September, 1976

. Vol. 25, No. 9

computers and people

formerly *Computers and Automation*

COMPUTER HELPS BOTANISTS

Computer Programming Using Natural Language: A Brief Report, 1976

Edmund C. Berkeley, Casper Otten,
Eric Albert, and Steve Emmerich
Berkeley Enterprises, Inc.
815 Washington St.
Newtonville, Mass. 02160

"Lots of people will no longer need to learn BASIC or FORTRAN or COBOL or any other artificial programming language in order to tell a computer what they want it to do."

This is a brief report on a system of computer programs running on our computer, a PDP-9 of Digital Computer Corp. This system of computer programs we call GENIE. Here is a flowchart of GENIE.
For a class of problems:

FIRST INPUT:
instructions from a manager to a clerk for a calculation, expressed in ordinary natural language

SECOND INPUT:
one or more worked examples of the calculation, preferably at least two

THIRD INPUT:
the calculation layout sheet, the report layout sheet, etc.

BLACK BOX:
an automatic computer program called GENIE, which produces:

THE OUTPUT:
a computer program which takes in a set of data in regard to the class of problems being calculated and gives out the desired set of answers

The first input is called the Statement Analyzer. The second input is called the Worked Example Analyzer. The third input is called the Layout Analyzer. The Black Box is called the Program Maker; at present there are seven program makers, for COBOL, FORTRAN, Dartmouth BASIC, Business BASIC, FOCAL, APL, and PDP-9 machine language.

Suppose we consider a concrete example, a class of simple problems which we shall call "Wage Report". An illustration and numerical example is shown in Table 1. In Table 2 appears one set of instructions or statements, and then a second set of instructions; these are alternative inputs to the Statement Analyzer. Of course, many hundreds of thousands of other sets of instructions could be used instead. Table 3 shows input to the Worked Example Analzyer. In this case no input is needed to the Layout Analyzer. Tables 4 to 9 show programs automatically produced by the GENIE system. These are in six programming languages: FORTRAN, Dartmouth BASIC, Business BASIC (for the BASIC/Four computer), APL, FOCAL, and COBOL. The automatically produced program in PDP 9 machine language is too long to be presented in this article.

Some Questions and Answers

Q: The programs produced show some differences, i.e., they do not all give just the same results. Why?- A: True. They were finished at different times, and modification so that all the program makers should produce just the same results seemed less important than some other tasks.

Q: Will the Statement Analyzer understand every set of instructions that expresses what is to be done?- A: It should, provided the set of instructions is in ordinary natural language, and is clear and unambiguous to a clerk, and conforms with a few simple rules, such as "Put heading names in parentheses or quotes."

Q: Does the system handle IF, GO TO, lookup in tables and files, etc.?- A: Yes, of course.

Q: If computer programming using ordinary natural language becomes widespread, will human programmers be needed any more? - A: Lots of people will no longer need to learn BASIC or FORTRAN or COBOL or any other artificial programming language to tell a computer what they want it to do. But human programmers will still be needed for some tasks, such as devising algorithms, etc.

Q: How do I get more information? - A: Write us on your letterhead. We'll try to reply.

Table 1: WAGE REPORT — ILLUSTRATION AND NUMERICAL EXAMPLE

(1)	(2)	(3)	(4)
Dues Amount	Tax Exemption	Income Tax Rate	Social Security Contribution Rate
CONSTANT	CONSTANT	CONSTANT	CONSTANT
3.00	150.00	.11	.0585
(5)	(6)	(7)	(8)
Gross Pay	Taxed Portion	Income Tax	Federal Insurance Contribution Act
DATA	=(5)-(2)	=(6)*(3)	=(8)*(4)
200.00	50.00	5.50	11.70
(9)	(10)	(11)	(12)
Club Deduction	Intermediate Total	Total Deductions	Net Pay
=(1)	=(7)+(8)	=(10)+(9)	=(5)-(11)
3.00	17.20	20.20	179.80

Table 2: INPUT TO THE STATEMENT ANALYZER – TWO ALTERNATIVES

```
CALL THIS PROGRAM WAGE-REPORT.
PUT 3.00 INTO ITEM 1 (DUES AMT).
PUT 150.00 IN ITEM 2 (TAX BASE).
PUT .1100 IN ITEM 3 (TAX RATE).
PUT .0585 INTO ITEM 4 (SS RATE).
GET AN ITEM OF DATA, AND CALL IT ITEM 5
   (GROSS PAY)
SUBTRACT ITEM 2 FROM ITEM 5 AND PUT THIS
   RESULT IN ITEM 6 (TAXABLE INCOME).
ITEM 3 TIMES ITEM 6 GIVES ITEM 7 (INCOME
   TAX).
THE PRODUCT OF ITEM 5 AND ITEM 4 IS
   ITEM 8 (FICA).
MOVE ITEM 1 TO ITEM 9 (CLUB).
ITEM 10 IS ITEM 7 ADDED TO ITEM 8.
ITEM 10 PLUS ITEM 9 IS ITEM 11
   (DEDUCTIONS).
THE DIFFERENCE BETWEEN ITEM 11 AND ITEM 5
   EQUALS ITEM 12 (NET PAY).
```

```
THIS PROGRAM'S NAME IS WAGE-REPORT.
ENTER 3.00 INTO ITEM 1 (DUES AMT).
PUT 150.00 INTO ITEM 2 (TAX BASE).
MOVE .1100 TO ITEM 3 (TAX RATE).
ENTER .0585 INTO ITEM 4 (SS RATE).
PUT DATA INTO ITEM 5 (GROSS PAY).
ITEM 5 MINUS ITEM 2 IS ITEM 6 (TAXABLE
   INCOME).
ITEM 3 MULTIPLIED BY ITEM 6 GIVES ITEM 7
   (INCOME TAX).
ITEM 5 TIMES ITEM 4 IS ITEM 8 (FICA).
STORE ITEM 1 IN ITEM 9 (CLUB).
ITEM 7 PLUS ITEM 8 IS EQUAL TO ITEM 10.
ADD ITEM 10 TO ITEM 9 TO GET ITEM 11
   (DEDUCTIONS).
ITEM 5 LESS ITEM 11 EQUALS ITEM 12
   (NET PAY).
```

Table 3: INPUT TO THE WORKED EXAMPLE ANALYZER

```
(1) 3.00 .=DUES AMT
(2) 150.00 .=TAX BASE
(3) .11 .=TAX RATE
(4) .0585 .=SS RATE
(5) 200.00 222.22 .=GROSS PAY
(6) 50.00 72.22 .=TAXABLE INCOME
(7) 5.50 7.94 .=INCOME TAX
(8) 11.70 13.00 .=FICA
(9) 3.00 .=CLUB DEDUCTION
(10) 17.20 20.94 .=INTERMEDIATE TOTAL
(11) 20.20 23.94 .=TOTAL DEDUCTIONS
(12) 179.80 198.28 .=NET PAY
```

Table 5: Dartmouth BASIC Program

```
100 REM " WAGE REPORT"
101 BEGIN
120 A1=3.00
140 A2=150.00
160 A3=.1100
180 D5=1
200 A4=.0585
220 INPUT A5
240 A6=A5-A2
260 A7=A6*A3
280 A8=A5*A4
300 A9=A1*D5
320 B0=A7+A8
340 B1=B0+A9
360 B2=A5-B1
380 END
```

Table 6: BUSINESS BASIC Program

```
100 REM "WAGE REPORT"
101 BEGIN
102 OPEN (7)"LP"
120 A1=3.00
121 PRINT(7) " DUES AMT",A1
140 A2=150.00
141 PRINT(7) " TAX BASE",A2
160 A3=.1100
161 PRINT(7) " TAX RATE",A3
180 D5=1
200 A4=.0585
201 PRINT(7) " S S RATE",A4
220 PRINT(7) " GROSS PAY"
221 INPUT A5
222 PRINT (7)A5
240 A6=A5-A2
260 A7=A6*A3
261 PRINT(7) " INCOME TAX",A7
280 A8=A5*A4
281 PRINT(7) " FICA",A8
300 A9=A1*D5
301 PRINT(7) " CLUB",A9
320 B0=A7+A8
340 B1=B0+A9
341 PRINT(7) " DEDUCTIONS",B1
360 B2=A5-B1
361 PRINT(7) " NET PAY",B2
380 END
```

Table 4: FORTRAN Program

```
C       WAGE REPORT
        WRITE(3,100)
        READ(1,100)A1
 100    FORMAT(F8.5)
        WRITE(3,200)
        READ(1,200)A2
 200    FORMAT(F8.5)
        WRITE(3,300)
        READ(1,300)A3
 300    FORMAT(F8.5)
        READ(1,400)A35
 400    FORMAT(F8.5)
        READ(1,500)A4
 500    FORMAT(F8.5)
        WRITE(3,600)
        READ(1,600)A5
 600    FORMAT(F8.5)
        A6=A5-A2
        WRITE(3,700)A6
 700    FORMAT(1X,'TAXABLE INCOME',F8.5)
        A7=A6*A3
        WRITE(3,800)A7
 800    FORMAT(1X,'INCOME TAX',F8.5)
        A8=A5*A4
        WRITE(3,900)A8
 900    FORMAT(1X,'FICA',F8.5)
        A9=A1*A35
        WRITE(3,1000)A9
 1000   FORMAT(1X,'CLUB',F8.5)
        A10=A7+A8
        A11=A10+A9
        WRITE(3,1100)A11
 1100   FORMAT(1X,'DEDUCTIONS',F8.5)
        A12=A5-A11
        WRITE(3,1200)A12
 1200   FORMAT(1X,'NET PAY',F8.5)
        END
```

(please turn to page 26)

Vol. 25, No. 9
September, 1976

"Computers and People", formerly "Computers and Automation", is published monthly, 12 issues per year, at 815 Washington St., Newtonville, MA 02160 by Berkeley Enterprises, Inc. Printed in U.S.A. Second Class Postage paid at Boston, MA, and additional mailing points.

Subscription rates: United States, $11.50 for one year, $22.00 for two years. Canada: add $1 a year; elsewhere, add $6 a year.

NOTE: The above rates do not include our publication "The Computer Directory and Buyers' Guide". If you elect to receive "The Computer Directory and Buyers' Guide", please add $12.00 per year to your subscription rate in U.S. and Canada, and $15.00 elsewhere.

NOTE: No organization in Switzerland or Morocco is authorized or permitted by us to solicit subscriptions for or receive payments for "Computers and People" or "The Computer Directory and Buyers' Guide". Such subscriptions and payments should be sent directly to us.

Please address mail to: Berkeley Enterprises, Inc., 815 Washington St., Newtonville, MA 02160

Postmaster: Please send all forms 3579 to Berkeley Enterprises, Inc., 815 Washington St., Newtonville, MA 02160

Change of address: If your address changes, please send us both your new address and your old address (as it appears on the magazine address imprint), and allow three weeks for the change to be made.

computers and people

formerly *Computers and Automation*

Computers and Programs

Computers and Privacy

The magazine of the design, applications, and implications of information processing systems – and the pursuit of truth in input, output, and processing, for the benefit of people.

Front Cover Picture

Some flowering plants are on the way to extinction. Researchers at Notre Dame University say that the number of different kinds of flowers in North America is declining. They have analyzed, with computer assistance, 35,000 specimens of plants collected in North America since 1800. See the story on page 22.

Key

[A]	–	Article
[C]	–	Monthly Column
[E]	–	Editorial
[F]	–	Forum
[N]	–	Newsletter
[R]	–	Reference

NOTICE

*D ON YOUR ADDRESS IMPRINT MEANS THAT YOUR SUBSCRIPTION INCLUDES THE COMPUTER DIRECTORY. *N MEANS THAT YOUR PRESENT SUBSCRIPTION DOES NOT INCLUDE THE COMPUTER DIRECTORY.

EDITORIAL

"The Thinking Computer: Mind Inside Matter"

A recent very interesting and important book is "The Thinking Computer: Mind Inside Matter" by Bertram Raphael, a member of the staff of Stanford Research Institute; the book was published in 1976 by W.H. Freeman and Co., 660 Market St., San Francisco, CA, 94104.

This book is a survey and report of a large number of interesting efforts to put "thinking" inside a computer. Bertram Raphael himself has had a lot of experience in this area, particularly in the construction of a rather famous robot called "Shakey". Shakey has a TV camera and other accessories and communicates through an antenna with its "brain," a computer: this is an example of the interesting principle that a robot may be quite separate physically from its brain.

According to a report published in July 1973 in the "Boston Globe", you can put Shakey into an office, and Shakey will "observe" the locations of doors, windows, and other major objects of the office. Then you can give Shakey a command like "go into the room three doors down the corridor and move a box from one corner of that room into a position where it blocks the door to the room." That's all the instruction Shakey needs. Shakey will proceed to execute the task, dealing reasonably with features and obstacles in its path — for which the computer program has been prepared.

We are fortunate to have permission from the publisher to print a major part of Chapter 6, "Natural Language", commencing in this issue.

The outline of this chapter is:

1. Natural and Artificial Languages
2. Syntax and Semantics
3. Formal Representations of Natural Language
4. Phrase Structure Grammars
5. Ambiguity
6. Transformational Grammar
7. Approaches to Semantics
8. Programming Computers to Understand English
9. Question-Answering in Restricted Domains
10. Representation and Use of General Knowledge

It is not feasible for us to print all of this chapter, which is very long. Omissions have been indicated by the sign "....."; and readers who are interested are urged to get hold of the book and read it in full. The book is crammed full of interesting and not too difficult examples and illustrations, written by an author who knows how to explain vividly and well.

It is impossible to ignore the mounting evidence that computers can think and do think in many, many ways. And this book is a milestone along the path of recording the evolution of a thinking computer which will rival man, and a worthy addition to every library, both computer and non-computer.

Edmund C. Berkeley

Edmund C. Berkeley

Editor

The Handling of Natural Language by a Computer — Part 1

Bertram Raphael
Stanford Research Institute
Menlo Park, CA 94025

"When scientists encounter a problem that is too difficult for them to solve directly, they usually look for simpler versions or special cases of the problem to work on first."

Natural and Artificial Languages

It has been said that the fact that man uses language to communicate distinguishes him from the other beasts. Of course, this claim is a great oversimplification. Some form of language is used at many levels in the animal kingdom. The jungle is filled with sounds and motions that convey messages of danger or desire. Dolphins seem to have an elaborate system for sonic communication, and the dancing motion of bees seems to be capable of describing the distance and direction of food sources with amazing precision. Still, all human cultures use spoken language for far more than helping to satisfy basic bodily needs. The spoken and written word are a basis for describing complex situations, for developing abstract ideas, for combining the separate knowledge of many people into jointly developed conceptions, and for accumulating and conveying information from one generation to the next in an evolving society.

The Knowledge of Language

Included as part of the knowledge that is passed along from generation to generation is the knowledge of language itself. Since the time of Babel, thousands of languages have been passed along from parent to child within their own tribes or nations. About a thousand years ago King Frederick II of the Holy Roman Empire thought that language was an inherent (or instinctive) ability of human beings. To test his theory, he took a group of newborn infants away from their parents and brought them up in isolation from the rest of society. Even their nurses were not allowed to disturb the experiment by speaking in the children's presence. The idea was to observe what "built-in" language the children would begin speaking spontaneously: the language of their local society, Latin, Hebrew, or some other ancient "source" language? Unfortunately, the experiment failed; for a variety of reasons that I'm sure you can imagine, none of the children survived more than a few years, and none ever began to speak at all.

Observable Languages

Languages that have ever been used as the principal means of communication in the day-to-day business of a human society are called natural languages. (Those natural languages that are still in use somewhere in the world are called living languages.) These languages are natural in the sense that they are existing, observable entities, just as are the natural biological species, or the physical constituents of matter. Scientists may study and describe them, but cannot arbitrarily force them to change. Living languages change, just as biological species change, by a process of evolution; many small changes are introduced in every generation, and a few — those that improve the effectiveness of the language in some sense — survive to become permanent features. Thus we may expect most natural languages, which have experienced many centuries of such evolution, to be extremely practical and efficient communications media.

Artificial Languages

Artificial languages are languages that have been invented by people for particular kinds of communication. Musical notation is an artificial language with which a composer or arranger conveys his conceptions to musicians. Predicate calculus is one of many artificial languages for special domains of mathematics. And of course, FORTRAN, ALGOL, LISP, and so on, are artificial languages specifically designed to simplify the programming of computers.

Why should computer users be interested in natural language? The specially developed computer languages seem much better suited for communicating with computers. Precisely because they are artificial languages, they may be changed or expanded at the whim of their designers whenever the users think up some new features they would like to have to make some class of programming task easier. Natural languages cannot be tampered with in this way. Moreover, scientists do not yet fully understand the complexities of any natural language, and therefore do not know how to write compilers for natural languages, or other direct computer implementations of them.

Teaching Natural Language to Computers

On the other hand, this book is about how to make computers smarter and more useful, and about the roles computers will play in the next two decades, as they become less expensive and more accessible, rather than the highly restricted and specialized roles they have played until quite recently. In the

Excerpted from Chapter 6, "Natural Language," of *The Thinking Computer: Mind Inside Matter* by Bertram Raphael, published by W. H. Freeman and Co., San Francisco, Calif., copyright 1976, and reprinted with permission.

past computers were used primarily by experts in a small number of specialized fields — engineering, physics, banking — generally with the aid of teams of highly trained computer specialists. Now we are in a period of transition to a time when computers will be almost as readily available as typewriters or telephones, and the potential users will be people from all walks of life — doctors, lawyers, school children, housewives — who will have their own direct access to considerable computing power. How will they use it? Will every educated person learn computer programming as a basic skill, along with reading and arithmetic? Perhaps, but the burden of acquiring this additional skill would surely impede people from taking full advantage of the availability of this new resource. Instead of teaching computer languages to people, perhaps a preferable approach would be to teach natural languages to computers; then people would be able to communicate with computers as easily as they communicate with other people, in a common language already familiar to them.

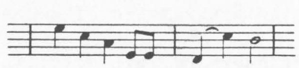
This article considers the problems that have been encountered and the progress that has been made in the process of trying to teach computers to understand natural language.

Syntax

The study of languages, both natural and artificial, has traditionally been divided into two major areas: syntax and semantics. Syntax deals with the formal structure of the strings of symbols that make up the sentences of the language, without regard for their meanings.

Grammar

The elementary symbols of the language may be combined only in certain ways, as prescribed by a set of rules called the grammar of the language. The basic task for syntactic analysis is to tell which strings of symbols are grammatical, i.e., legitimately belong to the language, and which are not. A musician can play only grammatical music; a computer can compile only grammatical FORTRAN. Figure 1 shows some examples of grammatical and ungrammatical samples of well-known artificial languages.

	Grammatical	Ungrammatical
Music		
Logic	$(\forall x)(\forall y)(\forall z)[P(x,y) \wedge P(y,z) \supset P(x,z)]$	$P(\forall]]x \wedge \supset (z)y$

FORTRAN	`DO 10 I=1,100`
	`DO 20 J=1,100`
	`20 A(I,J)=0`
	`10 B(I)=0`
		`DO 10 I=1 =100`
		`DO 20 1,100=J`
		`10 A(I,J)=0`
		`20 B(A) =0`

Figure 1: Grammatical and ungrammatical sample formal language segments.

Semantics

Semantics refers to the meanings of the symbols and of the grammatical symbol-strings of the language. The semantics of a musical score consists of tones, durations, and sound qualities; the semantics of a computer program consists of arithmetic and symbolic operations taking place in the registers of a computer. Since ungrammatical expressions seldom have semantic interpretations, we usually study formal languages in two sequential phases. First, syntactic analysis determines whether the expression is grammatical; then, if it is grammatical, semantic analysis determines what it means. (In the computer-language example, this separation is especially clear: a compiler operates in an almost purely syntactic manner, and only at "run time," when the results of the compilation are executed, does the semantics emerge.)

Until the 1950s most linguists thought that natural language could be analyzed in separate sequential phases: first syntactic and then semantic. Natural language has features that are clearly syntactic and features that are clearly semantic. It was believed that certain syntactic transformations could be applied to natural language without affecting meaning, just as a musical score can be changed in certain ways without changing the sounds it represents. This belief was the basis for massive efforts during the 1950s to have computers translate text from one language to another. First large dictionaries were placed on computer tapes so that the translations of individual words could easily be looked up. Then elaborate grammars were developed to explain the differences, from one language to another, of such obvious syntactic features as word order, noun cases, and verb tenses. Linguists working with computer scientists hoped that, if their programs captured enough of the syntactic differences between the languages, then translated sentences would come through with their meanings undisturbed. Unfortunately, these experiments failed miserably, producing translations whose meanings differed from the original in all kinds of strange, unexpected ways. For example, when the biblical quotation, "The spirit is willing but the flesh is weak," was translated from English to Russian and then back to English, what came out of the computer was, "The wine is agreeable but the meat has spoiled."

Fuzzy Boundaries

The lesson learned from these early efforts at mechanical translation was that the boundary between syntax and semantics in natural language is extremely fuzzy. No system of grammatical rules has been discovered, or now seems likely to be discoverable, that can describe the structural properties of natural language without being concerned also with semantics. The formal ways in which words can be strung together and the meanings of those strings appear to be interrelated in subtle and complex ways.

Some English sentences are clearly both grammatical and meaningful, "John gave the carrot to Mary," and some are clearly so ungrammatical as to be meaningless, "To gave Mary John carrot the." However, we also can construct English expressions whose meanings are perfectly clear even though the expressions are obviously ungrammatical: "I ain't never been there," "Me, Tarzan; you, Jane," "Them's them," and expressions that are perfectly meaningless even though they give the impression of being completely grammatical: "'Twas brillig, and the slithy toves did gyre and gimble in the wabe," "Colorless green ideas dream furiously." In fact, grammar (syntax) and meaning (semantics) of natural language are inextricably intertwined, almost from the basic definitions.

Mathematician's Grammar vs. Linguist's Grammar

The linguist — the scientist who studies the nature of language — would agree with the mathematician in defining the grammar of a language to be a set of rules that identifies which sentences belong to a language and which sentences do not belong.

However, the mathematician, who is concerned with artificial languages, is satisfied to define the language by the grammar; a given sentence is part of a given artificial language if and only if the grammar permits it to be. The linguist, on the other hand, is faced with an existing natural language. The grammars he constructs are only approximations to the real, but unformalized, grammar of the language. His ultimate test of whether a given sentence is part of a given natural language is to ask some native speakers of the language; and they will usually reply that the sentence is grammatical only if it is meaningful to them. Therefore any grammar constructed as an approximate description of a natural language must attempt to separate, not only allowable from structurally ill-formed sentences, but also meaningful from meaningless sentences.

In the next section we shall examine in more detail the kinds of grammars that have been proposed to describe the structure and meaning of sentences of natural language, and in the section following that I will present examples of computer systems that have been programmed to "understand" some aspects of natural language.

Formal Representations of Natural Language

The simplest form of grammar, for a natural or an artificial language, is a list of all the permissible sentences. If we had such a list, we could tell whether a candidate sentence was in the language described by the grammar merely by looking at the list to see whether the sentence was there. Unfortunately, interesting languages generally have such a large number of sentences that a list of all of them is not feasible. Instead, we need some kind of concise statement, such as a formula or a set of rules, that constitutes a summary description of the huge, perhaps even infinite, number of sentences.

Phrase-Structure Grammars

A phrase-structure grammar is the most common way to represent an infinite number of sentences by a small set of rules. In it each rule consists of a symbol, an arrow, and a string of symbols, and means that the string of symbols on the right of the arrow may be substituted for the symbol on the left of the arrow. Certain symbols, called terminal symbols, never appear on the left of any arrow. ....

The grammar G defines a language L that contains only six words in the sentence, "John gave the carrot to Mary." What other sentences does L contain? By simply interchanging the use of rules 9 and 10 in the derivation, we see that L also contains "Mary gave the carrot to John," which is also a reasonable English sentence. However, by applying the rules in different ways, we can show that L also contains a variety of other "sentences" including "John gave John," "The Mary gave John to the carrot," "Carrot gave the Mary to carrot," and so on. ....

Clearly the language L is very different from English. However, if we can characterize the differences precisely enough, perhaps we can find ways to change G, by adding to or replacing its rules, to produce a new phrase-structure grammar G' in which the language L' accepted by G' would be closer to English than L is. By repeating this procedure with L', and so on, we can get successively better versions of a phrase-structure grammar for English.

Scientists in the 1950s posed the question: "Could such a procedure produce a phrase-structure grammar that accepts a language close enough to ordinary English to be useful for such practical purposes as mechanical translation?" This goal was considered possible, partly because phrase-structure grammars were known to be powerful enough to describe rather complex artificial languages, such as computer-programming languages, with precision. This goal was also considered desirable, because the formal properties of phrase-structure grammars were well understood. In particular, although our discussion has explained how to use a grammar for generating random acceptable sentences (by starting with a sentence S and making successive substitutions), there are efficient procedures for running the grammar backwards: i.e., starting with any sentence and producing its structural description (or proving that the sentence is not grammatical). This property of a grammar is extremely important for practical applications, but it is not always present in more complex kinds of grammars. ....

The practicality of phrase-structure grammars (and the programs that make use of them) depends upon the existence of broad general word categories, such as noun and verb, whose roles may be summarized by rules of the grammar. Since such categories seem to be grossly inadequate when we examine natural language closely, phrase-structure grammars, as a sole means of describing natural language for computer processing have been generally abandoned.

Ambiguity

Almost every change made in a grammar (of phrase structure or similar nature) either to increase its scope (to accept more of English) or to narrow its range (to rule out more nonsense), requires an increase in the number of rules of the grammar. The new rules often interact with previously established rules in unexpected ways. As a result of such interactions, complex grammars frequently have several different ways of accepting the same sentence: i.e., one sentence may have several different structural descriptions. We then say that the sentence is ambiguous with respect to the grammar; each structural description generally corresponds to a different possible meaning of the sentence.

The most elaborate phrase-structure grammar of English ever implemented on a computer was developed at Harvard in the early 1960s. It had many thousands of rules, and as a result frequently found several, sometimes dozens, of different structural descriptions for the same sentence. For example, one five-word sentence was found to have four distinct interpretations! When given the sentence,

"Time flies like an arrow,"

the Harvard system produced structural descriptions corresponding to the following meanings:

1. Time moves in the same manner that an arrow moves.
2. Measure the speed of flies in the same way that you measure the speed of an arrow.
3. Measure the speed of flies that resemble an arrow.
4. A particular variety of flies called "time-flies" are fond of an arrow.

Now these are all certainly meaningful interpretations of the sentence, even if some of them are rather peculiar. In fact, I suspect that the designers of the Harvard analysis system were delighted that their program was capable of dreaming up such novel interpretations. However, if we want to produce practical computer systems that will understand what we say, we don't want the program first to list all the possible obscure and peculiar (although syntactically correct) interpretations of everything we say, and then to zero in on the one that makes the most sense. Instead, we shall prefer that our programs immediately recognize the "right" interpretation, based perhaps upon previous conversation or perhaps upon the program's general knowledge of what makes the most sense. Linguists are now moving toward theories of language, which we shall review briefly below, that tie meaning into syntactic analysis. Eventually these theories may evolve into a formal basis for language analysis by computer that will augment the ad hoc implementations now under development.

Transformational Grammar

A major innovation in the structure of grammars designed to approximate the grammar of natural language is called transformational grammar. This approach to organizing grammar was first proposed by Chomsky at MIT in about 1957, and has undergone continued study, elaboration, and modification ever since. Transformational grammars attempt to keep most of the appealing simplicity of phrase structure, yet enrich its descriptive power in order better to approximate natural language. The mechanism for this enrichment is the addition of a series of levels of language description. At the bottom of the ladder is a phrase-structure grammar, called the base component of the transformational grammar. This base component can be used to generate strings of terminal symbols and their structural descriptions, just like the phrase-structure grammars discussed previously. However, the structural descriptions generated by the base component are not intended to describe unique acceptable English sentences as they would appear in text; instead, these structural descriptions are intended to describe the unique meanings that English sentences can have, and are called the deep structure representations of sentences. The other major component of transformational grammar, the transformational component, consists of a set of rules (called transformations) that can change the deep structures produced by the base component into other structural descriptions — surface structures — that do describe actual sentences. Sometimes several transformations may be applied successively, producing several intermediate levels of description between the deep structure and the surface structure.

Transformations

The same deep structure, when subjected to different transformations, can produce surface structures that appear quite different from each other. For example, "John gave the carrot to the aardvark," "Did John give the aardvark the carrot?" "The carrot was given to the aardvark by John," and perhaps even, "The large burrowing nocturnal African mammal of the order Tublidentata received the biennial herb of the family Umbelliferae from John," can all have the same deep structure: a representation of the transfer of a certain vegetable from a certain person to a certain animal. The four transformations illustrated above each have wide general use: one produces a simple declarative sentence, one produces a question, one constructs the passive voice, and one paraphrases the sentence (with the help of a dictionary!). At the same time, several different deep structures may sometimes lead to the same or similar surface structures; for example, a complete transformational grammar should have four distinct deep-structure representations corresponding to the four different meanings of "Time flies like an arrow."

Transformational grammars are a much better tool than phrase-structure grammars for describing natural language, because of the richness and flexibility provided by the transformational component. However, two major problems have hindered the development of useful computerized language analysis systems based upon such grammars.

A Practical Problem

Transformational grammars are basically generative in nature. They prescribe how to use a base component to produce deep structures and how to transform deep structures to produce surface structures that represent all the sentences of a given language (which may be an approximation of some natural language). They do not prescribe how to analyze a given candidate sentence to tell whether it belongs to the language, and if so what its deep structure is. If the sentence has several possible deep structures ("Time flies like an arrow"), we have no guidance as to how to choose one — the problem of ambiguity has changed form, but has certainly not gone away. Unlike phrase structure rules, transformations are very difficult to run "backwards," and attempts to do so have generally resulted in slow, cumbersome procedures.

An Underlying Theoretical Problem

Deep structures are supposed to represent unique meanings of sentences. But how can one represent a meaning? Isn't that really the purpose of natural language in the first place? One might argue that, since we do not know precisely how meaning is conveyed by natural language, we have merely put off the problem, and perhaps obscured it, by replacing it with the problem of how to convey meaning by the deep structures hidden under the transformations. This is a little like the school of philosophy that once "explained" the nature of intelligence by hypothesizing a "homunculus" — a little man who makes all the intelligent decisions — inside everyone's head.

Meaning Is the Key

Things are not really all that bad. In the first place, the simple realization and general acceptance that understanding meaning is the key to understanding language is a significant result of the past twenty-five years' (1950-1975) work in linguistics and related fields. Second, every natural language certainly does have a structure that must be analyzed in order to understand the meanings conveyed, and transformational grammar provides one good framework for describing this structure. Finally, we are not completely in the dark about how to represent meanings. Chomsky proposed one way in his original descriptions of deep structure, and many scientists are actively engaged in exploring different approaches, both to fit into the transformational-grammar scheme and to use in other more ad hoc language-processing systems. The next subsection reviews some of these approaches to the automatic handling of the meaning of language, or (to use a fancier word that means the same thing) semantics.

Approaches to Semantics

The basic approach to semantics, like the approach to syntax, has been to define certain groupings or other logical structures comprising words. However, semantic groups contain words that play a similar role in conveying meaning, as opposed to syntactic groups of words that play a similar role in forming sentence structures.

(please turn to page 18)

Jabberwocky and Gizzmo

A number and word puzzle for a nimble mind and a computer

Neil Macdonald
Assistant Editor

Probably the most famous nonsense poem in the English language is "Jabberwocky" by Lewis Carroll. In "The Annotated Alice" with an Introduction and Notes by Martin Gardner, published by Clarkson M. Potter Inc., New York, in 1960, 351 pages, many of the famous nonsense words invented by Lewis Carroll are commented on. At least "galumph" and "chortle" have found their way into the Oxford English dictionary.

This poem has a strong connection with the type of puzzle called GIZZMO, which from time to time is published in "Computers and People". For an example, see page 18 of this issue.

The following information leads up to a JABBERWOCKY GIZZMO puzzle which is stated below.

Carroll's Poem

'Twas brillig, and the slithy toves
  Did gyre and gimble in the wabe:
All mimsy were the borogoves,
  And the mome raths outgrabe.

"Beware the Jabberwock, my son!
  The jaws that bite, the claws that catch!
Beware the Jubjub bird, and shun
  The frumious Bandersnatch!"

He took his vorpal sword in hand:
  Long time the manxome foe he sought -
So rested he by the Tumtum tree,
  And stood awhile in thought.

And, as in uffish thought he stood,
  The Jabberwock, with eyes of flame,
Came whiffling through the tulgey wood,
  And burbled as it came!

One, two! One, two! And through and through
  The vorpal blade went snicker-snack!
He left it dead, and with its head
  He went galumphing back.

"And hast thou slain the Jabberwock?
  Come to my arms, my beamish boy!
O frabjous day! Callooh! Callay!"
  He chortled in his joy.

'Twas brillig, and the slithy toves
  Did gyre and gimble in the wabe:
All mimsy were the borogoves,
  And the mome raths outgrabe.

Special Context Words: Number and Frequency

BANDERSNATCH	1	GALUMPHING	1	SLITHY	2
BEAMISH	1	GIMBLE	2	SNICKER-SNACK	1
BOROGOVES	2	GYRE	2	TOVES	2
BRILLIG	2	JABBERWOCK	3	TULGEY	1
BURBLED	1	JUBJUB	1	TUMTUM	1
CALLAY	1	MANXOME	1	UFFISH	1
CALLOOH	1	MIMSY	2	VORPAL	2
CHORTLED	1	MOME	2	WABE	2
FRABJOUS	1	OUTGRABE	2	WHIFFLING	1
FRUMIOUS	1	RATHS	2	NU 29 FQ 43	

Jabberwocky Gizzmo Puzzle 769

The numbers in the following version of the poem stand for the special context words in "Jabberwocky." Replace each number by an ordinary natural English word which makes good sense and good poetry in the context.

The best solution(s) from readers will be published in future issues of "Computers and People".

'TWAS 1 AND THE 2 3
DID 4 AND 5 IN THE 6;
ALL 7 WERE THE 8
AND THE 9 10 11.

"BEWARE THE 12, MY SON!
THE JAWS THAT BITE, THE CLAWS THAT CATCH!
BEWARE THE 13 BIRD, AND SHUN
THE 14 15!"

HE TOOK HIS 16 SWORD IN HAND;
LONG TIME THE 17 FOE HE SOUGHT.
SO RESTED HE BY THE 18 TREE
AND STOOD AWHILE IN THOUGHT.

AND, AS IN 19 THOUGHT HE STOOD,
THE 12, WITH EYES OF FLAME,
CAME 20 THROUGH THE 21 WOOD,
AND 22 AS IT CAME.

ONE, TWO! ONE, TWO! AND THROUGH AND THROUGH
THE 16 BLADE WENT 23.
HE LEFT IT DEAD, AND WITH ITS HEAD
HE WENT 24 BACK.

"AND HAST THOU SLAIN THE 12?
COME TO MY ARMS, MY 25 BOY.
O 26 DAY. 27! 28!"
HE 29 IN HIS JOY.

'TWAS 1, AND THE 2 3
DID 4 AND 5 IN THE 6;
ALL 7 WERE THE 8,
AND THE 9 10 11.

The Invasion of Privacy and Electronic Fund Transfer Systems: Spotlight on the Invaders

Seymour F. Thompson
835 Edmands Road
Framingham Centre, MA 01701

" . . . invasion of privacy of the type we all fear the most has very little to do with where the records are or how many there are or whether they're automated or centralized. Invasion has to do with invaders."

Concern over Privacy

An article by Paul Armer, published in Computers and People, June 1976, expressed concern about potential invasions of privacy in an Electronic Funds Transfer System (EFTS). /1/ This is one of many expressions of concern by computer people, consumer advocates, and others. The government commission on privacy held hearings on the subject. The National Commission on EFTS lists invasion of privacy as one of the social issues to be considered.

There are few specific recommendations about how to solve this problem in EFTS, but some broad generalizations have been made. Paul Armer says, " ... I believe that any instance in which a great deal of information about an individual is concentrated in one place represents a threat to his privacy. In our existing systems, privacy is assured under all but the most unusual circumstances by the sheer cost and inconvenience of a search. ... it would seem prudent to me to slow down the pace at which EFTS are being implemented. A bit more time might enable us to erect some safeguards."

Systems Safeguards?

When Paul Armer and Dan McCracken made presentations before the National Commission on EFTS /2/, George Oran, the commission member from the Federal Home Loan Bank Board, asked what Armer and McCracken would recommend to the commission to help them solve the invasion of privacy problem and the potential problems of errors, fraud, and theft on the software side. Their recommendation was to slow down the speed of implementation of the EFTS. Presumably, time should be taken to design systems safeguards as well as legal safeguards in these problem areas.

Emphasis Should Be on the Invaders, Not the Invasion

While I agree with both Armer and McCracken that systems safeguards are very desirable, I do not believe they, or the privacy commission or the EFTS commission, have attacked the invasion of privacy problem from the root direction. The emphasis has been placed on the concept of invasion, but little attention has been paid to the invaders. Other than some general references to "1984", Big Brother, Russia, other communist countries, the IRS, FBI, CIA, and the Nixon administration, the real brass tacks of the situation that has emerged in this "land of the free" have not been touched.

Let's take a look at the real invaders, not some postulated ones, in America, today. Let's examine their motivations, their methods and the resources for invasion of privacy. Then we'll know whether it makes any truly significant difference whether we have centralized EFTS files or not. We'll also have a better idea of what kind of legal systems and other safeguards may have an impact. But most importantly, let's attack the invaders, expose them, and keep them away from our records — wherever they may be.

Long-Standing Invasion

In an article in Computers and Automation a few years ago /3/, Richard E. Sprague pointed out that our privacy had been invaded in America for years and that computerizing files would not change the fundamental opportunities or motivations of the invaders. This article appeared before legislation concerning retail credit files and consumers' rights of access to their own records was enacted. In those days, circa 1950-1970, retail credit records contained much negative information about a person, including newspaper articles and pure gossip. They contained very little positive information, but had plenty of errors. These records were available to just about everyone, except the person whose file it was. A lawyer could pay a private detective to see a person's file without his or her knowing it, or knowing what was in the record. A person could not, legally or forcefully, obtain access to his or her own file or change it to make corrections or delete obsolete material.

Freedom of Information Laws

After the passage of the new laws, a person has the right to see his or her own credit file and the right to change it legally. But how often has this ever been done? How does a person know how many credit files there are on him, and where they are? Who has been looking at them during the last few years?

If you're like me, the answers to these questions are all "No." In spite of my intense concern about invasion of privacy and in spite of the fact that I caught an invader red-handed tapping my home phone, I still do not know where all the records about me are located, nor do I know who has been looking at them.

Getting "Your File" from the CIA and the FBI

I sent to the CIA and FBI for their files on me, which any American citizen may now do under the new freedom of information legislation. The results were exasperating and laughable. Months of delays and lots of apologies came from both agencies. The CIA said their file on me came mostly from the FBI. When I finally received something, it was a copy of a report by the FBI on the people who founded the Institute of Management Sciences (TIMS) in 1949. Why the FBI was investigating TIMS was not clear. I was included as one of the original charter members, hardly comparable to being a card-carrying Communist, and this was way before Senator Joe McCarthy's era.

Invasion Has to Do with Invaders

The point is that invasion of privacy of the type we all fear the most has very little to do with where the records are or how many there are or whether they're automated or centralized. Invasion has to do with invaders. I believe we can classify individuals and groups that wish to gather or disseminate information about you or me, into two categories. Category 1 consists of (1) those whom we want to have information about us, and (2) those whom we don't mind having information about us. On the other hand, Category 2 consists of (1) those whom we do not want to have information about us or to disseminate information about us, or (2) those who, if they do have information about us, we want to know about it beforehand and want the legal right to block them from having or disseminating the information.

We Want Good Credit

Examples of Category 1 are retailers or banks or other credit establishments, either local or national, with whom we have a good credit standing and record or with whom we wish to establish a good credit or bank standing. None of us has any objection to giving three or four credit or bank references when applying for a loan or opening a new charge account. If we are honest citizens, we want positive, good credit information to be passed around, including individual transaction or payment records, if it helps establish our financial reputation and our credit worthiness. We can object to errors creeping into those records, especially the type of error introduced by manual or computerized receivables systems, such that the error is perpetuated and tends to create a negative impression. We can also object to gossip, rumors, and false data being placed in our records. That still does not influence our thinking with regard to giving these institutions access to all of our records. Many other examples of Category 1 could be listed.

We Do Not Want Snooping

Category 2 is the category we are really concerned about. Let's examine these organizations, groups, and individuals in some detail.

Here is a list of known or suspected invaders of privacy belonging in Category 2.

1. Federal Government Agencies
   - FBI
   - CIA
   - IRS
   - Army intelligence
   - Navy intelligence
   - NSA
   - Federal Grand Jury Strike Force
   - White House
   - House Un-American Activities Committee
   - U.S. Department of Justice
   - U.S. Treasury Department
   - Law Enforcement Assistance Agency
2. State and Local Governments
   - State-wide cooperating law enforcement groups
   - State attorney generals
   - State police
   - City police
   - Local district attorneys
   - Special local strike forces (SWAT, etc.)
   - Regional law enforcement groups (western states)
   - Local courts
3. Private Organizations
   - Private detective agencies
   - Credit-investigating organizations

The Senate Intelligence Committee Investigation

Thanks to the efforts of the Church committee and other Senate and House committees investigating U.S. intelligence agencies and other agencies such as the IRS, we now know how extensive the invasion of our privacy has been through the past few decades.

To those who would say, "Ah yes, but after Watergate and the Church investigations, all that activity has stopped," it is only necessary to point out what has been revealed about FBI "bag jobs" during the months of June and July 1976. J. Edgar Hoover presumably ordered that illegal burglaries of "leftist" groups and individuals' homes, offices, and files be stopped in the 1960s. Yet we now know that these FBI invasions of privacy continued until at least 1975, and are very likely still going on.

Continued Snooping

Can anyone doubt the probability that the CIA is still invading the privacy of Americans through mail surveillance, telephone tapping, bugging, and illegal burglaries? Perhaps the army has ceased its invasions of privacy and its creation of a gigantic file on thousands of innocent citizens? /4/ Perhaps the IRS has discontinued its invasion of privacy practices initiated during the Nixon years? Perhaps the NSA has stopped the use of sophisticated data processing and communications techniques to monitor the telephone conversations of thousands of individuals? Perhaps the Postal Service has ceased its mail interceptions on behalf of other government agencies?

Perhaps. But can we really be sure? The Church committee and the privacy commission have barely scratched the surface of what has been going on. It is said by those close to the Justice Department investigation of the FBI that literally hundreds and hundreds of FBI agents and top executives from Hoover on down have been involved in criminal activities affecting privacy. Thousands of FBI privacy invasions have yet to be exposed. If there are thousands of FBI "bag jobs," there must be tens of thousands of CIA privacy invasions.

The IRS is of special interest in the EFTS case because of their potential activities in gathering data illegally (or legally) about the financial status of Americans. The Supreme Court recently ruled that the IRS has the right, under court order, to demand and to receive the bank account records of any U.S. citizen from any financial institution. Many honest taxpayers will be disturbed by that ruling.

(please turn to page 19)

Set Theory and Systems Improvement

Konrad E. Sadek
56 Parkwood Dr.
Cambridge, Ontario
N1S 3L1, Canada

Dr. Edward A. Tomeski
Professor of Management
Fordham University
Bronx, NY 10458

" '. . . there are considerably more persons in the United States engaged full-time in the creation of records than there are farmers. Our annual paper work cost is a multi-billion dollar figure that exceeds the annual federal budget.' "

Introduction

The use of set theory can be a helpful technique for systems designers and analysts, since it can identify the areas of interrelationship between several systems. The approach can facilitate systems improvement by pointing out duplication of effort in related systems, and thus encourage the development of lean systems that are most economical for an organization as a whole.

The Importance of Interdependent Systems

Many, if not most, computer applications are designed and developed relatively independently of each other. This leads to a composit of systems which may be efficient individually but which too often are incompatible with each other. This condition can result in a collection of systems that cause major expenses for an organization.

This article will show how set theory can be used to help design interdependent systems. This does not mean that certain systems cannot be developed independent of each other if the applications are sufficiently unrelated or management is aware of the penalties involved if identifiably interdependent systems are developed in isolation from each other.

Properties of Sets in Systems

First, let us state some general properties of sets as we will use them in this article:

AuB = A union B
AxB = A intersects B

The counting principle for the union of any three sets:

Let A, B, and C be any three sets, therefore if the sets are

1. disjoint, as in Figure 1,
   n(AuBuC) = n(A) + n(B) + n(C)
2. joint, as in Figure 2,
   n(AuBuC) = n(A) + n(B) + n(C) - n(AxB) - n(AxC) - n(BxC) + n(AxBxC)

Inventory Control Intersects Manufacturing

For example, Inventory Control intersects Manufacturing, IxM:

1. disjoint sets: the systems have been designed independently of each other, hence duplications of certain steps in both departments;
2. joint sets: there will be steps in the system which are common to both departments, but the systems have been designed to eliminate duplications.

For simplicity's sake, three departments (sets) of an organization will be used in the following illustration; however, the departments (sets) can be expanded to any number.

Let I = Inventory Control
M = Manufacturing
P = Procurement

Hence, an exclusive systems approach would be shown as in Figure 1.

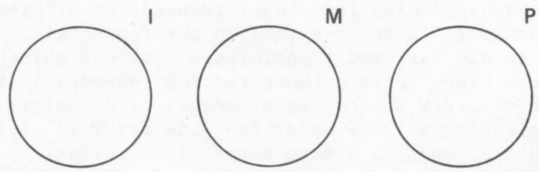


Figure 1

Figure 1 suggests that each department (system) functions independently. In actual operation this could not be the case — the functions of inventory control, manufacturing, and procurement are interdependent functions. If a formal system does not bridge all three departments, then some kind of informal system will fill the void.

Figure 2 shows a more realistic situation; the three functions are truly interdependent.

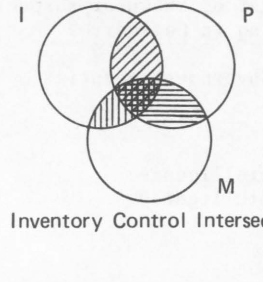


Figure 2: IxP, Inventory Control Intersects Procurement

Cost of Paper Work Exceeds Federal Budget

There are regions (representing work flow, forms, responsibilities) which are overlapping, i.e., IxM. These regions isolate the areas that are likely to contain duplication of effort (e.g., identical files held in Inventory Control and Manufacturing) which wastes organizational resources.

> "The actual number of full-time clerical workers and the true cost of paper work are elusive figures. Nevertheless, there are considerably more persons in the United States engaged full-time in the creation of records than there are farmers. Our annual paper work cost is a multi-billion dollar figure that exceeds the annual federal budget. A mountain of paper casts a shadow of inefficiency and waste over the land. Documents, the majority of which becomes worthless in less than five years, are stored in expensive containers in costly space long after they have fulfilled their purpose." /1/

The same principle holds for IxP and MxP. But the problem does not stop here. An even more complex situation exists by intersecting IxPxM. This is the region in which work, functions, decisions, and forms are tripled.

Systems Improvement Through Overhead Value Analysis

A growing number of organizations are using overhead value analysis in information systems design. Such an effort logically starts in the area of IxMxP, e.g., each department wants to keep a copy of a form, for they tend to mistrust each other. Hence, there may be three identical files. If the three files can be reduced to one file, there will be an obvious benefit to the firm. Here is a classical example of the situation:

> "A defense plant maintained a costly staff of 800 engineers for original research on a multitude of projects. A management expert, wondering about the necessity for this outlay, investigated fifty sample research projects. To the amazement of the top officers, he demonstrated that in half the cases he could have obtained the same or better information by going to the library." /2/

Similar problems do occur in large organizations between divisions and departments, or in medium to large firms between departments. One of the authors had an experience in one firm where the Procurement and Quality Assurance departments had identical files on vendor performance. The files were created four years earlier, because the Quality Engineering department had a bad reputation of not knowing their work. Two years later, the Quality Engineering department became one of the best in the country, but still the two sets of files were kept. Old habits die hard, and management had to force the buyers to not duplicate the information available from the quality engineers. Instead, a master file was created to which all have access.

Cutting Down on Paper Work

Another example are forms. There are so many forms or documents in many organizations, which must be handled by many different persons, and frequently some persons do the same work on the identical form. For example, let's examine what happens to a simple form which goes to all three departments, Inventory Control, Manufacturing, and Procurement (see Figure 3).

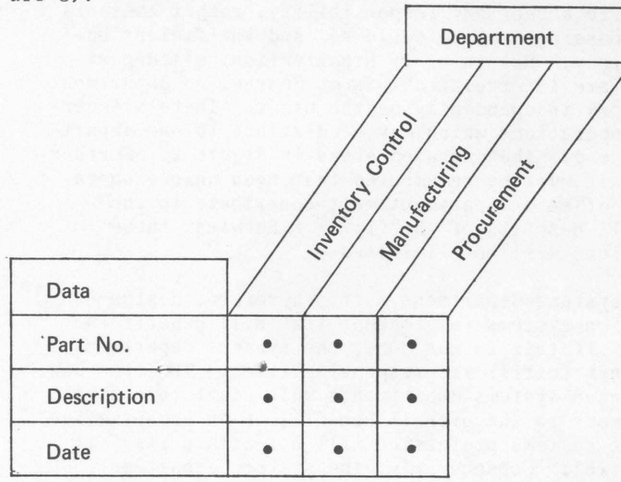


Figure 3

As can be seen, each department writes down the part number and description; the date may vary from department to department. Three people do identical work — writing down the part number and description — and, most likely, all three departments may want to file a copy. The form can be designed to function better, as shown in Figure 4.

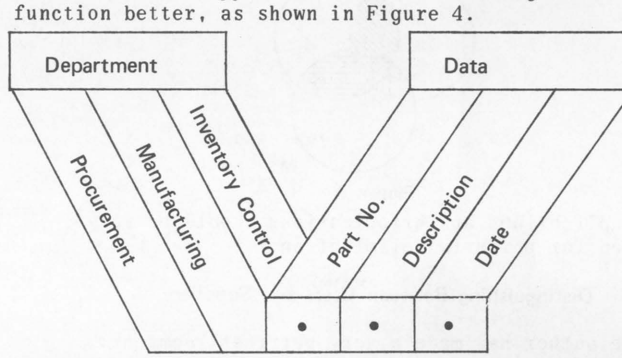


Figure 4

The new form is more functional and easier to use. The schematic form in Figure 4 gives exactly the same information as the schematic form in Figure 3, and it also saves clerical efforts. The above simple illustration is just an example, but there are many cases in government and industry which are similar in nature.

> "Even under rigidly controlled systems, paper work often begets more paper work. Ambiguous, inaccurate, or tardy records foster explanatory or corrective documents. When paper work is uncontrolled, clerical empires quickly form. The burdensome cost of record making and the large percentage of errors prevalent in the finished product are caused by paper work's own complicated process. Ornate, overlapping, complicated routines are customary in the average administrative division." /3/

Buck Passing, Empire Building and Inefficient Bureaucracy

In the case of the exclusive systems design, an organization takes the approach:

$$IuMuP = n(I) + n(M) + n(P)$$

All steps of the systems in I, M, and P have been designed independently, as it is shown in Figure 1.

Also, a disjoint (subsystem) approach gives way to inefficiency and discourages team work. No one wants to accept any responsibility, rather there is buck passing, empire building, and inefficient bureaucracy. But in every organization, all departments are interrelated to some degree; no department can work independently of the other. There are certain operations which may be distinct to one department, e.g., the unshaded areas in Figure 2. Furthermore, it must be remembered that even unique operations of each department must contribute to the overall benefits of the firm. Otherwise, these functions will be eliminated.

A systems department must, therefore, design or redesign systems in a manner that will benefit the firm. If this is not done, the systems department does not fulfill its responsibilities. The time has come when systems departments must start to contribute more to the overall benefits of an organization, or the systems profession will not attain its full potential. Consequently, the systems department must consider the joint set (total systems) concept, i.e.

$$n(IuPuM) = n(I) + n(M) + n(P) - n(IxM) - n(IxP) - n(MxP) + n(IxPxM).$$

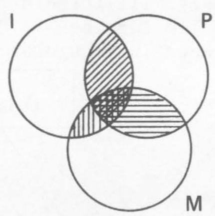


Figure 5

All duplications and triplications should be scrutinized for potential elimination.

Distinguishing Between Users and Suppliers

One author has made a very pertinent comment:

> "Substantial and permanent reductions in overhead costs are possible if users and suppliers of services work together in weighing costs and benefits, and identifying options and risks." /4/

This statement is true, but it must be pointed out that quite frequently it becomes very difficult to distinguish between the user and the supplier of services. If this happens, it may become difficult to cut costs. In the unshaded area of Figure 5, the users and suppliers can readily be distinguished. As an example, the supplier is Inventory Control, which forwards the purchase requisition to Procurement (user). But in the shaded areas, e.g., IxP, the distinction between user and supplier may not be readily seen. For example, which department is accountable for damaged parts which have been put on the production floor. Inventory Control may say, "Procurement is accountable, since Procurement must buy parts to specification." Procurement says, "Inventory Control should have informed Procurement that they, Inventory Control, received damaged parts." In the mean time, the production line has stopped, and the blame is passed from one department to the next. If the organization has, for example, eight departments or a number of divisions, the buck passing can be time-consuming and very expensive.

One of the authors was once assigned to analyze the "Vendor Evaluation Report" which Cost Accounting supplied to Procurement monthly. This report was produced by Cost Accounting at a cost of 24 man-hours per month. Cost Accounting insisted that the report incorporated all the information Procurement needed. On the other hand, Procurement wanted different kinds of data, but Cost Accounting was not willing to change the report. Therefore, Procurement used only certain sections of the report and worked 8 additional man-hours per month to revise the report according to its particular needs. After a lengthy discussion, Cost Accounting agreed to change the report, which still took 24 man-hours, but the 8 man-hours in Procurement were eliminated. Before the changeover, Cost Accounting was a 100% supplier and user, and Procurement (user) used only 60% of the report. After the report had been changed, Procurement was also a 100% user. Many kinds of these reports are being generated daily everywhere, and the systems department, unfortunately, is often a contributer to this inefficiency.

Solving a Problem: Everyone Wants to Get Involved

If there are three or more departments, e.g., AxBxCxDxE, there may be one supplier and four users, or three suppliers and two users, or three suppliers and three users, because one department may be a supplier and a user simultaneously. It may become almost impossible to define who should really be involved in solving a given problem; everyone wants to get involved. In this case, systems design or redesign may become nearly impossible. Most of the time, a compromise solution will be adopted which really does not optimize the system.

A Step-by-Step Approach

Some firms take the outlook: let's get one system fully operational first, and then the next. This approach can still be taken, and interdependent systems design preserved. The analyst must first consider the total system in detail and then break the total system into subsystems. When he takes this step, the analyst must make certain that the subsystems are compatible with each other. The first subsystem will set the parameters for the second system; the first and second subsystems will set the parameters for the third one, and so on until all systems have been designed. For example, a total system consists of three subsystems A, B, and C. The analyst, after considering the total system in detail, must start with system A, then B, and at last C (see Figure 6).

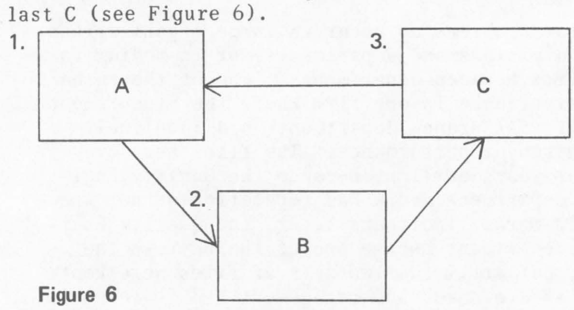


Figure 6

If the total system concept is considered A-B-C, only minor changes may be needed to complete the cycle.

Deciding Which Parameters to Use

If this is not done, e.g., subsystem B is first designed, then A, the question will arise which system should C follow, i.e., which parameters to use. If system C takes the parameters from B, then systems B and C are not compatible to A, as shown in Figure 7. In Figure 7, the arrow from C to A is in

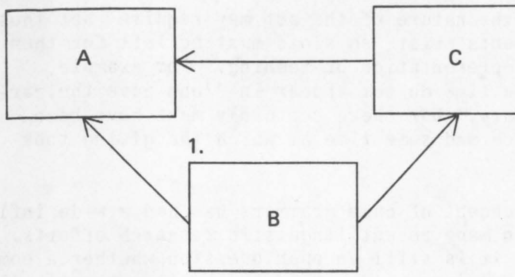


Figure 7

the opposite direction to the arrow of B to A, hence B and C are not compatible to A.

In Figure 8, sets A and B, and B and C are joint, but the sets A and C are disjoint. Therefore, there will again be an informal system which bridges systems A and C.

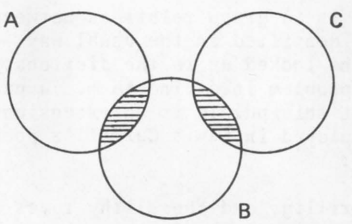


Figure 8

By using the counting principle for sets for Figure 8:

$$n(AuBuC) = n(A) + n(B) + n(C) - n(AxB) - n(BxC)$$

i.e., there will be duplications and triplications in the total system.

Options for Designing Compatible Systems

There are four options open to the systems analyst:

1. Redesign system A to make the system compatible to systems B and C, or
2. Redesign systems B and C to make them compatible to A, or
3. Start all over again, from A-B-C, or
4. Constant systems redesign, i.e., some systems will be always outdated, and hence a permanent staff is needed to keep updating systems.

If we consider the first option, redesign system A to make the system compatible to systems B and C, Figure 8 shows that systems (sets) A and B, and B and C are compatible, but systems A and C are disjoint. Hence, to join all three sets there will be systems redesign for the area AxC and the area AxBxC (see Figure 9), which may be a major task. The same principle holds for the second option.

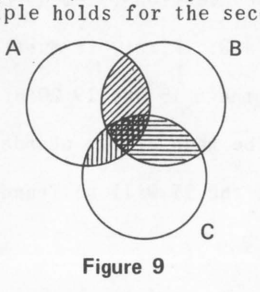


Figure 9

Using the fourth option, constant systems redesign, which is the exclusive system approach shown in Figure 1, the three systems, just like a jigsaw puzzle, will never fit together. Or if any two systems are compatible, the third system will never be congruous with the other two systems (see Figure 10).

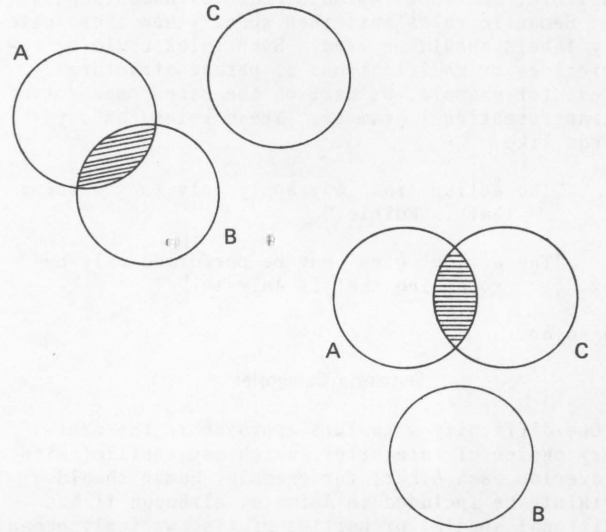


Figure 10

As it has been shown, a lot of extra time and money will be spent when information systems function with the subsystem (disjoint sets) concept.

Organizing for Interdependent Systems

The systems manager must consider the overall organization, i.e., IuPuM. The project leader must give special attention to the area of IxPxM, since all three departments interact in this region and most of the money can be saved. An analyst must spend most of his time and concern in the areas of IxP, IxM, and MxP, since an overlapping of functions, work, decisions, and forms may occur at this stage. A junior analyst may start to work in the exclusive areas of any department, for in these areas only department-wide systems are designed. Only one manager has to decide whether to accept or reject the system.

The Person Must Fit the Job

In the area IxP, the managers of Inventory Control and Procurement can decide if the system will be accepted or rejected. The analyst may have some negotiating to do, and most likely a compromise solution will be adopted.

In the region of IxMxP, even more discussions, meetings and negotiations will be needed, since three managers share the authority to accept or reject a system. What may be an ideal system for one manager may be the worst for another. In this case, the project leader must be an excellent mediator. The job description, as outlined, depends on the complexity and structure of the organization. The person must fit the job.

Increased Complexities

The model can be expanded to any number of divisions or departments (sets). The complexities increase as the number of departments increase, and it will be more and more difficult to determine which functions and decisions are duplicated, tripled, quadruplicated, and so on. *(please turn to page 19)*

Raphael – *Continued from page 10*

One frequently used technique simply assigns a word, as part of its dictionary definition, a list of all the semantic categories to which it belongs. These categories include classifications such as Countable, Abstract, Animate, Edible, Human, and so on. Semantic rules must then specify how these category labels should be used. Such rules could be restrictions or modifications of phrase-structure rules: for example, as part of the base component of a transformational grammar. These rules can say things like,

> "The action 'eat' may apply only to something that is Edible,"
>
> "The action 'give' may be performed only by something that is Animate,"

and so on.

Semantic Categories

One difficulty with this approach is the arbitrary choice of categories, which may conflict with or overlap each other; for example, Human should certainly be included in Animate, although it has additional special properties of its own (only human animals tell jokes, keep pets, start wars, and so on). A tree structure based upon inclusion provides one way of organizing a set of semantic categories; that is, each node of the tree is the name of a category that includes all the categories below it.

Any attempt to construct and use such a tree, however, introduces new difficulties. We keep running into a problem: Which of the many possible ways of dividing a category should be used first? For example, should "Human" be subdivided first into Male/Female, Adult/Child, Healthy/Sick, Rich/Poor, Black/Yellow/White, or what? Each such division would result in a differently structured subtree, and would produce some special simplifications and some other special complexities in the necessary semantic rules.

Inclusion is only one criterion for organizing a tree of semantic categories. Other organizing principles could be used also, to produce additional trees. ....

Case Grammars

A different approach to representing sentence meanings, called case grammars, is preferred by many linguists. In this approach, every activity has a whole family of implicit participants. Each of these participants has a particular role in the activity. For example, the action of giving always requires, as associated participants, a giver, a receiver, and a thing being given. (These roles are called cases, because they correspond roughly to the syntactic cases — nominative, dative, accusative, and so on — of classical grammatical theory). Because each act of giving also takes place at some particular place and time these parameters are also included as participants. In general, this theory proposes that each sentence containing an action — usually specified by the verb — can have its meaning represented by a list of the participants in the action and their associated roles. Moreover, only a small number of possible roles — probably less than ten, including the agent, mechanism, source, place, time, and so on — are needed to describe the action fully. Even though all the participants may not appear explicitly in the sentence describing an act, the nature of the act may require that those participants exist, so slots must be left for them in the representation of meaning. For example, place and time do not appear in "John gave the carrot to Mary," but there certainly must have been some place and some time at which the giving took place.

The concept of case grammars has had a wide influence upon many recent linguistic research efforts. However, it is still an open question whether a computer system can be designed that will automatically perform the necessary case analysis to extract such representations of meaning from English text. .....

(To be continued in next issue)

GIZZMO

The puzzle is to grasp relations between things that are not identified in the usual way — their names cannot be looked up in the dictionary — and then solve a problem involving them. A case could be argued that this puzzle is an extension of the principle displayed in Lewis Carroll's poem "Jabberwocky":

> 'Twas brillig, and the slithy toves
> Did gyre and gimble in the wabe; ...

In the following puzzle all the special context nouns have been replaced by numbers. Each number stands for the same noun throughout. "A" stands for "A" or "AN." "32s" stands for the plural of the noun designated by 32.

The (or a) solution will be published in the next issue. Any other solution reported by a reader will also be published.

Hint: To obtain one solution, the entire set of nouns in alphabetical order can be found on page 25. The puzzle then simplifies to fitting them in.

GIZZMO 769

The 1 is a 2 with a 3 of 4. About 5 percent of the 6 of the 1 has a 7 of more than 8 9s in 10 in some 11s. The 12 of the 13 6 through the 3 occurs in 11s but the 13 makes up only about 14 percent of the 1's 6. The 1 is a 15 2 with a partial 16 7. Is this a common situation? Can we expect other 2s to have a 3? If they do, will the 3 always be of 4 as on the 1 or is a 2 4 3 a rare occurrence?

To answer these questions, let us consider the requirements for a 3. First, it must be made of a 17 that is 16 at the 6 18 and 19 20 of the 2. Second, the 17 must be made of 21s abundant in the 22, so that enough of the 17 will be found on the 2 to form a 3.

Thompson — *Continued from page 13*

But what disturbs those citizens who followed the Watergate revelations closely is the use of special sections of the IRS for collecting information about citizens, including some congressmen, for political and other nontax purposes. Invasion of property, spying, wire tapping, bugging, personal surveillance, breaking and entering, and other techniques were used by IRS agents. In one case, the IRS provided agents to work with Justice Department agents to plant microphone bugs on a district attorney. IRS agents, operating radio receiving devices near the home, recorded conversations aimed at framing the citizen for bribery. The citizen (District Attorney Jim Garrison) was very nearly convicted at a trial using this IRS "entrapment" information. The Justice Department special strike force headed this criminal action. J. Edgar Hoover and the FBI had refused to participate. In this case, the blackest of the three agencies was the Justice Department, trailed closely by the IRS, with the FBI coming out relatively clean. /5/

The Motivation of the Invaders

What has motivated these privacy invaders? In a majority of cases, one all-pervading slogan describes it, "anticommunism." The leftist people and groups have been suspected of being communists or controlled by communists or of actions that would lead to communism taking over America. Even today, there are plenty of Americans who "see communists everywhere," and whose motivations in "law enforcement" (or whatever) are basically anticommunist in nature. Many of these people firmly believe that invasion of privacy is not only permissable, but absolutely essential if we are to prevent the communists from taking over. Somehow, communists, in their view, are nearly always black, young, drug users, long-haired, leftist, and more recently, female.

The FBI Cointelpro Activity

The FBI Cointelpro activity, which many believe is still going on, was basically an anticommunist project. J. Edgar Hoover was the fiercest anticommunist of all after Joe McCarthy died. The use of provocateurs, one of Cointelpro's weapons, was widespread, much more widespread than has yet been realized by most of us. We will eventually learn that most of the bombings, disturbances, murders, and mayhem attributed to the Far Left was achieved by FBI and other police provocateurs. Readers of such underground publications as Counterspy /6/ are already aware of this, and are aware of the fact that these illegal acts are continuing.

The Resources of the Invaders

What are the resources of these government invaders? They are vast. No one really knows, even now, how much money has been available to the CIA and FBI for these illegal, invading, activities. It is certainly well known to computer people that the technical resources, including computers and software people, of the army, FBI, CIA, NSA, and IRS overshadow all other resources in the computer field. In effect, it can be stated that the resources of these agencies are basically unlimited. If they choose to invade the privacy of anyone or any group and are not stopped, they can and will do it. They can go after information, wherever it may be, using a variety of technologies. The information certainly does not have to be all in one place; that makes no difference at all to them. Money is no object. People's time and other resources are no object. They will do it regardless.

Electronic Fund Transfer Systems: Now Being Added

EFTS and the invasion of privacy issue are superimposed on this vast history of illegal and clandestine privacy invasions. In comparison to what has already happened, with no EFTS networks of any size and no centralized files of any consequence, the future incremental change in invasion of privacy that might occur because of centralization of EFTS files and networks will be relatively insignificant.

System Safety Designs?

Safety approaches in EFTS software designs to prevent invasions will be useless against the resources and power of the invaders, as they have been constituted up to now. Can you imagine trying to devise a software-hardware combination that would freeze out the CIA, FBI, and IRS, if they were not stopped in other ways from tapping files?

Stop Fooling with "Invasion" — Concentrate on Dealing with the Invaders

The message to computer people is this: stop fooling around with the invasion of privacy problem as though it is a technical issue. Turn yourselves into social protesters, and join those who are trying to expose and control, or eliminate the ugly invaders. Do not worry about invasions of privacy, worry about the invaders and their motivations and resources. Cut them off where it hurts most, at the source of their strength, by complete public exposure. Convince computer people on the inside of these invading organizations that they should expose the activities from the inside. A bright light shining on any kind of rat or vermin usually makes it go elsewhere.

This is the pivotal place where the efforts of the privacy commission, the ACM, SIGCAS, AFIPS, and others concerned about the issue of privacy should be concentrated.

References

/1/ Paul Armer, "Electronic Funds Transfer Systems and the Consumer," Computers and People, vol. 25, no. 6, June 1976, pp. 8-9, 19.
/2/ Workshop on EFTS Technology, a seminar of the National Commission on EFTS, Washington, D.C., June 15, 1976.
/3/ Richard E. Sprague, "The Invasion of Privacy and a National Information Utility for Individuals," Computers and Automation, vol. 19, no. 1, January 1970, pp. 48-49.
/4/ At Fort Holobird, Maryland.
/5/ Richard E. Sprague and Edmund C. Berkeley, "The Attempted Framing of Jim Garrison," Computers and Automation, vol. 22, no. 12, December 1973, pp. 23-29.
/6/ Counterspy, a quarterly publication of The Fifth Estate, Washington, D.C.

Sadek and Tomeski — *Continued from page 17*

Footnotes

/1/ Victor Lazzaro, Systems and Procedures: A Handbook for Business and Industry (Englewood Cliffs: Prentice-Hall, 1968), p. 215.
/2/ Victor Lazzaro, ibid, p. 215.
/3/ Victor Lazzaro, ibid, p. 229.
/4/ John L. Neuman, "Make Overhead Cuts that Last," Harvard Business Review, May-June 1975, p. 116.

SOME NOTES ON GRAPHING

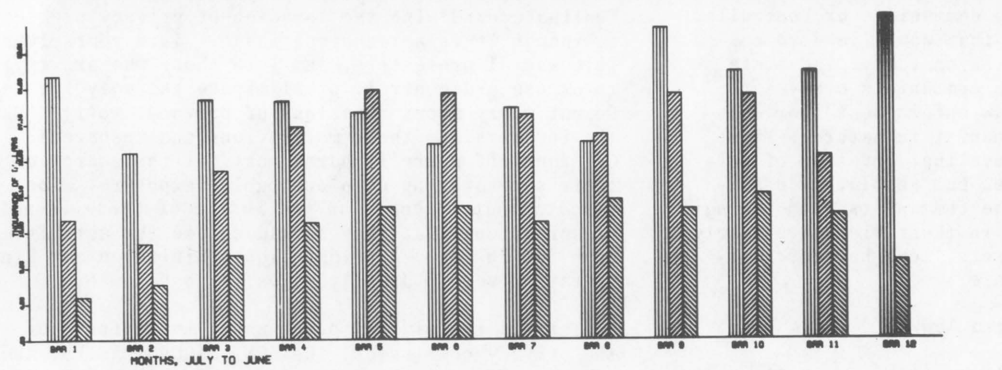
Stan Wall
Department of Computer Science
California State University, Chico
Chico, CA 95929

The cliche, "A picture is worth a thousand words", describes the value of a graph. One simple graphical representation of data can convey innumerable concepts concerning facts, not only of an element's value or size, but also the interrelationships of the elements of a graph, the trends of the given elements, their weaknesses and strengths — as well as other principles that numbers in themselves cannot convey.

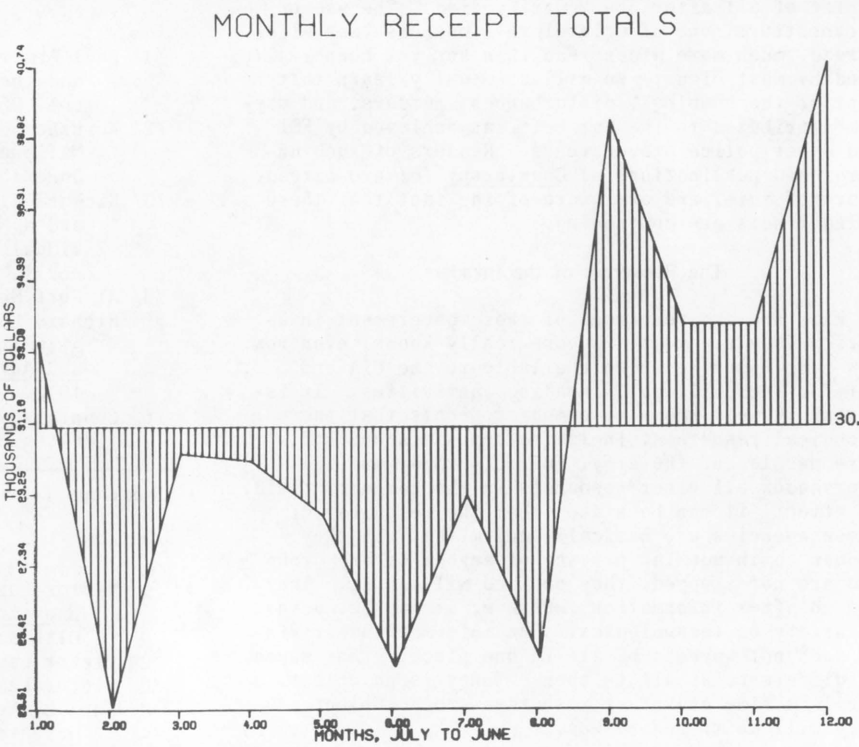
These concepts can then be observed and interpreted in a variety of ways by different individuals. A top management planning function might observe in a graph, the trend of sales of a process, and accordingly base its scope of planning. The same graph may be observed by a budget manager who would vividly note the dollar amounts depicted by the graph. In essence, the graph represents a basic fact or set of facts, which can be interpreted in many different ways by many different levels of view.

Several of the graphs shown here clarify this concept of different levels of meaning in graphical representation. The graphs depict monthly disbursements and receipts. For example, the grouped component bar graph gives a record of disbursements for each month — but it also clarifies the trends in disbursements, the relationship between the subsets within each element — plus a concept of the total scope of disbursements and receipts. It would be a challenge for most individuals to describe these amounts and indicate all the interrelationships of the data in just a thousand words. Further, I do not believe that words can as clearly convey precise information and communicate this information as readily as numbers (particularly when the numbers are graphed, and in several varied presentations).

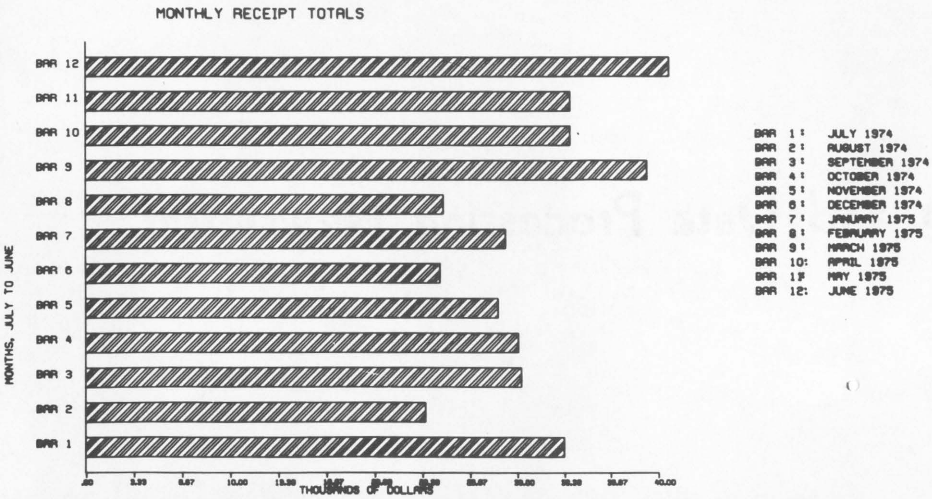
The automation of the graphing task is not an attempt to achieve a more complex or detailed presentation, but rather to relieve the individual from the clerical function of scaling the data, constructing lines, and all the other aspects of a graph. By utilizing the graphic system as a tool, the user can obtain the desired results simply by supplying the data to the graphing routines, never having to personally perform further calculations. Computer-aided graphing is an indispensable, necessary, interdisciplinary tool.



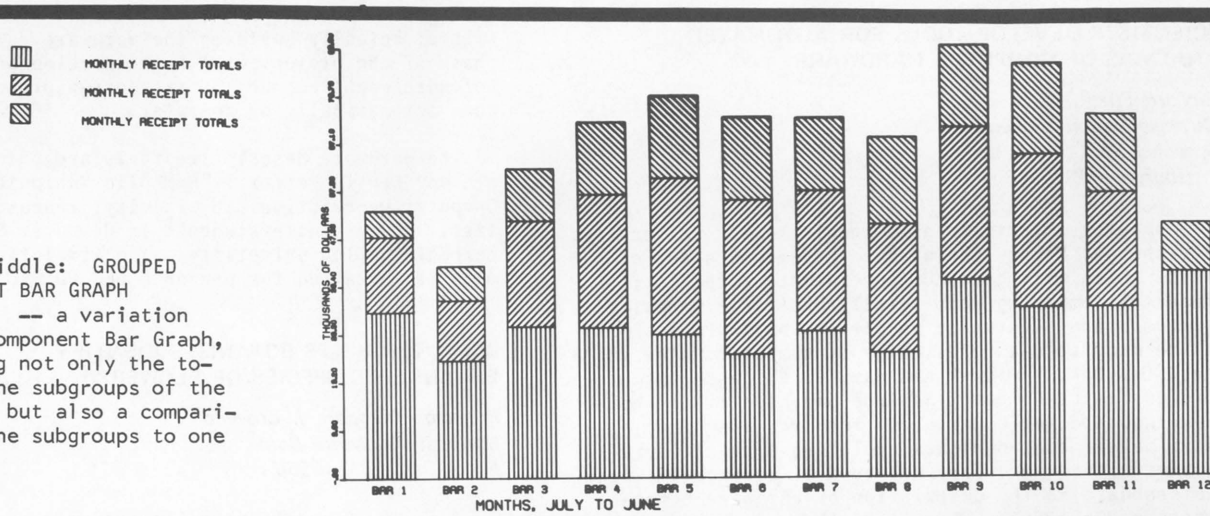
Above: GROUPED COMPONENT BAR GRAPH A variation of the COMPONENT BAR GRAPH, depicting not only the total of the subgroups of an element, but also a comparison of the subgroups to one another. The individual who studies the graph gains a much clearer picture of the subelements by this grouping arrangement.



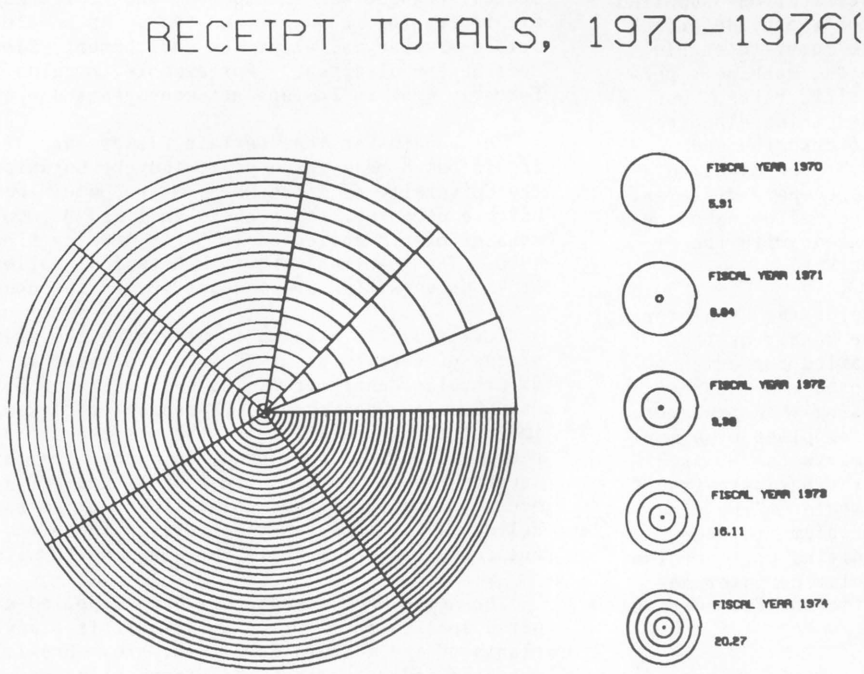
Right: SILHOUETTE LINE GRAPH A form of line graph, showing how, over time or distance, a variable differs from some constant value.

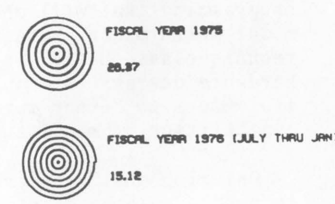


Left, top: HORIZONTAL BAR GRAPH — a variation of the basic bar graph. "It is my opinion that the shaded graphs have a strength that the blank bar graphs do not possess, giving a 'weighted' effect to the graph. Here monthly receipts are drawn on the plotter at the right of the graph. The work shown here is a photographically reduced version of a much larger graph."



Right, middle: GROUPED COMPONENT BAR GRAPH — a variation of the Component Bar Graph, depicting not only the total of the subgroups of the element, but also a comparison of the subgroups to one another.





Bottom, left: A new form of PIE GRAPH

This graph utilizes only arcs, and was fairly straightforward to program and plan.

Computing and Data Processing Newsletter

SCIENTISTS DEVELOP TOOLS FOR AUTOMATED ANALYSIS OF COMPUTER HARDWARE

Richard Puz
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

Computer scientists at Carnegie-Mellon University, Pittsburgh, Pennsylvania, have developed a set of tools which will give EDP specialists a realistic means of comparing commercially available computer systems, and eventually enable new hardware designs to be generated automatically by computer. The scientists, Dan Siewiorek and Sam Fuller, associate professors of Computer Science and Electrical Engineering, and Mario Barbacci, research associate, have developed techniques employing a special programming language, ISP, designed at Carnegie-Mellon, to standardize the description of hardware components and operating characteristics of computers.

"The technical manuals and specifications supplied by computer manufacturers often don't provide enough information, or data in comparable form, to enable EDP specialists to easily measure one machine's performance against another's on specific tasks," Dan Siewiorek said. "So we have worked to develop the programming that will enable us to describe and model performance characteristics." The modeling technique has enabled the group to compare different hardware designs through simulation, allowing potential users to better match their needs with the capabilities of machines on the market.

Entering an ISP description causes the simulator to handle data in exactly the same manner as the described machine, allowing a detailed performance test in which processes are monitored and recorded step by step. The advantage of using the simulator, rather than testing actual models supplied by a manufacturer, is that standard benchmarks can be used in the testing process, allowing the hardware itself to be measured against a uniform scale. "The simulator allows us to track sample problems through every phase of each computation, giving us a precise means of finding out which particular computer designs are best suited and most efficient for certain kinds of tasks," said Siewiorek.

Since the simulation techniques provide a detailed analysis of machine operations, they can also be used as tools by manufacturers to test new designs without actually building the hardware. The final phase of the project will be to develop additional software enabling much of the design process to be done automatically by computer.

The hardware descriptive tools are one phase of a comprehensive effort, "Symbolic Manipulation of Computer Description," by faculty, research scientists, and graduate students in Computer Science at Carnegie-Mellon University. The simulation programming is formatted for use on a DEC PDP-10 computer.

COMPUTER HELPS BOTANISTS IDENTIFY ENDANGERED SPECIES OF FLOWERS

Professor Theodore J. Crovello
University of Notre Dame
Notre Dame, IN 46556

A study of specimens collected over the past 175 years indicates that certain flowers in the United States, considered "endangered" and "threatened," are increasing in number and type. As people move into new areas and alter the environment, flowers decline and disappear. For example, orchids that formerly grew in Indiana not many years ago are gone.

The conclusion that certain flowers are disappearing is based on a three-year study by botanists at the University of Notre Dame, Notre Dame, Indiana. Using a computer, researchers analyzed 35,000 specimens of plants collected in North America since 1800. The specimens are from botanical collections at 75 universities and museums across the country.

"Our study is helping to determine the endangered status of certain plants," said Professor Theodore J. Crovello, chairman of the University of Notre Dame Biology Department. "Research, for example, indicates that relatives of the mustard are disappearing. On the other hand, we might find that a species thought to be extinct or disappearing has been mislabeled in some collections; or hadn't been collected for some years; or exists in only a few collections, but is still thriving in the wild."

The Notre Dame researchers have supplied endangered species information to scientists studying plants of the mustard family in Ohio. Preliminary research indicates that while some plant species in Ohio are threatened and endangered, others can be removed from the list. Analysis of certain spring flora, such as various types of spring water cress, shows some are disappearing while others are plentiful.

Another study focuses on all 2,500 different species of flowers found in Indiana. The Flora Indiana Project (FLIP) is attempting to determine which plants are endangered or threatened (a less serious category).

The biologists built their files of information by entering names of plants, when and where found, environmental conditions at time of collection, and other information. The computer produces reports by states and counties within states, showing numbers of plants gathered by century and decade since 1800.

"This is more than just an esthetic loss, because man benefits from certain plant species," said Dr. Crovello. "For example, many important medicines come from plants, yet most plants have not even begun to be analyzed for products useful to people. Who knows what potentially valuable chemicals we are losing with every species that becomes extinct!

"We hope our work will help people set their priorities, and to make them and their elected representatives better aware of the consequences of their actions. For example, currently the Federal Office of Endangered Species has only two botanists to monitor the endangered status of the 20,000 kinds of plants that grow (or have been growing) in the United States."

The Notre Dame study of endangered species of flowers used an IBM System/370 Model 158 computer.

CHEMICAL LAB FILLS ORDERS MORE QUICKLY AND MORE ACCURATELY WITH COMPUTERIZED RECORD-KEEPING SYSTEM

Joe Pittman
Varian Data Machines
2722 Michelson Drive
Irvine, CA 92664

An automated record-keeping system at USS Chemicals, Pittsburgh, Pennsylvania, is helping laboratory technicians more quickly ascertain the kinds of analyses needed for each shipment of chemical products and insure that the required analyses are performed before a shipment is released. The Laboratory Services System (LASS) at USS Chemicals uses a minicomputer to enter orders, compile order lists, release shipments, and compile daily shipment reports.

The data base contains some 600 customer addresses, along with information regarding any special testing requirements, customer product codes, and whether a Certificate of Analysis must be shipped with the product. Previously, this information was entered in loose-leaf binders and retrieved manually.

Now an order from a salesman can be entered into the computer within seconds by typing a 10-digit customer code, the USS Chemicals order number, the requested shipping date, and the customer order number. With those four entries, the computer searches the address files and, in one second or less, displays the address associated with the customer code. After the operator verifies that the customer address is the correct one, the computer displays any requirements for a Certificate of Analysis and requests the products and weights in each order. Following that entry, the computer displays the USS Chemicals identification and the customer code, if any, for verifications with the actual order. Once the orders are stored in the computer, they can be easily cancelled or altered by an operator.

A variety of lists, which previously had to be compiled and typed up by a clerical worker, can be displayed in seconds on a CRT terminal or printed out on a high-speed printer. Lists of all the orders in the file may be displayed chronologically by shipping date, showing the order number, shipping dates, products, and the first line of the customer address. A list giving all the orders scheduled to be shipped between any two dates can be displayed showing the order date, quantity, customer address, and the total quantity for each product between the dates given. Lists may also be displayed showing all the orders in the file for a given customer or a particular product.

A sample from every product shipment which is to leave the plant must be analyzed by USS Chemicals technicians to ensure that the product meets sales specification requirements before it can be released for shipment. For example, all shipments of plasticizer, a plastic softening agent, are tested for appearance, odor, specific gravity, water content, acidity, and APHA color.

Additional analyses are often required to determine whether the product meets special customer specifications. A wire manufacturer may want an analysis of a plasticizer shipment to determine the product's direct current resistance; another manufacturer may want the product tested for viscosity. LASS has enabled USS Chemicals technicians to exercise tighter control over shipping procedures. Approximately 200 special customer specifications are maintained in the LASS data base.

Previously, it was possible for a technician to miss the customer specifications and release the shipment on the basis of the sales specifications only. Now, this is virtually impossible. Further, the ability to remotely produce the Certificate of Analysis has greatly speeded up the process of releasing a shipment.

The minicomputer has also streamlined the production of daily half-sheet shipping reports. These reports, giving product, customer, test results, etc., on each of the previous day's shipments, go to the sales, material handling, and production departments. Before LASS, a secretary spent 70 to 80 percent of her time typing up the half-sheet reports. It now takes about 20 minutes each morning to produce them.

The USS Chemical minicomputer is a 32K Varian V72.

COLLEGIANS OVERCOME WRITING HANDICAPS IN A COMPUTER-SUPPORTED ENGLISH LAB

Jeff Hinger
San Antonio College
1300 San Pedro
San Antonio, TX 78284

More than half the students starting college in the United States this year cannot write a complete English sentence. National college entrance tests show that many don't know the difference between a noun and a verb. To help overcome this deficiency, San Antonio College, San Antonio, Texas, operates a special remedial English laboratory where thousands of students combat writing handicaps by practicing on a computer.

The laboratory, a supplement to traditional classroom methods, combines audio-visual teaching with individualized, computer-assisted instruction. Working with typewriter and video display terminals, the students review and test themselves on the basic concepts of sentence structure and usage. Within a short time, most are able to incorporate these concepts into their own writing and move on to earn top grades in freshman composition classes.

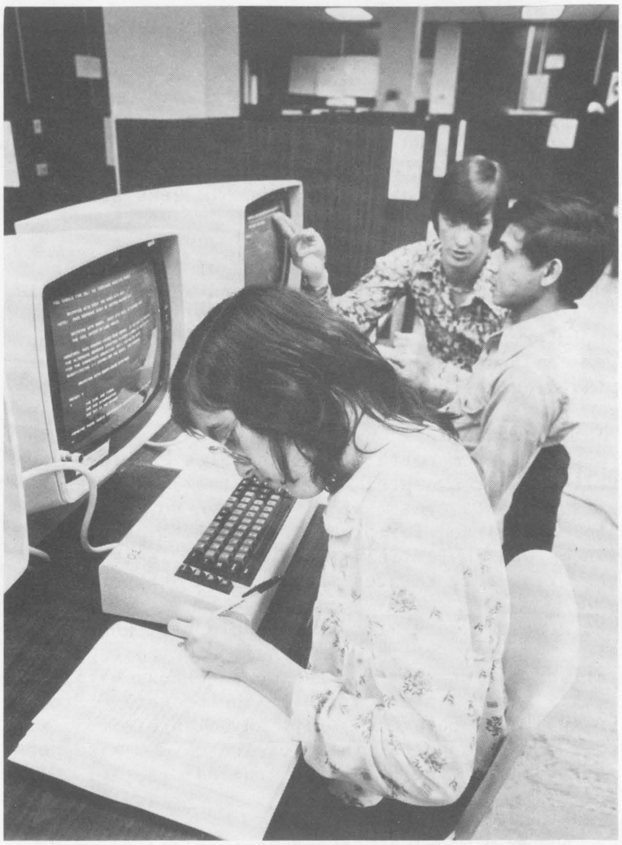
Students who enter San Antonio College with low college entrance test scores (50 percent in 1975) are required to enroll in basic English, a remedial course, before they can qualify for classes in freshman composition. The students supplement their classroom work with at least one hour a week in the multimedia lab.

Linked to the college's computer, a bank of terminals forms the hub of the lab. Working in an individual carrel, each student reviews instructional material and practices his writing skills. As he progresses, color slides flash on the carrel's screen, narrated by synchronized audio tapes.

Periodically during the session, the student moves to one of the computer terminals for test exercises. To begin each computer-assisted exercise, the student types his full name. The computer responds with directions on how to complete the test. When he answers a question correctly, the student is rewarded with personalized praise, "Good work, Marty." If he makes an error, the computer prints or displays the right answer, explaining why it is correct. At the end of each exercise, the student's performance is scored instantaneously. This gives him immediate feedback and a chance to observe a direct relationship between his efforts and the results.

The learning process is self-paced. Slow students can spend time mastering each phase of their work, while fast learners make rapid progress. One student recently completed a whole semester's work in five weeks.

Student tutors and proctors lend assistance when necessary. The lab also offers a special remedial English program for students with impaired hearing, substituting a visual text for auditory information recorded on tape. The computer, however, does not replace the instructor; rather, the system handles repetitive drill work at a pace that's best for each individual student and frees the teacher for more creative tasks.



At work in the English Multimedia Laboratory at San Antonio College, these students are learning to write clear, grammatically correct sentences with assistance from a computer. Using video display or typewriter terminals, they communicate directly with the computer, reviewing and testing themselves on the basics of written expression. The computer-supported lab has helped thousands of students move on to earn good grades in English composition and improve their overall college performance.

The achievement level of students who complete the English lab program is dramatic. Before the introduction of computer assisted instruction, nearly half of the remedial English students failed more advanced classes in freshman composition. Now, nearly 75% earn grades of "C" or better. Ramifications are spreading beyond the English department. Professors in other disciplines report that the students are much better equipped now to communicate their knowledge of subject matter, and thus are earning better grades.

The English lab uses an IBM System/370 Model 158 computer.

A COMPUTER THAT NEVER FORGETS A FACE

Office of Public Information
Case Western Reserve University
Cleveland, OH 44106

A computer that never forgets a face? Why not? Engineers at Case Western Reserve University, Cleveland, Ohio, are programming a computer so that it can not only see, but recognize and remember faces.

The project began six months ago, with the primary goal of facial recognition and re-identification. Human subjects were shown portraits and asked to describe them to a computer, using features from a list given to them. The computer then searched a population of stored descriptions for the best fit to the description furnished by the subject.

Now going a step further, explains Professor Leon Harmon of the Biomedical Engineering Department, who is the head of the project, "We are trying to get a computer to look at a human face and tell you what it sees." Photographs are being taken of subjects, mostly Case Reserve students and faculty. An artist has been employed to transfer traces of the profile images onto sheets to be fed into the computer. The machine then extracts facial characteristics and files them into its memory banks. The traces are fed into the computer by a scanner. As the computer scans these traces it is asked to recognize those subjects with certain facial characteristics. It should be able to choose not only profiles with similar features, but also those that differ.

The next step will be to find out if the computer will be able to recognize and re-identify the features in its file. Harmon comments, "We're working towards automated personnel identification. Basically, what we're asking is, can we, for a reasonably sized population such as a group of company employees, get a machine technology to accurately recognize a face rather than having the employee use an I.D. card?"

An additional basic research thrust being explored is getting computers to see accurately, recognize what they see, and most importantly, finding out how well computers can make sense of what they see.

COMPUTER AIDS IN FOREST FIRE DETECTION BY PREDICTING THE LOCATION AND TIME OF FIRES

Rudy Platiel
The Globe and Mail
Toronto, Ontario
M5V 2S9, Canada

When pilots take off from Northern Ontario airfields on their daily mission to spot new forest fires, a computer in Toronto will already have told them how many new fires they can expect to find, where to look for them, and at what time. The detection routing program is only one of a number of ways in which the computer is playing an increasingly large role in the annual forest fire battles by the provincial Ministry of Natural Resources.

Already the computer has virtually wiped out the need for the old forest fire observation towers. In 1967, there were 356 observation towers in Ontario; now there are about eight.

Ontario's use of computers in fighting forest fires is the most advanced in Canada, ministry officials say. Each day, weather information is fed into the computer which then generates a profile of how fires will behave and react, as well as how severe the fuel factor will be.

"In the past, that was done manually. We can now run that total provincial program in a matter of a few minutes, where in the past each person out in the field, once he got his weather, had to work through all the tables," said ministry official Mr. Goodman.

The computer also supplies a forecast for the next day, keeps track of manpower, aircraft and equipment (down to the number of axes) throughout the province, and eventually may even direct equipment and men to certain areas on the basis of predicted fire outbreaks.

The Ontario ministry has a network of lightning sensors at its 150 weather stations throughout the province. These sensors record the number of lightning discharges within a 20-mile radius and, from a formula that considers weather and fuel conditions, predicts the number of fires to be expected in a certain area. Early detection of new fire outbreaks is crucial in the annual forest fire war.

The majority of fires are caused by people, and, in addition to weather and lightning discharges, data on the activities of people are also fed into the computer. According to Mr. Goodman, "The computer comes back and tells us with a fair degree of accuracy, how many fires we can expect for each area and where they might be.

"Then there is another program we can run, given those fires we are going to have in the next 24 hours, here is where you should fly your airplane and at what time, to find them. So it helps us plan our detection routing each day. It saves us money and we find the fires quicker," replacing the old random air search of a few years ago.

The next step is the development of a computer program that will be able to provide fire fighters with a prediction on how a specific fire will develop under the forecast conditions.

COMPUTERIZED X-RAY MACHINES TO AID CANCER THERAPY

Mitch Waldrop
University of Wisconsin
610 Walnut Street, Rm. 1215
Madison, WI 53706

A new diagnostic X-ray method is promising to produce safer and more effective radiation therapy for cancer. The technique, called computerized tomography (CT), is being adapted by physicists and radiologists at the University of Wisconsin, Madison, Wisconsin, to accurately locate a tumor and its boundaries within a patient's body. With this improvement in localization, radiation therapists will be able to focus cancer-destroying beams precisely on the tumor, thus minimizing the chance of radiation damage to healthy organs nearby.

Computerized tomography takes its name from "tomos," the Greek word for a cut or thin slice. CT machines produce X-ray images of a cross section of the body by taking hundreds of ordinary X-rays, each from a slightly different angle. A computer then correlates all this X-ray information into a single cross-sectional image.

CT machines have come into wide clinical use since their development in England a few years ago. Hailed as one of the most important scientific advances in decades, they are giving physicians and medical researchers a clear view into regions of the body that are blind spots on ordinary X-rays.

Unfortunately, says Dr. William Caldwell, associate director of the university's Clinical Cancer Center, the computer programs of commercial CT machines are not designed for the kind of high-precision work which would be useful in radiation therapy. The reconstructed image might show the kidney boundaries distorted by one-half inch, for example. Thus, one part of the radiation therapy research group is developing computer programs that can give much more accurate images.

Another part of the group is systematically evaluating the performance of a recently acquired, whole-body CT machine. Patients with operable tumors will be asked to have a CT scan; the findings of that study will then be compared to what is found by the surgeons.

The Wisconsin researchers are working under a grant from the National Cancer Institute.

Hint for GIZZMO 769

ATMOSPHERE COSMOS COVER DEPTH EARTH

ELEMENT LAND LIQUID MILE OCEAN PLACE

PLANET PRESSURE PROJECTION SEVENTY SIX

SOLID SUBSTANCE SURFACE TEMPERATURE THIRTY

WATER

Berkeley – *Continued from page 3*

Table 7: APL Program

```
      $DL WAGE REPORT
[1]   'DUES AMT: '; V1 ← 3.00
[2]   'TAX BASE: '; V2 ← 150.00
[3]   'TAX RATE: '; V3 ← .1100
[4]   V35 ← 1
[5]   'SS RATE: '; V4 ← .0585
[6]   V5 ← $QD $NL $QD ← 'ENTER GROSS PAY'
[7]   V6 ← V5 - V2
[8]   'INCOME TAX: '; V7 ← V6 $ML V3
[9]   'FICA: ';  V8 ← V5 $ML V4
[10]  'CLUB: ';  V9 ← V1 $ML V35
[11]  V10 ← V7 + V8
[12]  'DEDUCTIONS: '; V11 ← V10 + V9
[13]  'NET PAY: ';  V12 ← V5 - V11
[14]  $DL
```

Table 8: FOCAL Program

```
1.01 C WAGE REPORT
1.02 ASK V1
1.03 ASK V2
1.04 ASK V3
1.05 ASK V35
1.06 ASK V4
1.07 ASK V5
1.08 SET V6 = V5 - V2
1.09 SET V7 = V6 * V3
1.10 SET V8 = V5 * V4
1.11 SET V9 = V1 * V35
1.12 SET V10 = V7 + V8
1.13 SET V11 = V10 + V9
1.14 SET V12 = V5 - V11
1.15 QUIT
```

Table 9: COBOL Program

```
IDENTIFICATION DIVISION.
PROGRAM-ID. WAGE-REPORT.
AUTHOR. ERIC ALBERT.
INSTALLATION. BERKELEY ENTERPRISES.
DATE-WRITTEN. 1976.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-360.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT CARD-FILE ASSIGN TO SYS026-UR
    SELECT PRINT-FILE ASSIGN TO SYS028-UR
DATA DIVISION.
FILE SECTION.
FD  CARD-FILE
    RECORD CONTAINS 80 CHARACTERS
    LABEL RECORDS ARE OMITTED
    DATA RECORD IS CARD.
01  CARD.
    05 INFO    PICTURE 9(4)V99.
    05 FILLER     PICTURE X(74).
FD  PRINT-FILE
    RECORD CONTAINS 133 CHARACTERS
    LABEL RECORDS ARE OMITTED
    DATA RECORD IS OUT.
01  OUT.
    05 FILLER     PICTURE X(133).
WORKING-STORAGE SECTION.
01  ITEMS.
    05 ITEM-5       PICTURE 9(4)V99
 VALUE ZERO.
    05 ITEM-6       PICTURE 9(4)V99
 VALUE ZERO.
    05 ITEM-7       PICTURE 9(4)V99
 VALUE ZERO.
    05 ITEM-8       PICTURE 9(4)V99
 VALUE ZERO.
    05 ITEM-9       PICTURE 9(4)V99
 VALUE ZERO.
    05 ITEM-10       PICTURE 9(4)V99
 VALUE ZERO.
    05 ITEM-11       PICTURE 9(4)V99
 VALUE ZERO.
    05 ITEM-12       PICTURE 9(4)V99
 VALUE ZERO.
01  PRINT-FORMAT-LINE-1.
    05 FILLER       PICTURE X(10)       VALUE
 SPACES.
    05 ITEM-7-OUT      PICTURE Z,ZZ9.99.
    05 FILLER       PICTURE X(5)      VALUE
 SPACES.
    05 ITEM-8-OUT      PICTURE Z,ZZ9.99.
    05 FILLER       PICTURE X(5)      VALUE
 SPACES.
    05 ITEM-9-OUT      PICTURE Z,ZZ9.99.
    05 FILLER       PICTURE X(5)      VALUE
 SPACES.
    05 ITEM-11-OUT      PICTURE Z,ZZ9.99.
    05 FILLER       PICTURE X(5)      VALUE
 SPACES.
    05 ITEM-12-OUT      PICTURE Z,ZZ9.99.
    05 FILLER       PICTURE X(63)      VALUE
 SPACES.
01  CONSTANTS-SECTION.
    05 ITEM-1       PICTURE 99V999       VALUE
 03.000.
    05 ITEM-2       PICTURE 9999V999
  VALUE 0150.000.
    05 ITEM-3       PICTURE 9V99999
 VALUE 0.11000.
    05 ITEM-35       PICTURE 99V9       VALUE
 01.0.
    05 ITEM-4       PICTURE 9V99999
 VALUE 0.05850.
PROCEDURE DIVISION.
BEGIN.
    OPEN INPUT CARD-FILE, OUTPUT PRINT-FILE.
EXECUTE.
    PERFORM READ-ITEMS.
    SUBTRACT ITEM-2 FROM ITEM-5 GIVING
 ITEM-6.
    MULTIPLY ITEM-6 BY ITEM-3 GIVING
 ITEM-7.
    MULTIPLY ITEM-5 BY ITEM-4 GIVING
 ITEM-8.
    MULTIPLY ITEM-1 BY ITEM-35 GIVING
 ITEM-9.
    ADD ITEM-7, ITEM-8 GIVING ITEM-10.
    ADD ITEM-10, ITEM-9 GIVING ITEM-11.
    SUBTRACT ITEM-11 FROM ITEM-5 GIVING
 ITEM-12.
    PERFORM PRINT-DATA.
    GO TO QUIT.
READ-ITEMS.
    PERFORM READ-CARD.
    MOVE INFO OF CARD TO ITEM-5.
READ-CARD.
    READ CARD-FILE RECORD AT END
    DISPLAY '-END OF FILE REACHED',
    GO TO QUIT.
PRINT-DATA.
    PERFORM PRINT-A-LINE-1.
PRINT-A-LINE-1.
    MOVE ITEM-7 TO ITEM-7-OUT.
    MOVE ITEM-8 TO ITEM-8-OUT.
    MOVE ITEM-9 TO ITEM-9-OUT.
    MOVE ITEM-11 TO ITEM-11-OUT.
    MOVE ITEM-12 TO ITEM-12-OUT.
    WRITE OUT FROM PRINT-FORMAT-LINE-1.
QUIT.
    DISPLAY '1' UPON THE-PRINTER.
    CLOSE CARD-FILE, PRINT-FILE.
    STOP RUN.
```

GAMES AND PUZZLES for Nimble Minds – and Computers

For two more puzzles, see pages 11 and 18.

Neil Macdonald
Assistant Editor

It is fun to use one's mind, and it is fun to use the artificial mind of a computer. We publish here a variety of puzzles and problems, related in one way or another to computer game playing and computer puzzle solving, or to the programming of a computer to understand and use free and unconstrained natural language.

We hope these puzzles will entertain and challenge the readers of *Computers and People.*

NAYMANDIJ

In this kind of puzzle an array of random or pseudorandom digits ("produced by Nature") has been subjected to a "definite systematic operation" ("chosen by Nature") and the problem ("which Man is faced with") is to figure out what was Nature's operation.

A "definite systematic operation" meets the following requirements: the operation must be performed on all the digits of a definite class which can be designated; the result displays some kind of evident, systematic, rational order and completely removes some kind of randomness; the operation must be expressible in not more than four English words. (But Man can use more words to express it and still win.)

NAYMANDIJ 769

0	0	0	2	7	0	1	0	7	8	0	0	2	2	9	2	2	9	3	3
3	1	1	3	3	5	6	9	7	1	5	5	2	0	2	9	2	5	3	8
3	1	3	5	5	8	5	6	9	5	4	3	3	6	6	0	0	2	5	5
1	1	3	4	3	2	8	3	1	8	6	5	5	2	7	4	5	4	7	1
6	9	5	9	8	5	0	2	1	5	7	3	5	7	6	3	5	4	0	6
8	3	5	9	7	3	6	5	4	5	7	7	4	8	6	4	0	6	2	9
3	6	6	5	2	1	6	0	6	7	0	0	7	2	4	9	1	7	3	8
3	6	7	1	9	8	8	7	4	6	0	8	8	6	6	4	2	1	8	5
7	4	8	5	6	8	8	2	6	6	2	2	8	2	8	6	8	9	0	1
7	5	9	5	0	3	0	5	4	9	0	3	3	3	0	6	8	0	8	8

MAXIMDIJ

In this kind of puzzle, a maxim (common saying, proverb, some good advice, etc.) using 14 or fewer different letters is enciphered (using a simple substitution cipher) into the 10 decimal digits or equivalent signs for them. To compress any extra letters into the 10 digits, the encipherer may use puns, minor misspellings, equivalents like CS or KS for X or vice versa, etc. But the spaces between words are kept.

MAXIMDIJ 769

NUMBLES

A "numble" is an arithmetical problem in which: digits have been replaced by capital letters; and there are two messages, one which can be read right away and a second one in the digit cipher. The problem is to solve for the digits. Each capital letter in the arithmetical problem stands for just one digit 0 to 9. A digit may be represented by more than one letter. The second message, which is expressed in numerical digits, is to be translated (using the same key) into letters so that it may be read; but the spelling uses puns, or deliberate (but evident) misspellings, or is otherwise irregular, to discourage cryptanalytic methods of deciphering.

NUMBLE 769

```
                  S A F E T Y
              x   B R E E D S
              -----------------
                T G E D D Y R
              R B A T S E N
F = B       S R E S R R A
          S R E S R R A
          D S T N R N
      T N R E F D G
      -------------------------
  =   T E S E R Y B Y G S R R

            76809  2
```

We invite our readers to send us solutions. Usually the (or "a") solution is published in the next issue.

SOLUTIONS

NAYMANDIJ 768: Make diagonal of 9s.

MAXIMDIJ 768: The earth is a small planet.

NUMBLE 768: Every shoe pinches some feet.

Our thanks to the following individuals for sending us solutions: Russell Chauvenet, Silver Spring, MD: Naymandij 767; Maximdij 767; Numble 767 – Byung Sun Choung, San Diego, CA: Numble 767 – Frank DeLeo, Brooklyn, NY: Naymandij 767; Maximdij 767; Numble 767 – T. P. Finn, Indianapolis, IN: Maximdij 767; Numble 767 – Ronald C. Graves, Ashland, MA: Naymandij 767; Maximdij 767; Numble 767.

COMPUTER GRAPHICS AND ART

COMPUTER GRAPHICS and ART is a new international quarterly of interdisciplinary graphics for graphics people and computer artists. This new periodical is aimed at students, teachers, people from undergraduate and graduate institutions, researchers, and individuals working professionally in graphics. Its topical coverage is broad, embracing a variety of fields. It is useful, informative, entertaining, and current.

Send your manuscripts, papers, art, and ideas to:

Editor, *COMPUTER GRAPHICS and ART*
Berkeley Enterprises, Inc. – Chico Branch
555 Vallombrosa – No. 35
Chico, California 95926

Here Are Some Samples:

Learning Through Graphics
by Dr. Al Bork, University of California, Irvine, California
A ten-year forecast for computers, education, and graphics by a leading authority.

Art of the Technical World
by Dr. Herbert Franke, Munich, Germany
Computer art as the bridge between the two realms of art and leisure.

Expanding the Graphics Compatability System to Three Dimensions
by Richard F. Puk, Purdue University, Lafayette, Indiana
Design considerations for a user-oriented 3-D graphics system.

A Personal Philosophy of Ideas, New Hardware, and the Results
by Duane Palyka, University of Utah, Salt Lake City, Utah
The frame-buffer from Evans and Sutherland allows the artist to treat the computer as a paint and brush medium.

How to Build Fuzzy Visual Symbols
by Alex Makarovitsch, Honeywell Bull, Paris, France
A new approach to computer art and graphics by a computer scientist.

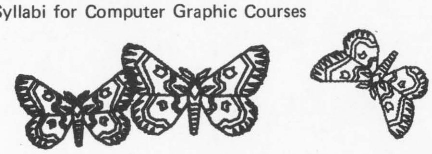
The State of the Art of Computer Art
by Grace C. Hertlein, Editor
Comparisons of early computer art and today's newer art. What is art? What is art in computer art?

Inexpensive Graphics from a Storage Cathode Ray Tube
by Charles J. Fritchie and Robert H. Morriss, Tulane University, New Orleans, Louisiana
Illustrations and photographic techniques used to achieve graphics from a storage tube CRT.

An Investigation of Criteria for Evaluating Computer Art
by Thomas E. Linehan, Ohio State University, Columbus, Ohio
The new aesthetic of computer art requires a departure from the previous, formalist-traditionalist doctrines for evaluating art.

List of Coverage for Up-Coming Issues

Applied Arts and Graphics
Architectural Graphics
Cartography Systems
Computer-Aided Design
Computer Assisted and Managed Instruction Utilizing Computer Graphics
Computer Graphics in Physics, Chemistry, Mathematics, etc.
Computer Programs for New Applications
Display Systems and Graphics
Fine Art and Media Explorations
Graphics in Business
Hardware Systems and Graphics
Interactive Graphics Languages and Systems
Languages for Computer Graphics and Graphics Primitives
Software Systems and Graphic Requirements
Statistical Packages and General Graphing
Syllabi for Computer Graphic Courses



- -(may be copied on any piece of paper)- -

To: *COMPUTER GRAPHICS and ART*
Berkeley Enterprises, Inc. – Business Office
815 Washington St.
Newtonville, Mass. 02160

( ) Please enter my PERSONAL (U.S. and Canada) subscription to CG&A ($10 per year).
( ) Please enter my FOREIGN PERSONAL subscription to CG&A ($13 per year).
( ) Please enter my LIBRARY/DEPARTMENTAL subscription to CG&A ($15 per year).
( ) Enclosed is my PERSONAL CHECK AND/OR PURCHASE ORDER for CG&A.
( ) Enclosed is $2.50 for a sample copy of CG&A (applicable toward a subscription).

FULL REFUND IN 30 DAYS IF NOT SATISFACTORY

Name______________________________________
Title______________________________________
Organization________________________________
Address____________________________________
____

HERE IS YOUR OPPORTUNITY FOR FEEDBACK TO US:

( ) I hope to submit for publication in CG&A material on the following topics:______________________________
( ) I am interested in reading materials by the following authors:______________________________
( ) I am particularly interested in coverage of the following subjects:______________________________
( ) I would like to receive materials on other Berkeley Enterprises, Inc. publications: ( ) *COMPUTERS and PEOPLE* ( ) *The COMPUTER DIRECTORY and BUYER'S GUIDE* ( ) *People and the PURSUIT of Truth* ( ) *The Notebook on COMMON SENSE and WISDOM* ( ) *WHO'S WHO in COMPUTERS and DATA PROCESSING* ( ) Books
( ) I am interested in: ( ) black and white computer art reprints at low cost (a bonus for subscribing to CG&A) ( ) 77 page FORTRAN IV art manual ( ) 45 page interdisciplinary graphics bibliography by G. Hertlein
( ) Additional Comments (attach another paper if needed):______________________________