing researcher to dynamically interrogate the intermediate results of the process. Among these tools are histograms, linear traces, and direct pictorial displays of matrices being generated. These displays may reflect the magnitude or phase, real or imaginary part of any matrix; all conversions between coordinate systems are automatic. The user may control the content of histogram or direct image displays by citing value ranges to be included or excluded. This thresholding is specified by analytic expressions of the form

include 90% < x

exclude .1 < sin(x) ≤ .3, x < 2000

The first will include the smallest 90% of matrix values, excluding the largest 10%. The second will exclude values whose sine is greater than the value .1 but less than or equal to .3 and values greater than 2,000; all others will be included.

The diversity and flexibility with which filter matrices may be specified are currently this system's greatest strength. Matrices can be described as curves written on system-provided grids or as sequences of mathematical functions. Either asymmetric or symmetric matrices may be created using graphic techniques. Symmetric matrices may have circular or elliptical cross-sections. The user may draw his curves with no pictorial background displays or he may request a background. With backgrounding he may update a derived linear trace through an existing matrix, creating a new, modified matrix, or generate an asymmetric filter derived from a display of unfiltered data.

System implementation will be carefully directed to allow easy amplification and extension of facilities. Effort is being expended to make components modular and separable, following the design of the existing graphic facility. The system will provide an important first step in achieving a meaningful combination of image processing and interactive graphic techniques.

ARPANET Graphics Experiment

Throughout this period, we continued to work on an ARPANET graphics experiment with MIT. Our programs were checked out by connecting two terminals at SDC through the ARPANET and running our programs locally. We have succeeded in getting a good connection with MIT and have passed some data over the link; however, owing to errors at all system levels and with the individual programs, we have not yet been able to use the character recognizer.

4.2 SUMMARY

In prior years, the near-term goal of the Graphic Input/Output Project has been to develop programming systems that utilize two-dimensional alphanumeric and symbolic notation in an attempt to use the language of mathematics, the most ubiquitous scientific notation, for numeric and symbolic programming. This goal has been achieved; but, in itself, it is only the first step in enhancing man's use of computers to solve context-dependent problems. Therefore, the goal was amplified this year to combine such a mathematical language with an extended pictorial facility, including cartesian or polar coordinate graphs, as a means of describing input functions and range conditions pictorially to augment a mathematically described function. The goal is to develop a contextual environment in which things best described mathematically are so written, but things better delineated pictorially are easily drawn.

4.3 STAFF

Joan Bebb, Principal Investigator
D. E. Albrecht
H. L. Howell
Jean Igawa

4.4 DOCUMENTATION

Bebb, Joan. Operators Guide to TAM. SDC TM-4846, 10 December 1971.

Igawa, Jean. Hand Printed Character Recognizer: Final Report for Phase V. SDC TM-4915, 11 April 1972.

Igawa, Jean. User's Guide for the 4πEP Character Recognizer. SDC TM-4900, 2 May 1972.

Saylor, Jean. An Interactive Two-Dimensional Programming Language. SDC SP-3634, September 1972. (Presented at the 1972 SIGPLAN Symposium on Two-Dimensional Man Machine Communication, Los Alamos, New Mexico.)

Williams, Thomas. An On-Line Two-Dimensional Computation System. SDC SP-3640, 22 February 1972. (To be presented at the 1972 Fall Joint Computer Conference.)

5. NETWORK RESEARCH AND DEVELOPMENT

The goals of the Network Research and Development projects are to make SDC's time-sharing facility a viable part of the ARPA Network (ARPANET) and to explore ways of utilizing, as well as contributing to, the ARPANET's resources. Efforts are concentrated in three major areas: (1) evolutionary changes to the time-sharing system itself to adapt it to changing user requirements, (2) implementation of the various levels of Host-Host ARPANET protocols, and (3) research into techniques for making it possible for the various members of the ARPANET to share data residing in existing data management systems.

During the year, a significant step was taken in the evolution of the time-sharing system--the SDC-funded development of the Interactive Common Operating System (ICOS), which combines the ADEPT and TS/DMS time-sharing functions into a single system. Although the system changes caused some operational perturbations, the net long-term benefits for our ARPA research program will be of considerable value.

5.1 SYSTEMS RESEARCH

5.1.1 Progress

During the year, Systems Research Project personnel served dual roles, involving both ARPA-funded activities and the SDC-funded ICOS development. ICOS has been, and will continue to be, of considerable value to our ARPA activities, since it minimizes the overhead costs of maintaining an operating system and, at the same time, maximizes SDC's availability as a node in the ARPANET. In addition to providing a vehicle whereby user needs can be "load-leveled" across the daytime period, ICOS will bring about a reduction in the ARPA funds associated with time-sharing maintenance and support. Enhanced system capabilities have also been provided, including a disk IPL and SAVE function, better disk-pack mount procedures, and a System Manager program that enables one to dynamically monitor and control user activity and reassign system resources in response to unique operating circumstances.

The most visible aspect of the ARPA contract usage of ICOS was a 2-1/2-hour increase in our prime-time accessibility, with future increases being likely as additional non-ARPA time-sharing users help to fund even longer daily periods of ICOS operation. These longer periods of network accessibility were combined with improvements in the scheduling and software paging methods for our ARPANET interface, and network performance within the time-sharing environment has been significantly improved as a result. Users were encouraged to use the ARPANET system, both to further test our network implementation and to gain experience with ARPANET usage. Scenarios were developed for several of the available programs, and considerable experience was gained with both TELNET and TIP access to these programs, with up to five simultaneous ARPANET users being supported.

5.1.2 Summary

The ICOS operating system, which was developed under SDC funding, contains most of the system innovations that were planned by the Systems Research Project for the contractual year. ICOS will provide our research projects with greater reliability and a longer operational day, and will meet their requirements for flexibility and change. The planned Remote Job Entry (RJE) facility was not implemented in ADEPT because of the ICOS implementation effort. An RJE capability will be implemented during the coming year and will allow ARPA users to have background production work automatically enqueued and operated during the late evening hours when system use slackens.

A system tuning effort will take place this next year to improve system operation for the large CONVERSE programs. It will be necessary to continue the DDP-516 and IBM 370 executive support activities for the CONVERSE, ARPANET, and Graphic Input/Output projects to maintain high system reliability and provide close operating-system and research liaison for the implementation efforts of these projects.

5.2 ARPANET ENGINEERING

5.2.1 Progress

The ARPANET Engineering Project was affected not only by the change in the operating system but also by changes in the protocol specifications, the Host machine, and, recently, the project personnel. Despite this, the project was able to develop operable TELNET and Network Control Program (NCP) systems that allowed us to connect to several other ARPANET nodes, including UCLA, UC Santa Barbara, and MIT. Some graphics data were passed over the MIT connection.

Several modifications were made to our TELNET implementation, both to add tutorial and other user-convenience features and to make it operate more efficiently. Although we had anticipated that the file-transfer protocol implementation would also be completed during this period, the protocol specifications have not been available. We have participated in the Network's efforts to produce the specifications via attendance at the Network Working Group Workshops, the meeting to discuss the new IMPSYS flow-control procedures, and the RJE protocol meeting at UCLA. Since a finalized protocol was not available for implementation, we concentrated on modifications to our present system that will be necessary for handling large data transfers. Since we anticipated that we would implement the initial protocol utilizing LISP (as we had the TELNET protocol), we added a symbolic capability to our LISP system. The resulting order-of-magnitude improvement in I/O transfer speed also significantly benefits other LISP users, such as the CONVERSE system.

Several difficulties were encountered in the TELNET protocol implementation and were isolated after detailed and extensive debugging. For example, the fact that some Hosts did not echo CLOSE commands caused problems in purging Network table entries when our system had closed the connections but the echoed CLOSE was not received from the foreign Host. A second and more severe problem was encountered when external users attempted to log in from a system, such as a TIP, that sends an ALLOCATE of less than one line of characters. This problem was solved by modifying the code for putting to sleep and awakening the proper EXEX component, particularly LOGIN. Further changes were required to reflect the revised ICOS system tables. The Network functions are currently very near operational under ICOS.

The integration of a second Host computer, the Raytheon 704 of the Voice I/O laboratory, was redirected to include an intermediate processor as an "intelligent" interface. A PDP-11/05 was selected for this function because of its highly cost-effective performance and the potential availability of existing designs for the hardware and software interfaces to the Network. We have been working with DEC, the University of Illinois, and NASA/Ames in this regard and are currently in the process of procuring the necessary system hardware.

5.2.2 Summary

The implementation of the ARPANET protocols continued during the year, with debugging, improvements, and experimental use being made of our NCP and TELNET programs. The implementation of the file-transfer protocol was delayed owing to the lack of a definitive protocol specification, but other internal system changes were made to make the transfer of large files more efficient. The development of the file-transfer protocol will be performed during the next year, and development of the NCP and the TELNET protocol, and the integration of the Voice I/O Laboratory into the Network, will continue.

5.3 COMPUTER NETWORK DATA SHARING

The Computer Network Data Sharing study is addressing the task of making it feasible for data to be shared among the data management systems interconnected by a computer network, assuming that the systems employ different query languages and manipulate data that are structured in a variety of dissimilar forms. The object of the study is to integrate these systems in a non-revolutionary manner--that is, without requiring the redesign or reimplementation of existing systems and languages.

From among several approaches to data sharing, the "integrated" approach, which employs a network-wide common data-sharing language and translation interfaces associated with the various data management systems available over the network, has been chosen for experimental implementation. Under this approach, a request in the common language would be translated into a target data management system's language, and the target data management system

would perform the necessary functions on its data.

The most natural common language for the user would, of course, be English, and use was made during the year of an early version of the CONVERSE system component that analyzes English sentences and compiles them into statements in a formal Intermediate Language (IL). This IL can then be regarded as the actual language in which data requests are passed to a target data management system. Because an IL and the target data management language are both formally well defined, it is possible to implement the translators automatically by means of a metacompiler such as SDC's CWIC. A detailed description of the integrated approach is reported in SDC document SP-3655, "Data Sharing in Computer Networks."

5.3.1 Progress

The Data Sharing Project's goal for this year was to investigate the properties of the CONVERSE IL and to develop translation modules for specific data management languages. The first task was to build a translator from IL to the language of SDC's DS/2 commercial data management system, using CWIC. The construction of this translator was relatively simple; it required only about three man-months. A difficulty encountered in trying to match data elements represented in the CONVERSE data base with the corresponding elements in the DS/2 data base was resolved with the aid of a table that describes the correspondence between data elements in the two systems.

The overall investigation of the version of IL that was used for this year's study indicated that it lacks several features that are necessary for the integrated approach to data sharing. Principally, it does not organize syntax in as well-structured a form, consisting of qualifier, replacement, and output, as is desirable. Consequently, a new Intermediate Language for Data Sharing (ILDS) was defined, and an additional translator from ILDS to DS/2, again using CWIC, was developed. (A detailed description of ILDS, including its syntax, semantics, and some examples, is contained in SDC document TM-4999, "The Translator from a Relational Based Language (ILDS) to DS/2.") Since IL and ILDS are recursive in nature, it is quite possible that a single IL or ILDS request will, when translated into a target data management language, result in more than one request or statement in the target language. In such cases, the translation interface module must perform a more complex process than simple translation, and the modules developed for DS/2 were able to produce series of requests when necessary.

An important result of building the translators was a clear definition of the properties necessary for a common language. In order to integrate data management systems, it is necessary to access different types of data structures. IL and ILDS are based on a relational data structure, and it is possible, in theory, to represent more complex (e.g., hierarchical or network) structures in a relational form. However, the representations of the same

query for data described in different data structures are, in general, different in both structure and complexity, making the translation from an IL based on a relational data structure to a data management language based on a different structure difficult. Consequently, a more general data structure was sought as a basis for ILDS. A possible candidate is an extension of the relational data structure, called the "hyper-relational" structure, in which names of relations are permitted in a relation, in addition to names of elements. This structure has two advantages:

1. It conveniently represents hierarchical and network data structures.

2. Since the logical representations of data in ILDS and the target management system are similar, the translation process should be simpler and more natural.

It is also important for the development of a common language to note that the data structures reflected in the syntax of a data management language may not be the same as the logical data structures defined when the data are initially represented in the data management system. The differences may take the following forms:

- The data management language may use only a subset of the logical structures. For example, in a hierarchical (tree) structure, only terminal nodes (leaves of the tree) may actually be expressible in the data management language.

- Additional links between structure elements that are not actually part of the logical structure may be implied by the data management language.

Since the common language should be based on the data structures that are implied and permitted in the target data management languages, the development of an ILDS based on "hyper-relations" has been postponed until a study of the above differences is made. This task will be performed during the next year.

Another important property of a common language is its set of functional features (such as the query and update functions). These features must be considered in light of their ability to be (a) represented in English and (b) translated. A study of these features as they are discussed in the MITRE[1] and CODASYL[2] reports has been initiated and will continue during the next year.

---

[1] "Data Management Systems Survey," MITRE Corporation, MTP-329, January 1969.

[2] "Feature Analysis of Generalized Data Base Management Systems," CODASYL Systems Committee, April 1971.

5.3.2 Summary

During this year, the Data Sharing Project was focused mainly on demonstrating the feasibility of building translators for a common language based on relational structures and on studying the requisite properties of a common data-sharing language in terms of its translation into disparate data management languages. The year's accomplishments and activities are summarized below.

1. A translator was built from CONVERSE's IL to SDC's DS/2 data management language, using the CWIC metacompiler.

2. The design for an ILDS--an intermediate language more suitable for data sharing--was begun.

3. A translator was built from ILDS to DS/2, using CWIC.

4. The concept of "hyper-relations" was developed as a possible basis for the common language.

5. A preliminary study of the requisite data management features for the common language was begun, along with a study of data structures reflected in several target data management languages.

During the coming year, the project will continue the study and development of the common language, based on this year's experience and conclusions.

5.4 STAFF

Systems Research

R. R. Linde, Principal Investigator
Beth Austin
W. F. Gardner
J. G. Hata

ARPANET Engineering

Dr. R. E. Long, Principal Investigator
A. Landsberg

Computer Network Data Sharing

Dr. A. Shoshani, Principal Investigator
I. Spiegler

5.5 DOCUMENTATION

Landsberg, A. ARPA Network Implementation Under ADEPT. SDC TM-4891, 3 March 1972.

Long, R. E. LISP 1.5 Primitives for Using the ARPANET. SDC TM-4310/201, 13 April 1972.

Shoshani, A. Data Sharing on Computer Networks. SDC SP-3655. (Presented at the Western Electronic Show and Convention, Los Angeles, September 1972.)

Shoshani, A., and Spiegler, I. Intermediate Language for Data Sharing. SDC TM-4998, in press.

Shoshani, A., and Spiegler, I. The Translator from a Relational Based Language (ILDS) to DS/2. SDC TM-4999, in press.

Shoshani, A., and Spiegler, I. The Translator from English through CONVERSE's Intermediate Language (IL2) to DS/2. SDC TM-5000, in press.