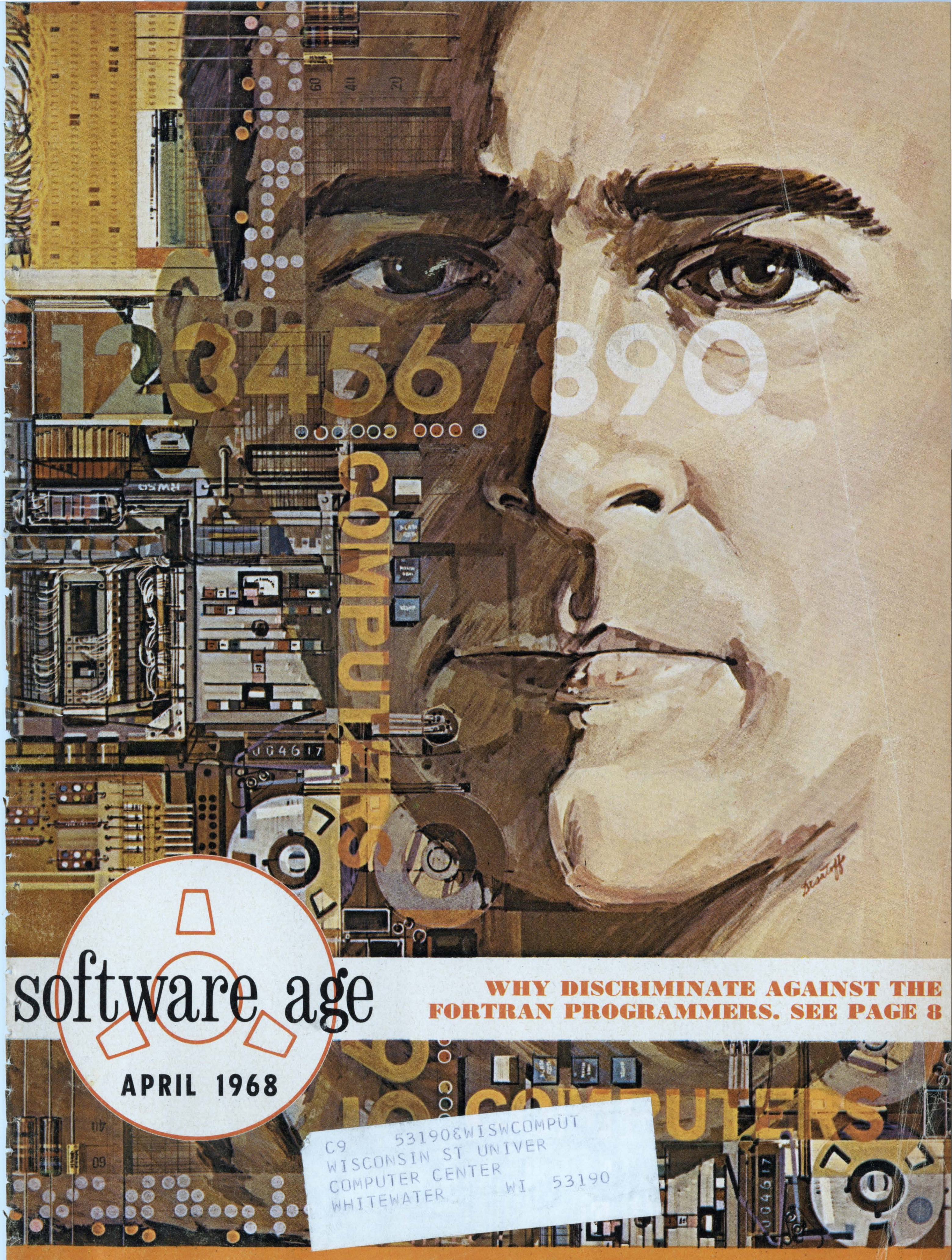


software age

APRIL 1968

WHY DISCRIMINATE AGAINST THE FORTRAN PROGRAMMERS. SEE PAGE 8

THE MAGAZINE FOR PROGRAMMERS • MATHEMATICIANS • ANALYSTS • EDP MANAGERS

software age

APRIL, 1968 Vol. 2–No. 3

CONTENTS **PAGE**

Publisher – H. L. Rothra
Associate Publisher – David W. French
Advertising Manager – Norman Jacobs
Advertising Assistant – Betsy Pavkovich
Production Manager – William J. Murphy
Director of Circulation – Howard Rogers
Circulation Manager – Judith Arnopolin

Circulation of this issue more than **91,000**

Software Age is published monthly by
Press–Tech, Inc.
(312) 869–1244

1020 Church Street, Evanston, Illinois 60201. Subscrip-free to qualified readers. Others, $10/yr. Individual copies, $1. Foreign subscriptions, $10/yr.

Main Sales Office: Norman Jacobs (Adv. Mgr.) or Norman Brodsky, 1020 Church Street, *Evanston, Illinois* 60201—Telephone (312) 869-1244. Gerald Green, 60 E. 42nd, *New York, N. Y.*—Telephone (212) 697-5356. Richard Faust, 9800 S. Sepulveda Blvd., *Los Angeles, California* 90045—Telephone (213) 776-0100. David S. Wood, P. O. Box 189, *Canton, Massachusetts* 02021—Telephone (617) 542-1466. Richard D. Clemmer, 4900 Red Fox Drive, Annandale, Virginia 22003 *(Washington, D. C.)*—Telephone (202) 461-9792. Richard Faust, *Palo Alto, California*—Telephone (415) 327-8340.

CONTROLLED CIRCULATION POSTAGE PAID
AT MADISON, WISCONSIN

Without moon men and planetary probes how on earth does United Air Lines attract top men to Elk Grove, Ill.?

The answer is simple.

United Air Lines has the largest computer system in the business world in Elk Grove Township, Ill. We have the kind of hardware that challenges even the best systems analysts and programmers.

Imagine working with hardware like this: three UNIVAC® 1108-II processors sharing a common 262K, 36-bit word core; 45 drums providing random access storage of nearly two billion characters of data; over 2800 Uniscope™ visual communication terminals that link 116 United cities to our central computer complex; 700 automatic printer units for passenger tickets and airbills.

This on-line computerized system will provide information in 17 basic categories ranging from passenger reservations, meal planning, crew and aircraft scheduling to flight planning and cargo loading.

This kind of information, and how it gets to our management, is critical to the operation of the world's largest airline. And because we want to stay that way, you can be sure our management will encourage you in the exploration of new areas of computer science.

Since this new system becomes operational this year, you can get in on the ground floor. If you can handle responsibility for projects from analysis, through coding, debugging, and documentation and if you've had two to five years experience in business or scientific applications, let us hear from you.

United also offers a few fringe benefits that might attract you to Elk Grove. Free air travel for you and your family, company-paid insurance, an excellent retirement program and an employee stock purchase plan are just a few.

Please send your résumé and salary requirements to Mr. H. K. Schlinker, Management Employment, United Air Lines Executive Offices, P.O. Box 66100, Chicago, Illinois 60666.

UNITED AIR LINES
An Equal Opportunity Employer

new installations

■ ***Computer Learning Corporation*** will use 25 Model 110 computers and 4 Keytape data preparation devices of Honeywell Inc. to help young adults pursue data processing careers. The firm offers training in keypunching, programming, systems design, engineering and management science.

Four computers are planned for the Washington area and others are slated for installation at offices in Boston and Atlanta later this year.

■ ■ ■

Rutgers University ordered two SDS, Scientific Data Systems, computers, a Sigma 2 and a Sigma 5, for studies in nuclear structure physics. Rutgers will use the $500,000 dual computer system to study the composition of atomic nuclei and the forces which hold these nuclei together.

In this application the computers will be used in conjunction with a tandem van de Graaff accelerator, which accelerates nuclei of light atoms to high energy where they strike the nuclei of the substance under study. The resultant radiations and their energies are measured and analyzed by the computer system. The Sigma 2 will gather and sort this information and feed it to the Sigma 5 for analysis.

■ ■ ■

The Boeing Company, Commercial Airplane Div., has awarded a $1.8 million contract for the lease of EAI, Electronic Associates, Inc., digital computing equipment.

Two EAI 8400 Mod II Digital Computing Systems, with Data Conversion and other peripheral equipment, will be utilized by Boeing in design and development segments of the Supersonic Transport program. The EAI 8400 is a 32-bit, general-purpose digital computing system which features one microsecond memory base register for dynamic program relocation and memory protection and a basic memory capacity of 8,192 words—expandable to 65,536 words.

■ ■ ■

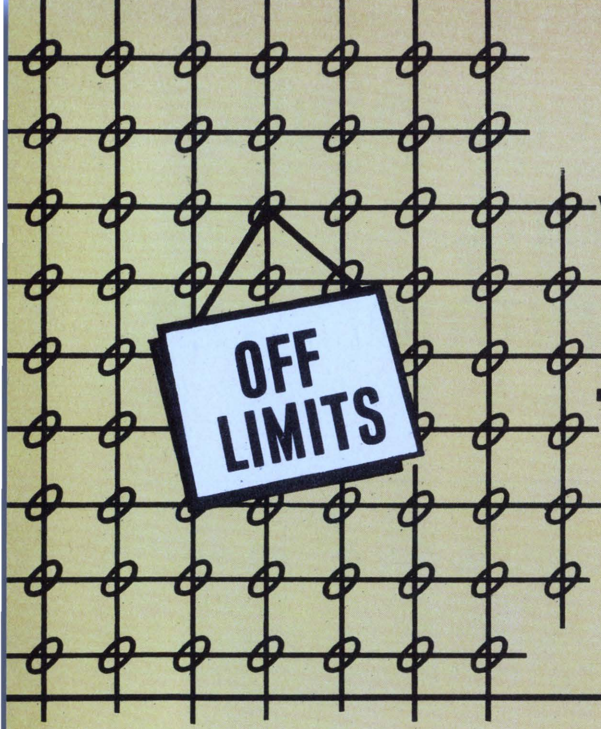
Central Casting Corporation, the main sources of "extras" in the motion picture and television industry, has ordered a GE-405 computer system from the General Electric Co.

The GE-405, smallest number of GE's 400 series of medium-scale information systems, will be used to calculate both gross and net daily earnings of approximately 3,000 extra players. It also will be used in processing the requirements of the Motion Picture Industry Pension Plan, the industry's Health and Welfare Fund and Screen Actors–Producers Pension and Health and Welfare Plan, all of which cover approximately 30,000 employees in the motion picture and television industry.

■ ■ ■

Information Systems Design, Inc., a pioneer West Coast computer service firm, has ordered a $2 million Sperry Rand Univac 1108 Computer System in order to keep pace with the increasing demand for computing service in the San Francisco Bay area.

Remote data communication terminals, located in the offices of customers, will be connected by telephone lines to the UNIVAC 1108 so that various types of engineering and scientific problems, as well as business applications can be received and processed with maximum convenience. ■



WHY DISCRIMINATE AGAINST THE FORTRAN PROGRAMMERS?

Robert A. Sebastian
Timothy J. Horrigan
CAYWOOD-SCHILLER, ASSOCIATES
Chicago, Illinois

■ You thought discrimination did not carry over into the computer world? Not so! For any group there is always another to foster its welfare and to keep it in its place—for its own good, of course. Take the FORTRAN programmers—graduate engineers, physicists, mathematicians—analyzing signals from space, hyper velocity impact, or nonlinear circuits. Why are they forbidden by the FORTRAN specification writers to overlap tape input/output; to alternate I/O areas; to manipulate files; to handle data checks; to end files with tape marks? Why? They are a stereotype—*the specification writers know.*

Their COBOL or assembly language counterparts are not forbidden these operations: They know computers; they have important input/out problems. The FORTRAN programmer creates 'scientific' programs in an ivory tower. There is no data in an ivory tower—*the specification writers know.* Besides, "The FORTRAN programmer would not know what to do with input/output capability if he had it," they say. "He does not know a tape mark from an IBG."

It may come as a shock to some people, but a large and growing number of FORTRAN programmers are not especially pleased with their I/O limitations. They look at their optimization programs; their matrix inversions; their simulations; their statistical analysis problems; their partial differential equations. They *know* they have input/output problems. They figure, that since they can master higher mathematics, they can also master tape marks.

To put it mildly, FORTRAN has a few inadequacies when it comes to magnetic tape input/output. In other areas FORTRAN programmers have no complaints. They like the notational ease of FORTRAN; its optimized subscripting; its lack of verbosity. They have a great investment in knowing the language inside out. It is familiar and effective. Now, in the troublesome I/O area, a new development, COMPAT-F, makes them the equal of their COBOL and assembly language counterparts: It provides the status and power of genuine input/output capability.

Inadequacies in FORTRAN Tape I/O Eliminated by COMPAT-F

Caywood-Schiller, Associates, specializing in 'hard core' highly mathematical operations research, was constantly running into these inadequacies. Problems with TOS/DOS FORTRAN for the IBM System/360 were particularly severe in that tape records are blocked 64 words to a block (63 actually, the first word is a FORTRAN control word). Thus, communication with the System/360 programming support packages, including sort/merge, was essentially impossible. Processing COBOL tapes with decimal fields was also impossible, even in OS FORTRAN. CSA therefore decided to write its own magnetic tape I/O System for the 360. This system, called COMPAT-F, consists of several multiple entry FORTRAN callable assembly programs that interface directly with the channel scheduler of the system control. It is designed to facilitate the full utilization of System/360 hardware features normally accessible only in assembly language, while retaining the ease of expression and notational familiarity of FORTRAN. A summary of significant COMPAT-F features is presented in Figure 1.

Read Any Tape

Any 7 or 9 channel magnetic tape (in the IBM Series 2400 Magnetic Tape Units) can be read by COMPAT-F, either forward or backward. The byte strings are transferred from tape to core by a "read" command—then a "scatter" command fills the proper FORTRAN words. Provisions are made to easily handle packed decimal, zoned decimal, or binary tape format. Blocked records are no problem, either. In fact, this is another desirable feature of COMPAT-F. When writing a tape the programmer may choose any blocking factor he wishes. He is not restricted to 252 bytes per block as with the normal TOS/DOS FORTRAN WRITE statement. Thus, better tape utilization may be realized.

When a new 360 is installed the business or COBOL-type programs are generally converted first. "Sure," the Data Processing Manager tells the FORTRAN programmer, "we can give you data for your forecasting model. We have weekly sales of all products for the last 5 years." What he failed to point out was that FORTRAN cannot read this data since it wasn't written in FORTRAN. This leaves the programmer with three alternatives, none of which is too appealing:

1) Learn COBOL and write a special purpose subroutine for this job.

2) Ask a COBOL programmer to write the special purpose subroutine.

3) Forget the whole thing.

It is not uncommon for one of these special purpose subroutines to consist of 60 to 70 statements in the Procedure Division alone. Contrast this to the three basic COMPAT-F statements that would do the job—a "read," a "check," and a "scatter." The "read" command transfers one physical record from tape to core; the "check" tests for end of file (other special conditions such as read error, write error, load point, tape unloaded, etc. may also be checked); and the "scatter" sets up the proper FORTRAN words with decimal-binary conversion, when required.

Write Compatible Tapes

The IBM utility packages such as sort/merge can be of great assistance to the programmer. But, in general, they won't read FORTRAN written tapes. Sort/merge, for example, cannot be used with FORTRAN written data if there are more than 63 words in a record.

A COMPAT-F written tape can be read by these packages as well as COBOL, assembly language, and, of course, COMPAT-F.

THE FUNDAMENTAL FEATURES OF COMPAT-F FOR IBM SYSTEM 360 FORTRAN

★ Allows FORTRAN utilization of data prepared by other program languages or by other computers.

★ Outputs FORTRAN tapes acceptable to utility programs (i.e., sort/merge), other program languages, or other computers.

★ Provides substantial computer time saving by allowing the overlapping of input/output with processing.

★ Permits increased tape utilization through usage of blocked records.

★ Enables FORTRAN to accept multi-file reels and/or multi-reel files.

★ Furnishes the capability of complete tape manipulation, including forward and back spacing of blocks and files.

★ Gives the programmer the option of taking specific action as a result of input/output errors.

In short, COMPAT-F combines the FORTRAN algorithmic capability for scientific computation with the similar capability of assembly language for data handling.

FIGURE 1

Overlap I/O and Processing

The concept of simultaneously reading, writing, and processing is generally unknown to a FORTRAN programmer—it can't be done. When FORTRAN reads, it reads, and no other operation may be occurring at the same time. However, the 360 hardware is not so limited and COMPAT-F gives the programmer the ability to exploit this capability. In other words tape input and/or output operations may be overlapped with processing. Even a FORTRAN programmer can compute that as much as two-thirds of the running time may be saved by this feature.

Program Responses to I/O Errors

What is more irritating to a programmer than to have a job abort while reading the 99,999th record? The program has already processed 99,998 records and with two more to go an unreadable portion of tape is encountered. The programmer is not consulted—he has no choice—his job is aborted by FORTRAN. With COMPAT-F, on the other hand, the programmer is in charge—he may attempt to read the record backwards (sometimes with more luck than reading it forward)—he may skip the record and go on to the next one—or he may elect to abort the program. The decision is, however, his.

Use with DOS, TOS or OS

COMPAT-F may be used with either DOS, TOS or OS FORTRAN. However, the advantages are greater with DOS or TOS. The I/O inadequacies of FORTRAN outlined in the previous sections mainly refer to DOS and TOS. Some, but not all, of these are eliminated by OS FORTRAN. For example, OS FORTRAN does not allow the overlapping or tape manipulation of COMPAT-F. Furthermore, the present requirement of a 256 K core will undoubtedly deter some installations from operating under OS.

Command Structure

Figure 2 illustrates the use of the two main commands of COMPAT-F—"Read Forward" and "Scatter."

Another feature of COMPAT-F tells how many bytes were actually read with each "read" command. If this number is designated by NBYTE, then NREC = NBYTE/30 is the number of records read in this example. Thus, if the last block contained less than 50 records, the "scatter" command would handle the situation if the repetition factor (—50) were changed to —NREC.

In all, COMPAT-F has 18 com-

EXAMPLE ILLUSTRATING THE TWO MAIN COMMANDS—READ AND SCATTER

A magnetic tape reel exists containing sales records for a large number of stock keeping units (SKU). There are 50 records in a tape block—each record contains 30 bytes with the following information:

Byte No.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
SKU			SALES				DESCRIPTION																	DATE				DEPT.	

The Description field is alphabetic; all other data is in packed decimal. Computations with this data require only the Sales of each record.

The "read" command is

CALL COMREF(TAPE6, INDATA(J), 1500)

This command reads 1,500 bytes, or one block (whichever is smaller) from Tape 6 into the array INDATA.

The "scatter" command is

CALL COMCAT(INDATA(J), – 50, – 30, 4, 7, 0 ISALES)

Bytes 4 to 7 of the first record of INDATA(J) are translated to binary and moved to ISALES(1). The zero in the argument list indicates that the read-in data was in packed decimal format. Then, bytes 4 to 7 of the second record of INDATA(J) are translated to binary and moved to ISALES(2). This process is repeated 50 times as designated by the optional repetition factor (– 50) in the argument list. The optional argument of – 30 indicates the number of bytes in each record.

FIGURE 2

mands which can be divided into 5 groups as follows:

Group I—Control Commands— The commands of this group are necessary for the operation of COMPAT-F. They supply the control framework required by the flexible symbolic identification of magnetic tape units and the overlap of input, output and processing.

1. **COMpat ASsigN** — CALL COMASN (UNIT1,N1,UNIT2, . . . , UNITK,NK)—The simple or explicitly subscripted variables UNIT1, UNIT2, . . . , UNITK are established as COMPAT-F symbolic tape unit identifiers by association with the corresponding SYS numbers specified by N1, N2, . . . , which may be literals, simple variable or explicitly subscripted variables of integer type.
2. **COMpat CHecK**—CALL COMCHK (UNIT) CALL COMCHK (UNIT,L)—The magnetic tape unit specified by the COMPAT-F symbolic unit identifier UNIT is checked for normal completion of its last operation, and waits if necessary. After a read (or read backwards), if the optional second argument L is present, the actual number of bytes read is stored in L. This command must be followed by a simple GO TO. When the last operation has an abnormal completion (e.g., end of file), COMCHK returns to the GO TO: otherwise it skips over the GO TO on its return.

Group II—Elementary Commands—The commands of this group are the COMPAT-F high-speed version of the tape commands normally available in DOS or TOS FORTRAN.

3. **COMpat WRiTe**—CALL COMWRT(UNIT,ARY,L) — The physical tape unit symbolically identified by UNIT is selected and the array ARY of L bytes in length is written on it as a single physical record.
4. **COMpat REad Forward**— CALL COMREF(UNIT, ARY,L)—The physical tape unit symbolically identified by UNIT is selected and L bytes of data or one block (whichever is smaller) is read into the array ARY.
5. **COMpat Back Space Block**— CALL COMBSB (UNIT)—The physical tape unit symbolically identified by UNIT is selected and back spaced one block.
6. **COMpat REWind** — CALL COMREW(UNIT) — The physical tape unit symbolically identified by UNIT is rewound.
7. **COMpat Write Tape Mark**— CALL COMWTM(UNIT)— The physical tape unit symbolically identified by UNIT is selected and an end of file symbol (tape mark) is written on it.

Group III—Expanded Commands—This group provides additional tape manipulation commands normally available only in assembly language.

8. **COMpat Forward Space Block** —CALL COMFSB(UNIT)— The physical tape unit symbolically identified by UNIT is selected and forward spaced one block.
9. **COMpat Forward Space File** —CALL COMFSF(UNIT)— The physical tape unit symbolically identified by UNIT is selected and forward spaced one file (to next tape mark).
10. **COMpat Back Space File**— CALL COMBSF(UNIT)— The physical tape unit symbolically identified by UNIT is selected and back spaced one file (to last tape mark).
11. **COMpat REad Backwards**— CALL COMREB(UNIT, ARY,L)—The physical tape unit symbolically identified by UNIT is selected and L bytes or one block of data (whichever is smaller) is read back-

wards into the array ARY. Execution of COMCHK on UNIT causes the automatic left justification of this data with ARY.

12. **COMpat Rewind and UNload**—CALL COMRUN(UNIT)—The physical tape unit symbolically identified by UNIT is selected, rewound and unloaded.
13. **COMpat ERase Gap**—CALL COMERG(UNIT) — The physical tape unit symbolically identified by UNIT is selected and a gap is erased.

Group IV—Analysis Commands—The commands of this group provide the (optional) capability of analyzing abnormalities occurring during input/output operations—including errors—and programming specific reactions.

14. **COMpat ANalysis Mode**—CALL COMANM—The COMPAT-F routines are modified (set to analysis mode) so that numerous abnormal conditions including input/output errors are returned to FORTRAN by the Channel Scheduler.
15. **COMpat CHecK**—CALL COMCHK(UNIT) CALL COMCHK(UNIT,L)—Checks, waits, left justifications, and abnormal returns are performed as in the non-analysis mode; however a data input/output error now causes an abnormal return.
16. **COMpat TeST**—CALL COMTST(UNIT,KODE) CALL COMTST(UNIT,KODE, L)—The physical unit symbolically identified by UNIT is selected and its status is determined; a status value is assigned the integer variable KODE accordingly. These values range from available through various types of busy to tape end conditions and input/output errors. If the optional third argument is present and the last operation is a read or read backwards the acual number of bytes read is stored in L. No left justification is performed on data read backwards.

Group V—Data Manipulation Commands—The commands of this group allow the gathering or scattering of data into or out of the array format required by the COMPAT-F commands. In the case of packed or zoned decimal input the scattering is accompanied by the optional automatic conversion to binary and vice versa in gathering for output.

17. **COMpat sCATter**—CALL COMCAT(ARY,NF1,NL1, LOC1,NAME1,NF2,NL2, LOC2,NAME2, . . . , NFK, NLK,LOCK,NAMEK) — With ARY as the input array the general parameter sequence—NF,NL,LOC,NAME—which is repeated as required, has the following significance: The field (bytes) of ARY between NF (the First byte of this field) and NL (the Last byte of this field) inclusively is moved into the field starting at byte position LOC in NAME. When LOC is zero (or minus one) the specified field of ARY is taken as packed decimal (or zoned decimal) and is moved into the word at NAME as a FORTRAN integer in binary format. The general parameter sequence may be prefixed by an optional repetition factor (preceded by a minus sign) to effect a chained scatter in the following manner: —NR,NF, NL,LOC,NAME. In this case NAME must be an array name.

18. **COMpat GAtheR**—CALL COMGAR (ARY,NF1,NL1, LOC1,NAME1,NF2,NL2, LOC2,NAME2,. . .,NFK,NLK, LOCK,NAMEK)—With ARY as the output array the general parameter sequence —NF,NL, LOC,NAME—has the same significance as in COMCAT except that the movement commences with the LOC byte in NAME and consecutively fills the field at ARY having NF as its First byte and NL as its LAST byte, inclusively. When LOC is zero (or minus one) the word NAME is taken as a FORTRAN binary integer and is converted to packed decimal (or zoned decimal) during the move to the specified field of ARY. The repetition factor applies as before.

Programmer in Command

As with any language, the clever programmer can use COMPAT-F to better advantage than the not-so-clever programmer. In COMPAT-F the programmer is "more in command" than he is in FORTRAN I/O. There are, therefore, certain things that he must know. For example,

1) Does the tape have a tape mark at the beginning? If so, he must have a "forward space file" statement before he attempts to read the tape.

2) Does the tape have a label? If so, he may either check the label or "forward space block" and skip it.

3) Are the numeric fields written in packed decimal, zoned decimal, or binary? If packed decimal or zoned decimal are used in the numeric fields, then the scatter with binary conversion is mandatory.

In general, the programmer must know the tape fairly well. With FORTRAN I/O, for example, he need not know the answer to any of the above three questions—FORTRAN "takes care of him." The problem is that FORTRAN overprotects him and, in so doing, takes away some of the powerful features of the 360 hardware. COMPAT-F gives him back these features. ■

Encoding from Alphanumeric Names to Record Addresses

ORIGINAL INDEX ITEM	TRIPLET
TOIL	$4L_\mu$
TOILE	$5E_\mu$
TOILER	$6R_\mu$
TOILET	$6T_\mu$
TOILETRY	$7R_\mu$
TOILETTE	$7T_\mu$
TOILFUL	$5F_\mu$
TOILSOME	$5S_\mu$
TOILWORN	$5W_\mu$
TOIT	$4T_\mu$
TOKAY	$3K_\mu$
TOKEN	$4E_\mu$
TOKOLOGY	$4O_\mu$
TOLA	$3L_\mu$
TOLAN	$6b\!\!/0\!\!/3L_\mu$
TOLBOOTH	$4B_\mu$
TOLD	$4D_\mu$
TOLE	$5b\!\!/0\!\!/4E_\mu$
TOLEDO	$5D_\mu$
TOLERABLE	$5R_\mu$
TOLERANCE	$7N_\mu$

DIRECTION OF SEARCH

Dr. Thomas C. Lowe

Informatics Inc.
4720 Montgomery Lane
Bethesda, Maryland

This work was performed at the Information Systems Laboratory, The Moore School of Electrical Engineering, University of Pennsylvania. It was supported by the Army Research Office (Durham) under contract number DA-31-124-ARO-D-352 and by the Air Force Office of Scientific Research under contract number AF-49(638)-1421.

■ In many applications it is necessary to obtain an absolute memory address for a record in external storage (such as disk) when given an alphanumeric name. Some approaches to this problem will be presented below.

This problem in general is called the encoding problem; the given alphanumeric name is called a *term*. It is not the purpose of this article to present any general solution, but rather what follows is a brief review of some well-known techniques, and an introduction of a new one.

Many workers have investigated problems in the area of address encoding. The basic problem is easy to understand: for example, assume that a term consists of up to 28

FEATURES OF DECODING MECHANISMS

	Symbol Tree	Key-to-Address Transformation	Balanced Tree	Triplet Search
Processes Variable Length Term	Yes	Yes	No	Yes
False Hits Possible When Term in File	No	Yes	No	No
False Hits Possible When Term Not in File	No	Yes	Yes	Yes
Extensive Character Processing Required	Yes	No	No	Yes
Easily Partitioned for Few External Memory Accesses	No	—	Yes	Yes
Easily Updated	Yes	Yes	No	No
Large Fast Memory Requirement	Yes	No	No	No

characters from an alphabet of 37 symbols. There are then about 10^{44} possible terms.

In a typical situation at most 10^4 or 10^5 terms will actually be used, so that the encoding problem is one of mapping the index items used into appropriate external memory addresses.

Several schemes for encoding will be considered in brief. Not all of these methods were originally proposed for this purpose, but they are applicable to it.

It is necessary that any system design be particularly oriented toward the characteristics of the equipment available. Not only do such factors as word size, fixed word length operation as compared to variable word length (or character oriented), and character handling features in fixed word length machines enter the picture, but as Bowlden[1] has pointed out, some older machines with table look-up features such as the IBM650 and the Control Data RPC-4000 have instructions that may make vast differences in the methods used. Associative memory devices are not considered here.

Symbol Trees

In symbol trees each node represents a symbol (that is, a single character) of a term and branches lead to following symbol nodes. If two or more terms have the same initial symbol sequence, they share an initial path down the tree until the first symbol differentiating the two symbol sequences is encountered. The average branching factor at the nodes and the tree depth may be controlled by considering an index item as a string of bits, and generating a new symbol sequence by partitioning the bits in a way different from the original character representation scheme. The size of the alphabet used in the tree representation can thus be varied. Fredkin has proposed the use of two bits per node in his "binary Trie"[4]. In the decoding application being presently considered, the external memory address could be placed at the last node of every path through the tree corresponding to an index item.

The efficient mechanization of such a tree structure has been given considerable attention. Susenguth[11]

(Continued on page 28)

GAME THEORETIC APPLICATIONS

James P. Dix
Programmer/Analyst
Control Data Corporation
Systems Programming Section
Analytical Services Department
Los Angeles Data Center

PART II of a two part discussion of gaming which draws heavily on the defense problem for its examples. Geometrical, Monte Carlo and other methods are applied to the solution of the model problem which was developed in Part I.

SIMPLE COMPUTER MODEL—GEOMETRICAL APPROACH

This model's objective is to present a descriptive picture of the "goodness" of a siting. It is not intended to be a mathematical solution. It must be, however, an equation, for example:

(3)

$$g_s = \sum_i w_i m_i \qquad \text{where:}$$

g_s = goodness of the site
m_i = i^{th} merit factor
ω_i = weight of i^{th} factor in the overall picture

After some consideration, two merit factors can be arrived at.

m : an indicator of the number of people that are not defended by a siting
v : the volume of space that measures the imbalance of coverage of the p_w surface.

In this model, we cannot perform any simulation or any optimization analyses since we will be looking only at the capabilities of the defense under a specific attack given the required number of ABM's in consideration of the balanced defense approach. That is, we will assume that the unanswered questions fall out as corollaries to the theorem that if we answer the question: How well are we defending the p_w surface?, then we have reached the game-theoretic best. The goodness here is defined as a geometrical good.

Let's look at the city once more and see what restrictions are imposed by this intercept surface (Figure 8).

The d_a or decision altitude which has been set by the firing doctrine is a measure of the discrimination and target tracking capabilities. It says that launching the ABM before the ICBM reaches this altitude would handicap the defense by wasting missiles.

The l_a or launch altitude which is computed as a function of the intercept point and the time to intercept is used to compute the surface of last penetration. Once the object has penetrated this surface, no site could launch any missiles to intercept the object and prevent its penetration of the p_w surface.

The next question is a complicated expression which yields this launch altitude. I mention it only for the readers who are interested.

(4) $$l_a = h(t_o + t(i_a)) \qquad \text{where:}$$

t_o = time to intercept
i_a = intercept altitude
$t(i_a)$ = time function which gives the time to impact as a function of altitude
$h(t_o + t(i_a))$ = altitude function (inverse of the time function) which gives altitude of object as a function of time to impact

Note from inspection of Figure 8 that there is an opening or a weak spot to the defense. Note further that it is near the heart of the city. But also notice that the opening is quickly closed if (1) we could move the intercept altitude down or (2) move the decision altitude up. The question that comes to mind then is how can we measure this weakness.

One way to measure the imbalance of coverage of p_w is by calculating the signed volume of space between the d_a and the l_a as a weighted function of d_a, and since d_a is a horizontal plane (for local defense), we have the formula:

(5)

$$\bar{v} = \frac{\int_{d_{a_1}}^{d_{a_2}} wv\,d(d_a)}{\int_{d_{a_1}}^{d_{a_2}} w\,d(d_a)}$$

v = the volume of space between l_a and d_a, where $v = v_o + A^*(d_a - d_{a_0})$ where v_o, A, and d_{a_0} are constants
w = the weight associated with d_a
d_{a_2} and d_{a_1} = the limits chosen for d_a

Another way to measure this vulnerability is to ask how many people are affected by the hole in the defense. The answer to this question is unfortunately not easy to obtain by this approach. For example, by intercepting lower, you could obtain such a measure by subtracting X number of people from the balanced defense plan. However, by doing so, you would disturb the entire game-theoretic balance of intercepting on the p_w surface. About the best you can say is that with this d_a, this siting plan fails.

With others, it might work. The formula we are looking for is then:

(6)

$$\overline{m} = \frac{\int_{d_{a_o}}^{d_{a_1}} w\, m\, d(d_a)}{\int_{d_{a_o}}^{d_{a_1}} w\, d(d_a)} \qquad \text{where:}$$

$m = 0$ or 1 depending on d_a;
if $d_a < d_{a_o}$, then $m = 0$
if $d_a \geq d_{a_o}$, then $m = 1$
w, d_{a_1}, and d_{a_2} are as before

The next step is to be able to compute these merit factors considering all types of threats. The three variables considered will be reentry angle, a_ϵ, reentry azmuth, θ, and weapon yield, y. The formula for $\overline{v}$ becomes then:

(7)

$$\overline{v} = \frac{\iiiint w\, v\, d\, d_a\, d\, a_\epsilon\; d\,\theta\, d\, y}{\iiiint w\, d\, d_a\, d\, a_\epsilon\; d\,\theta\, d\, y} \qquad \text{where:}$$

w = considered as a weight function of each variable

The equation 7 is merely the mathematical way of describing the sentence "By 'combining' all the variables together, we can look at the final merit factor and come up with our decision." This 'combining' must be rigorously followed because (1) every statement must be verified by a certain amount of logic or mathematics and (2) no computer program could possibly understand the laymen's terms.

A similar equation can be formulated for $\overline{m}$.[1]

Now to the computer program. The Geometrical Approach is a very simple approach insofar as the computer goes.

In finding the volume v for the purpose of solving equation 7, in principle, is straight forward. Volume is equal to Area times the Height.

(8)

$V = A\,(\overline{1}_a - d_a)$ where:
A = area of the plane cut out by the defended area
$\overline{1}_a$ = average altitude of the 1_a over some projection of this defended Area

In a computer program written in FORTRAN what is done is as follows:

Setup a two dimensional array HEQU which describes the p_w surface. HEQU(I,J) represents the height above the surface at which p_w people would be killed if a weapon of yield y were burst there. The point (I,J) represents a grid point, or a grid square with area A_g. If no such altitude exists for some (I,J), set HEQU(I,J) = 0.

The defended area A of equation 8 falls out immediately and has the equation

(9) $A = \left(\sum_{\substack{i,j \\ HEQU_{i,j} \neq 0}} 1\right) * A_g$ which is easily computed.

The next step is to calculate for each grid point (I,J) the launch altitude for each site. The minimum is then saved. As given before the launch altitude is determined by knowing t_o, the time to intercept and HEQU(I,J), the intercept altitude. The functions t and h of equation 4 must be fairly simple to compute or a tremendous amount of machine time can be eaten up. Hence the best procedure is to do all trajectory calculations (offense and defense) long before the volume calculations are done and to save the results in the form of a set of coefficients which are, perhaps, produced by a least square method.

This procedure is recommended for simulations which are called event oriented as opposed to clocktime oriented. The reduction in speed is usually enormous. The accuracy, however, can be somewhat limited. I will spend more time on this subject when discussing the Monte-Carlo approach.

We can now see that:

(10)

$$\overline{1}_a = \sum_{\substack{i,j \\ HEQU_{i,j} \neq 0}} 1_a \,(\text{minimum's}) / n \qquad \text{where:}$$

$$n = \sum_{\substack{i,j \\ HEQU_{i,j} \neq 0}} 1 \qquad \text{as in the Area equation 9.}$$

Having done this for every reentry angle, reentry azimuth and weapon yield, we can obtain the answer as per question 7. The number of calculations which are needed of those which are time consuming (those involving missile trajectories) amounts to:

(11)

$n_c = n_g * n_s * n_{a_\epsilon} * n_\theta * n_y * n_d$ where:
n_c = number of sets of calculations
n_g = number of grid points such that HEQU (I, J) $\neq 0$
n_s = number of sites
n_{a_ϵ} = number of reentry angles
n_θ = number of reentry azimuths
n_y = number of weapon yields
n_d = number of deployments

To achieve any kind of accuracy the number of probable sets of calculations should be greater than n_c.

By using equation 11, the number of sets of calculations for a small hypothetical city would be $n_c = 1000 * 3 * 3 * 3 * 3 * 3 = 243{,}000$ sets. To insure that we can safely solve this problem in, say, one hour of computer time, we must be able to perform each set of calculations in roughly .015 seconds. Under the above scheme this is moderately easy to do on the Control Programming time (assuming background model is already developed) is very minimal; perhaps 120 manhours.

In closing, this approach can be very useful in establishing a play that will work in most cases and will not cost a great deal to use.

THE MONTE-CARLO APPROACH

The purpose of this approach is to reduce a complicated strategy to a simple and understandable curve which can produce for you a merit figure. It makes use of the confidence level that is so often annoying when solving real problems. It answers satisfactorily the question: How many people will die if I push this button? Compicated chains of different logic can be linked together by merely playing a game in which you simulate conditions by throwing switches on or off depending on random numbers.

The merit factor described here is P_a, the price of admission, or the number of ICBM's necessary to penetrate.

In the Monte-Carlo Approach, we are free to bring into play all manner of hazards, both offense and defense, which under the Geometrical Approach would have been impossible. Some of these are ABM stockpile depletion, ABM aborts, miss distance at intercept, warheads–decoys problem, warhead dispersal, and intercept geometry.

You need a firing doctrine or a set of rules by which you should play under every set of circumstances. Not all plays are made but the logic must be there. Here again, though, we maintain our theory that we must site to defend the p_w surface and hence that our intercepts must be on that surface.

The Monte-Carlo method must establish a simulation procedure because of the importance of the interdependence of events over a period of time and the need to sample this interdependence at any moment in its history.

There are two types of simulations: (1) event oriented and (2) clock–time oriented. Both types accomplish the same purpose but in a slightly different manner.

The event oriented simulation is the easier of the two since you establish a chain of happenings in advance and determine the success or failure of the simulation by merely looking at the resultant happening.

In a time oriented simulation we must, at each clock pulse, check for success or failure. When timing is most important (e.g. the ABM and the warhead must be at the same place at the same time and the ABM burst must be so timed), it is desirable to use this type of simulation. You will then be able to observe a truer test of your system. However, it turns out that the time oriented simulation is good only for "feasibility studies" or "demonstration of effectiveness" and not for analysis work. It is too time consuming (on the computer) and therefore must be restricted to typical cases.

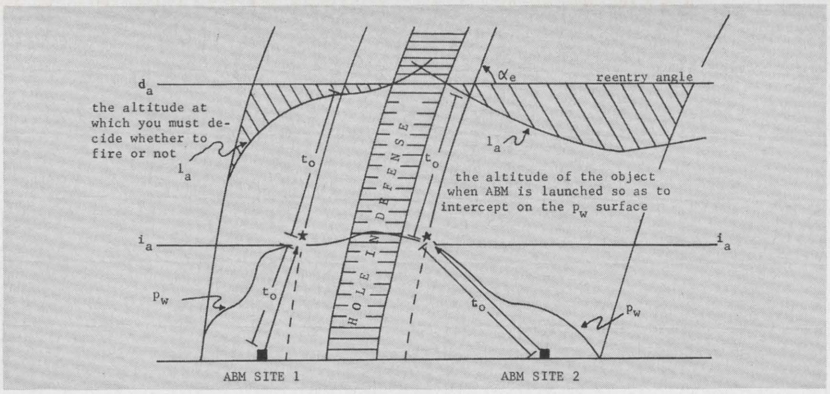


Figure 8

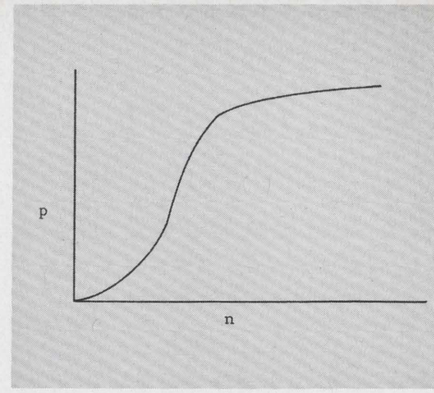


Figure 9

Ours will be event oriented and so we proceed to describe the events. In this approach, we can put in as many details as we need. However, for simplicity's sake, I only choose five events with meager descriptions.

(1) Detection and track
(2) Assignment of site to object, computing intercept point and proper salvo
(3) Discrimination
(4) Computation of launch altitude
(5) Determination of intercept

Due to the fact that this approach is much more time consuming than the one described in the Geometric Approach, we cannot afford to have trajectories intersect every grid square of the p_w surface; therefore, we select various likely "aimpoints" in this defended zone at which their side would aim their ICBM's. We can, perhaps, weight them, but we still intercept on this p_w surface.

The way this method is implemented is as follows:

(a) For every threat condition: aimpoints, trajectory, and warhead yield perform the next tasks
(b) Play X number of games (described below) until you obtain a set of points corresponding to an S curve (also described below)
(c) Perform smoothing to obtain the S curve and compute the P_a

(d) Perform some summation to link up all the threats

Since the purpose of a game is to obtain an S curve, I will first describe this curve. The probability of success of some happening many times depends on the number of times a particular event takes place. This "depending" is a function which when mapped looks like an S. (See Figure 9). The regularity of occurrence of the curve in analysis work is sufficient excuse to go into more detail.

Suppose, for example, n in Figure 9 represents the number of rocks thrown at a snake in an attempt to kill it. The probability for actually killing the snake should increase as the number of rocks thrown at him increases. Further, since *any* rock could do a specified job, the curve begins to rise rapidly. However, since the snake can only be so dead, the curve has to taper. This S curve is merely an expression of the normality of distribution of misses or failures which is well-known in statistics.

The same is true if n is the number of ICBM's and p is the probability of penetration, which our game's will provide.

A game is played by selecting a particular value for n and determining by random sampling whether or not the offense has penetrated. By playing enough games at that level of ICBM's, we can establish a probability of success for that n and be able to place a point on the curve. By doing this for every n, we can obtain the desired curve. Again, unfortunately, due to time considerations, we cannot afford playing games at every level or, in fact, the same number of games at all levels unless some small number of games were decided upon. Because we are only interested in a certain portion of the curve, namely around the value that gives the price of admission, P_a, we are led to another method for calculating the points on the curve. But first a word about the important part of the curve.

From what was said before we are interested in the P_a, or the number of ICBM's needed to penetrate but by the above, we see that penetration is a probability. By what means do we then determine P_a? Figure 10 gives the answer.

We want the place on the curve where if n goes beyond this number, we improve our probability very little and if n goes below this number our probability drops off quite a bit. This point is usually called the knee of the curve. It is obtained by sliding a line with some slope m up the p-axis until it becomes tangent to the curve. The point of tangency is taken to be the place where P_a is obtained. The slope of the line is a function of the "cost" of extra ICBM's versus the "importance" of the penetration. The higher the slope, the costlier the missiles; the lower the slope, the higher the importance of penetration. Therefore, we want to play most

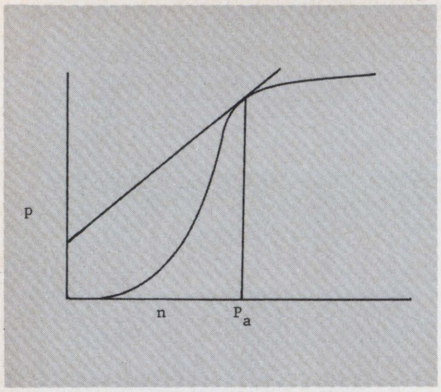


Figure 10

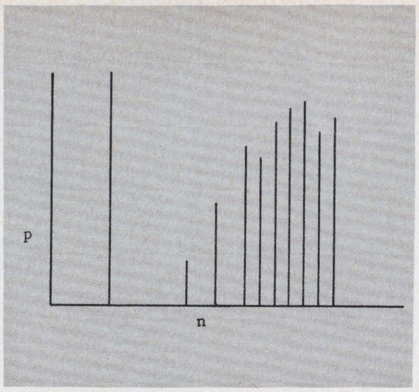


Figure 11a

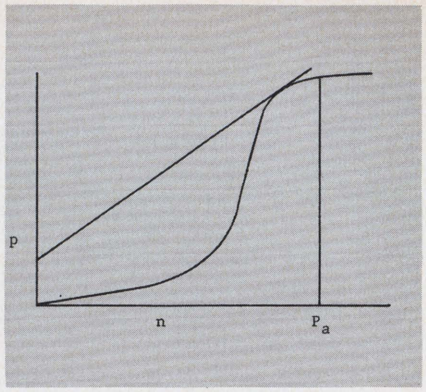


Figure 11b

of our games near the knee as that is the point of interest.

A well known procedure for the game playing is as follows:

1. Play a game with n = 1; add 1 to the number of games played at 1.
2. If you win (from a defense point of view) add 4 * q_n (integerized) to n. q_n = probability of loss at n and assumed 1 at beginning. If you lose, add 1 to the number of games played at n; subtract 1 from n unless n = 1 in which case do not change n.
3. Play the game at n and add 1 to the number of games played at n; compute q_n and unless the number of games is exhausted, go to 2.
4. When all the games are over, you typically will have a table of probabilities looking like Figure 11a. This data is then smoothed resulting in Figure 11b.

Although the two figures may not resemble one another remember that the same number of games were not played at each n.

Now to the games themselves. Since this is an event oriented simulation, we will not consider such multiprocessing defense problems as radar sweeps of sky and tracking capabilities. These multiprocessing and time sharing problems can be implemented more in the time oriented simulations.

This leads to engaging each object as it is detected. If necessary, give it a probability of detection that varies as the range of the object and draw a random number at a critical point. Detection is important since through this you can determine where the intercept will be. The aimpoint that is selected to bombard is offset because of two criteria: (1) cloud pattern and (2) accuracy-of-hitting-aimpoint indicator. The first has been preset by offense, but the second must be obtained by drawing a random number which will yield actual trajectory. The prelaunch computations are performed and a site is selected according to some rule. This leads to a computation of the launch altitude as in the Geometric Approach.

If we have no more missiles left from any site and if any of the warheads remaining successfully penetrates, we score a loss for this game. This last result depends on actual aimpoint and reliability of warhead.

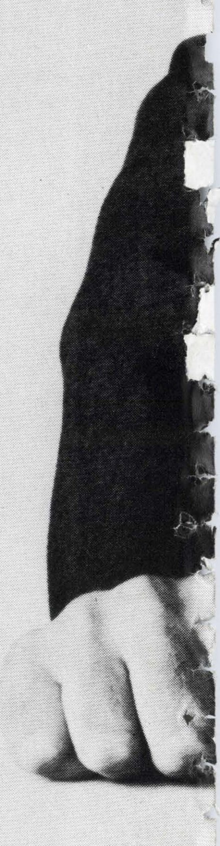
After establishing the assignment doctrine, we determine whether we must launch before we can discriminate or whether we can allow the object to penetrate assuming that it has been determined to be a decoy. We must be very certain that assumed decoy is not a warhead before we allow it to penetrate.

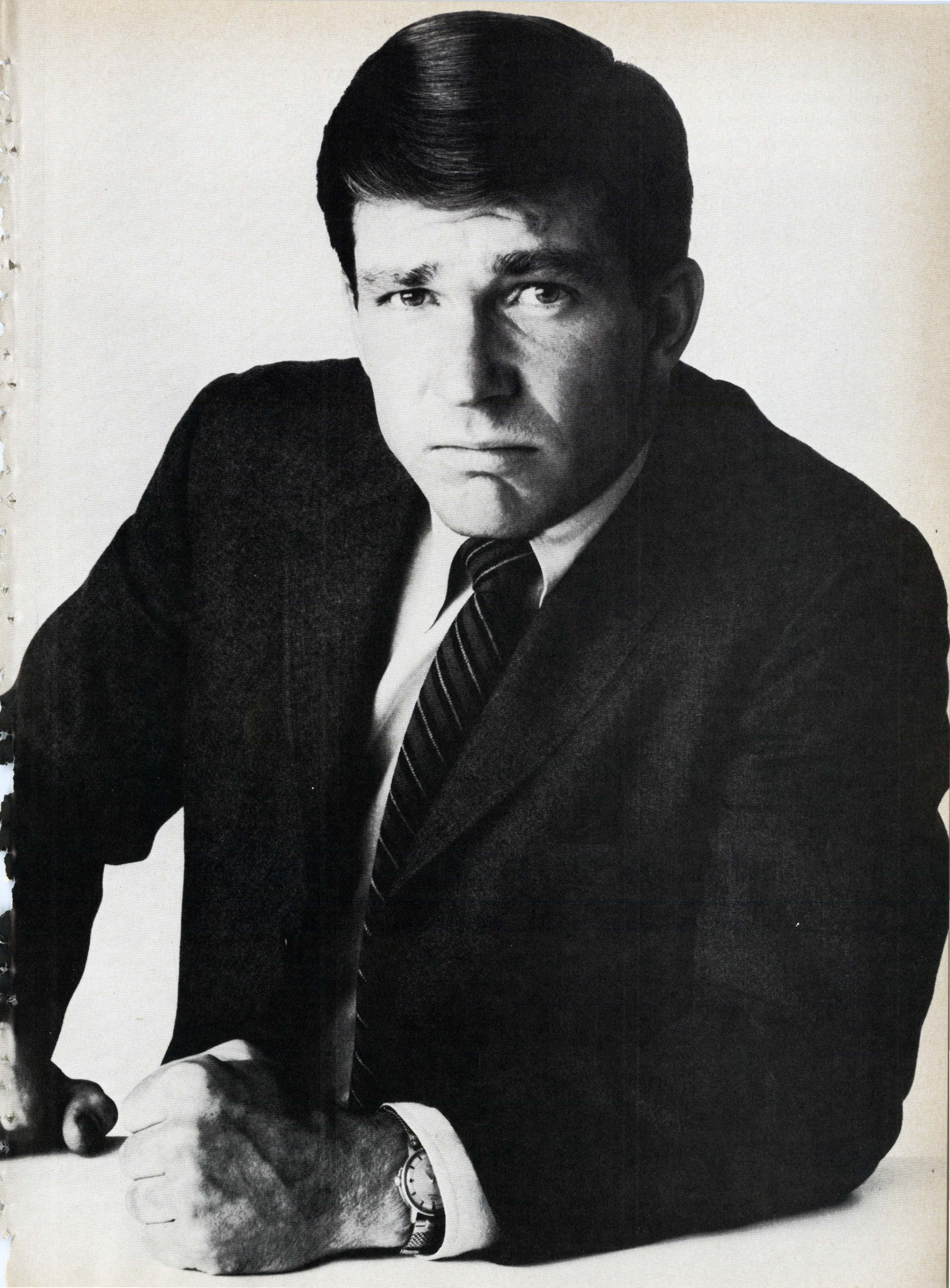
Now we must insure a certain reliability of our intercept; hence we must, if necessary, fire more than one missile at the target. The number of such is called a salvo and depends on the particular accuracy of intercept (or miss distance) at the intercept point (also on abort probability). This accuracy depends on the geometry of the intercept point. In our prelaunch calculations, then, we determine how many missiles we should send up (if we have them). As each missile is fired, random numbers are drawn and we determine if a success or failure happened. Actually, we need not waste our time with these calculations when dealing with decoys except insofar as the stockpile depletion occurs.

If any warhead penetrates, the game is lost. Every warhead of every ICBM is thus engaged. If all are knocked down, the game is won.

After determination of the price of admission via playing these games as directed, the summation takes place with respect to all the threats as in the Geometric Approach. We then obtain our merit figure.

Computer time is much more important here since one game may take precious seconds. If say, 50 games were played for each aspect of the threat and the same number of threats were considered as in SIMPLE COMPUTER MODEL —GEOMETRIC APPROACH, the number of games actually played, n_c, $> 50 * 20 * 3 * 3 * 3 * 3 * 3$, is greater than 243,000 games. If each game lasted one second, the number of hours of computer time would be around 67 hours for determining the correct deployment. This is far too much time and as it turns out, we must take away many of the niceties of the model. There





can be no sophisticated salvo size computation, nor can there be any massive site assignment doctrine. Simple, quick, and dirty methods are best. It is not difficult to have a good simulation program which can do the siting in about 20 hours. Less time than this reduces the effectiveness of the model.

Programming time is not too costly. The statistical background needed to write such a program is not intense since only simple decisions are made at each link. Such a program, once the background model has been developed, can be written in about 480 manhours.

SUMMATION

How can these approaches be applied to other game-theoretic applications?

The underlying motif is outlined as follows:

(1) Obtain an objective that is programmable. We setup the "goodness" of the plays in such a way that we obtain an equation as the one below.

$$g = \sum_i w_i m_i \text{ where } w_i \text{ is the weight of the } i^{th} \text{ merit } m_i$$

(2) Determine a game theoretic minimax that is independent of which plays can actually be made. This is the most difficult since it usually involves some very sophisticated mathematics. This minimax will be used as the guideline for setting up the various merit factors. That is, how well we play (according to some merit) is measured against the minimax.

(3) Line up the plays on both sides so that most calculations that do not involve interaction can be done ahead of time. When interaction does occur, perhaps a simple calculation can be performed or a simple table search can be done.

(4) Indicate opponent's plays by likelihood rather than some prefabricated method, i.e., establish weights for each of his plays.

(5) Perform each play through to completion obtaining the final merit figures.

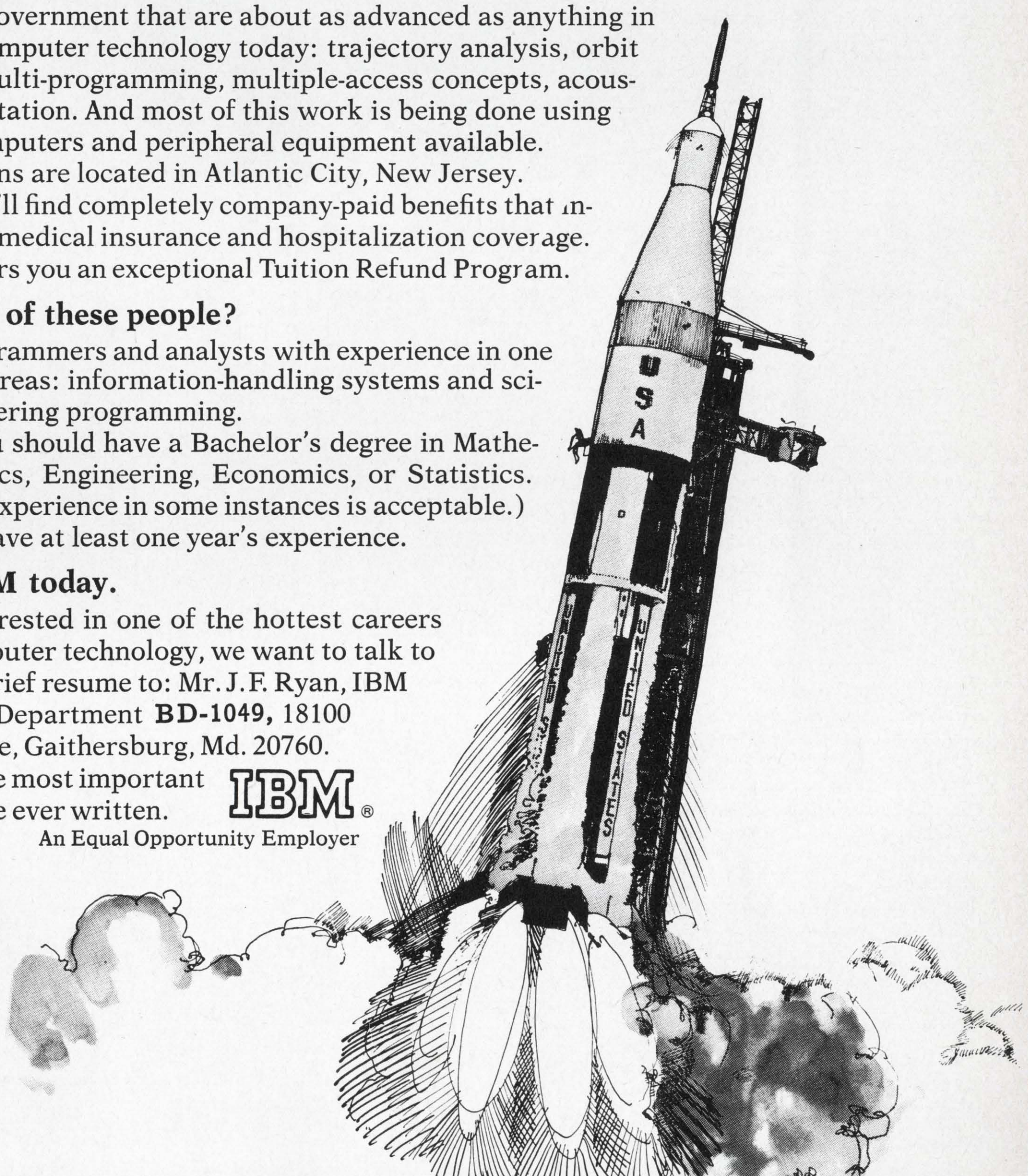
(6) Select the play that gives the highest g (see 1).

It must be reminded that computer solutions must be geared to importance of results. These programs require much computer time at a large cost and hence cannot be overly fancy.

To write a chess playing program, for example, that would play fairly "good" chess at a rapid rate by either of these two approaches, strikes me as reasonably difficult. The reason behind this is not so much the complexity of the game, but because the solution to chess is not very important. From these approaches, you pay for what you play for. ■

---

[1] By using the above symbolisms it is not intended to confuse the reader. I merely want to indicate that a mathematical method exists and that, because of this, certain computer methods are necessary to solve them.



USA
UNITED STATES

ENCODING FROM ALPHA-NUMERIC NUMBERS

(Continued from page 15)

has proposed a doubly chained memory structure that eliminates the reservation of computer memory space for the unused nodes that are represented by blank cells in Fredkin's Tries. But passing through a node via the second (continuation) chain does not involve that node's symbol in the symbol sequence, so extra linkages are needed. Scidmore and Weinberg[9] have proposed a tree structure which has no unused nodes also, although three linkage addresses are required at each node and frequently not more than one is used.

Cheyleur[2] has proposed the physical construction of a memory element utilizing not only sharing of initial symbol strings, but the sharing of polygrams in general.

Such symbol trees are of particular advantage in working with data of variable length. Some tree implementations are particularly easy to update.

Recalling that a "symbol" need not correspond to a character of the original alphabet, at least one comparison and address calculation must be performed for every symbol of the symbol string being traced through the tree. The symbol tree cannot give a false external memory address for an item not contained in the tree.

"Hashing" Transformations

In this method, first described as "open addressing" by Peterson[7], external memory is divided into equal sized portions, called "buckets". The term is transformed to an address by some algorithm. Such transformations may include selection of digits or bits, "folding" by adding some parts of the index item to others, or taking a remainder after division. The address so found is one of a bucket and is taken as the starting point of a directed cyclic scan of that bucket and all following buckets. The scan continues until either the desired item is found, or a flag is encountered. The flag indicates an unused memory location in which the item would be placed had it existed in the file.

Several transformation algorithms are discussed in one of the references[10] in which the use of a table of partial index items is suggested to reduce the number of occurences of identical addresses corresponding to different index items.

As the allocated memory becomes full, the number of searches for an item grows. Chaining from full buckets instead of the use of cyclic structure has been suggested.

Such items use a relatively small amount of high speed memory, but have the disadvantage that unless the complete file can be tested in advance the performance of the system is very difficult to predict.

Triplet Searching

As this method, proposed by Rubinoff[8], has not been described in the open literature, it will be covered in somewhat greater detail here than the other methods.

The first step in the triplet method is the arrangement of all the terms into lexicographic order and partitioning them into groups. The groups are not necessarily of the same length, as two restrictions are made on the partitioning*. The first restriction is that the groups may not exceed some predetermined number of index items and the second is that the initial K characters of the last item of a group may not be identical with the initial K characters of the first item of the following group. In other words, it is possible to distinguish the index items at a group division using only the first K characters of the index items. K is chosen depending on the equipment used, and it is convenient to make K the number of characters contained in a word if a fixed word length machine is used.

In order to search for a term, first the group in which the term lies must be located. This may be done by performing a binary search of the list of group identifiers, these group identifiers being the first K characters of the first term of each group. That is, a relatively short list of single words is searched so that the group in which the term falls may be identified.

After the correct group has been found, a list of *triplets* that have a one to one correspondence with the terms of the group is brought into core memory. The triplets consist of

* Another possible reason for modifying the length of groups is mentioned below.

an integer which need never exceed the length of the longest term, one symbol, and the linkage address (which may be a relative address) to external memory. Such a list of triplets is illustrated below: in the sample shown the different linkage addresses are all represented by μ; b̸ is used for a blank or space character immediately following the last non-blank character of a term and ∅ indicates an unused character position in a computer word.

Except for the two double triplets shown, with this scheme information is generally packed so that only one machine word is required for each term reflected in a triplet group, if a fixed word length machine with a reasonably large word is used.

Now, after the correct group of triplets has been found by a search of the group identifiers and the triplet group fetched into core memory, the triplets are searched in the direction shown above. If α is an integer and X a symbol, generated from the index as described below, then the comparison for the triplet $\alpha X \mu$ is as follows: the α^{th} character of the complete term is compared with the symbol X. If they are identical, then the search has been completed and μ is the desired address. If not, then the next triplet in the group is so inspected. If the last triplet (at the top of the list in the example) is reached with no successful comparison then the term is not in the file.

ORIGINAL INDEX

DIRECTION OF SEARCH ↑

ITEM	TRIPLET
TOIL	4Lμ
TOILE	5Eμ
TOILER	6Rμ
TOILET	6Tμ
TOILETRY	7Rμ
TOILETTE	7Tμ
TOILFUL	5Fμ
TOILSOME	5Sμ
TOILWORN	5Wμ
TOIT	4Tμ
TOKAY	3Kμ
TOKEN	4Eμ
TOKOLOGY	4Oμ
TOLA	3Lμ
TOLAN	6b̸∅3Lμ
TOLBOOTH	4Bμ
TOLD	4Dμ
TOLE	5b̸∅4Eμ
TOLEDO	5Dμ
TOLERABLE	5Rμ
TOLERANCE	7Nμ

In a few cases triplet pairs are required, which indicate two required conditions. The first is the length of the term, and the second a comparison as before.

Referring to the example triplet group, suppose that the μ address corresponding to the term TOLD is required. (For this discussion TOLD will be considered to be justified left in a field containing as many blanks as necessary.) First, the list of group identifiers (initial K characters of initial index items of groups) is searched, and the triplet group shown above is brought into fast memory. Then the triplets are examined one at a time. Since the seventh character of TOLD is not an N, the first triplet is not the correct one.

Similarly, the second and third triplets indicate R and D as the fifth characters of the complete term, which they are not. The fourth triplet indicates that the term is four characters long by 5b̸Ø (blank at end of word in the fifth position) but requires that the fourth character be an E by 4Eμ. This latter condition is not met by the word TOLD. Finally, the next triplet does indeed correspond to the correct term, and is 4Dμ. The match occurs when it is found that the fourth character of TOLD is D.

If a term is not in the system, a false hit is possible; entering the search process with TOLERABLY would retrieve the μ address associated with TOLERABLE. For this reason, in most applications the full term should be repeated at some location in the external memory record specified by the μ address, so that it may be checked.

Mention has been made before of the fact that the μ address may be partial or relative. Since the group identifier is already determined when the triplet search begins, there may be an additional address segment or base associated with the group identifier.

The triplet group is generated in the reverse direction from that in which it is searched (that is, from the top down in the example). The first term (TOIL) is assigned its last letter as its character identifier 4L and an appropriate μ. The second, third and all succeeding terms are compared character by character with the terms above them, scanning left to right. When the first character position in which the term being inspected differs from all items above it is found, the character and column position become the first two elements of the triplet. Thus, TOILSOME is assigned the triplet $5S\mu$ because T has appeared before in column 1, 0 in column 2, etc., but nowhere before has S appeared in column 5. When, as in the case of TOLE, the scan goes to the end of the word, the first half of a double triplet is used to indicate the length of the word (5b̸0̸), and the scan starts again, from right to left, but this time comparing column positions with only the previous terms of the same length as the current index item. Note that it is possible for this scheme to fail:

ABC	3 $C\mu$
BCA	1 $B\mu$
CAB	1 $C\mu$
CBA	Cannot be assigned.

In such a case, it becomes necessary to shorten the group of index items.

Balanced Trees

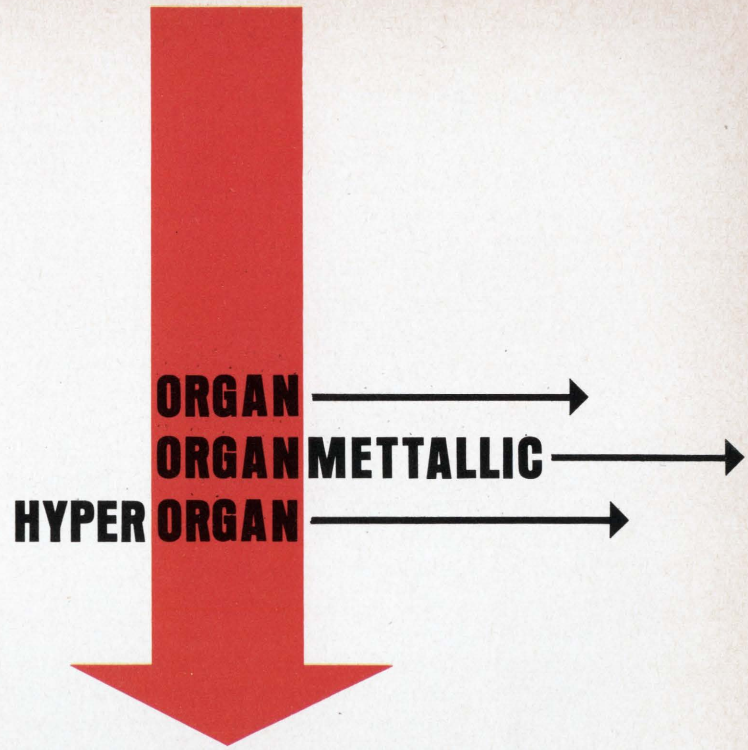
A tree encoding mechanism quite different from those discussed before has been described by Landauer[5,6]. In this tree every branching node (called the "associative catena"), except those on levels of the tree not completely filled, has a fixed number of branches. A term is also part of each node. Suppose that items are arranged in lexicographic order: $I_1, \ldots, I_N$. If the branching factor is m, then m branches come from every node. In particular, suppose that a node associated with I_j appears on level k of the tree. Then the filial set of node I_j on level k is the set of nodes $I_j, I_{j+1}, \ldots, I_{j+m-1}$ on level $k + 1$. If the tree has T levels, then there are m^T branches leaving the bottom of the tree; in the particular application being considered these branches would be links to external memory records.

Such a tree may be traced rapidly. Landauer has developed theories of searching, generating and updating them. The primary disadvantage of

(Continued on page 38)

Software design for 3G hardware . . .

STERILE



System of Terminology for Retrieval of Information Through Language Engineering

John Canter
Chairman of Department of Information Sciences
and Director of the Computer Center,
Point Park College

An investigation into the limitations of existing software systems for information retrieval applications has led to the development of a novel retrieval methodology. The system is entitled STERILE (System of Terminology for Retrieval of Information through Language Engineering). It permits the selection of information from structured indexes and/or from free text utilizing either permuted word forms or letter/character patterns, in addition to the traditional full-word-for-full-word matching.

■ Third generation computer hardware systems are a reality. The burden of responsibility for efficiently and effectively utilizing the inherent capabilities of the new families of computer systems now lies with the software community. In no other area of computer applications is this fact of life more forcefully brought to bear than in the area of information storage and retrieval.

The information explosion, which to a large degree was spawned by the emerging technologies of the World War II era, and is reaching maturity during the Space Age, has led to a variety of mechanized systems for isolating and retrieving applicable materials in response to a search query. For the most part, these techniques have revolved about the reworking of traditional word-for-word matching applications, in which the search term (in absolute or coded form) is required to be identical in form, num-

John Canter graduated in 1960 from University of Pittsburgh, Phi Beta Kappa. He is currently Chairman of the Department of Information Science and Director of the Computer Center at Point Park College, which will offer the first B. S. degree in Information Science this year. He previously worked with the IBM Corporation as an instructor in unit record and computer systems, served as director of the computer center of a multi-state retail drug chain, and served as Assistant Director of the University of Pittsburgh's Knowledge Availability Systems Center.

ber, etc. with a corresponding member of an indexing array before retrieval is achieved.

An analysis of applied indexing terminology, and an investigation into the problems associated with textual or free form retrieval has revealed the possibility for using a different approach for developing IR software systems. In many substantive areas, there is a repetition of word bases or sequential letter strings common to all related terminology. Thus, the letters POLY are common to all polymer terms. The letters LUBR occur in "lubricant", "lubrication", "high temperature lubrication", etc. For steel, the letters STEEL recur regardless of whether the subject is austenitic, bainitic, martensitic, high strength, stainless, etc. In the halogen family, IODO, IODI, HALO, HALI, FLUOR, CHLOR and BROM form the nucleus of thousands of applied chemical terms.

Consequently, a second mechanism for software development in information storage and retrieval systems is to retrieve documents from either textual or structured arrays based upon the occurrence of word bases and number/character strings rather than upon the matching of full word search analytics and full word document entries. In this manner, if a document were to discuss any of a multiplicity of similarly-related compounds, mechanisms or situations, and the indexer were to utilize any one of this structure of available terms, then the document, rather than being lost to the system, would be retrieved and made available for analysis and review. This is especially true in light of the changing terminology in which our society exists, with new scientific terms appearing daily.

A system which permits root word or character string searching has been developed. The system is entitled STERILE (System of Terminology for Retrieval of Information through Language Engineering). It permits the selection of information based upon permuted word forms or letter patterns, in addition to the traditional full-word-for-word matching.

STERILE is an information retrieval methodology which encompasses word analysis in both construction and meaning, strategy development, and application of these factors to an information file.

Mechanized and/or manual information retrieval operations revolve about the matching of terminology in a search question with the indexing terminology of the documents in a file. Whether the matching is accomplished by the needling of marginal hole-punched cards, by the manual correlation of the index entries and the search analytics, or through the electronic comparisons of digital computers, the operations are relatively similar. Terms may be joined together in a query by using "and", "or", or "not" connectors to form a search strategy, and the terms and connectors are then related to the indexed information. Documents matching the requirements of the question are isolated and retrieved.

STERILE does not alter any of the foregoing operations. It does, however, directly impinge upon the manner by which selected search terms are implemented.

As an example, an inquiry to a materials-oriented specialized information center may request "all documents relating to organic materials and their uses". In selecting search terms, an analyst would be expected to call for: organic; organic aluminum compound; organic boron compound; organic chemistry; organic compound; organic coolant; organic cooled reactor; organic fluorine compound; organic germanium compound; organic laser; organic liquid; organic lithium compound; organic material; organic moderated reactor; organic tin compound; organometallic; organometallic compound; organometallic polymer; etc. It is assumed that these are all viable terms for the given information center. Under existing techniques, the above series of terms would be searched, each term individually and specifically, in order to determine those documents which are applicable to the search query. However, STERILE alters the traditional approach, and requires, as the intervening step between selection of search terms and conducting of the search, an analysis of the selected word forms. The purpose of the analysis is to determine if, in the corpus of analytics chosen for searching, a commonality exists. In the example under consideration, the letters O-R-G-A-N are common to all terms.

STERILE operates on the prediction that it is unnecessary to match complete word for complete word, as in traditional information retrieval systems operation, and rather that a partial match of carefully selected word bases may be more efficient and comprehensive a methodology to apply to literature searching. In the above example, then, STERILE would suggest the use of ORGAN as a search term. There is, however, an obvious inadequacy, and that is that the letters O-R-G-A-N form a unique English word which has no relationship to the search query. STERILE encompasses a method for eliminating this objection.

While current indexing systems utilize three types of connectors to relate the strategy terms: "or" (+); "and" (*); and "not" (—), STERILE develops three additional connectors. They are: "permuted or" (@); "permuted and" ($); and "permuted not" (#). The traditional connectors, in STERILE, are known as "absolute or" (+); "absolute and" (*); and "absolute not" (—).

When an "absolute" connector is used, it indicates that the particular term to which the connector relates is to be treated as in conventional systems (i.e., in exactly the manner in which it appears), and as an entire-term-for-entire-term match.

STERILE's permuted connectors (@, $ and #) act in somewhat the same manner as do the traditional linking devices, with one major exception. The permuted connectors call for a methodology of document selection based upon the occurrence of specific letters in the indexing analytics in a predetermined pattern, regardless of whether or not the letters are meaningful as an entity, and whether or not they form a specific term. Thus, in using ORGAN as a permuted search term, the desire is not to isolate all documents containing the specific index term ORGAN, but rather to locate all documents which have an index entry containing the letters O-R-G-A-N, in that uninterrupted sequence, somewhere in the body of the term. Such terms would include ORGANIC, ORGANOMETALLIC, and HYPERORGANIC in which the letters O-R-G-A-N appear in a skein. It does not matter in the STERILE system whether the desired configuration of letters is the physical prefix, suffix, or base of the indexing term, since the scan is by letters and not words.

In the STERILE system, then @ORGAN specifies the selection of documents in which ORGAN appears in any portion of an indexing analytic. $ORGAN calls for all materials indexed by a permutation of ORGAN in conjunction with some other specified term or terms. #ORGAN requests that documents containing indexing terms with permutations of ORGAN be withheld from retrieval.

In the example requesting "all documents relating to organic materials and their uses", STERILE would call for a strategy of @ORGAN–ORGAN–ORGANISM–ORGANIZATION. The @ORGAN calls for all documents indexed by any word of which ORGAN forms a part; however, this is further restricted by negating those documents for which the letters O-R-G-A-N are derived from ORGANISM, ORGAN or ORGANIZATION.

In another situation, an information file user may request "information on triodes, diodes, tunnel diodes, and other related devices". By applyling the STERILE technique, term analysis reveals the commonality of the letters I-O-D-E. While these letters may not form a meaningful word, the IODE base may nonetheless be applied for searching. The STERILE format would be @IODE.

Of course, the STERILE system will not, with the @IODE strategy, locate documents indexed under SOLID STATE DEVICE, TRANSISTOR, etc., unless the system to which STERILE is applied contains an automatically-assigning thesaurus or cross-indexing and cross-referencing mechanism. In that case, the @IODE strategy could generate a series of additional STERILE and traditional terms and bases, such as ISTOR, SOLID STATE, etc. And as with conventional retrieval schemes, the selected word bases and full terms can, in STERILE, be related into complex search strategies, in which both STER-

ILE and traditional connectors may be freely intermixed.

Within STERILE, the degree of generality or specificity may be controlled on four levels. The first level relates to the analytics which are selected for searching. FABRY–PEROT LASER will be far more selective in its document retrieval than will the generic SOLID STATE DEVICE.

The second level is tied directly to the method by which the search terms are developed by the analyst. If generality is desired in a search dealing with fluorine and fluorine compounds, the @FLUORO would be called for. If specificity is desired, such as for fluoromica alone, +FLUOROMICA may be utilized, in which case STERILE will operate as a traditional full-word-matching retrieval system. Compounded terms, such as "Information Storage and Retrieval", will also lead to more specific results.

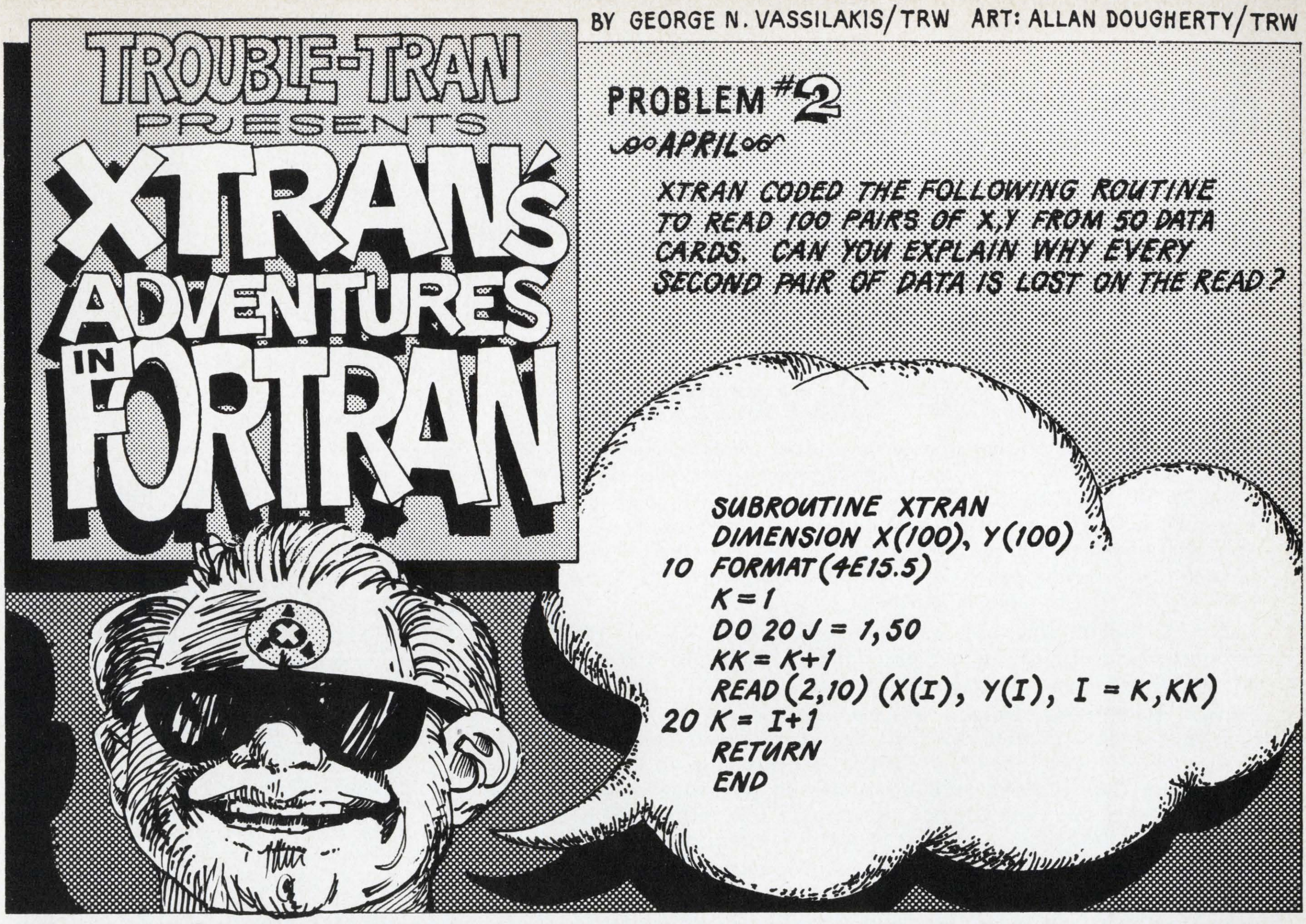
The third level of control which is available under the STERILE system, and which permits added flexibility in retrieval operations, is accomplished through the optional use of analog type (x out of n) searches. A weighted numerical factor, based upon the relative hierarchical values of the search analytics to the question, is assigned to each term in a retrieval strategy. Terms may include whole words or STERILE bases. A "hit level" or numerical value which a document must attain or exceed in order to be retrieved is established, and the search is conducted. If, for example, a strategy of @ORGAN @POLY @LUBR @TEMP +USE were created, numeric values of 2,3,2,1,1 might be assigned to each of the respective terms, and a "hit level" of 3 established for the search. Any document indexed under a combination of terms whose collective numerical value equals or exceeds 3 will be retrieved.

Finally, on the fourth level, the STERILE system provides the user with the option of searching text and/or structured indexes. If text is selected, the user may then designate whether the entire text, or only some sub-portion or combination, such as titles only, abstracts and last paragraphs, etc. is to be used.

There is only one mechanical difference which separates STERILE from conventional searching schemes, aside from the three additional connectors and the use of abbreviated permutations when desired. The difference is that each term in the strategy, including the first term, must be preceded by a designated connector. A connector of either (@) or (+) must be used before the first term to denote the mode of that term, either absolute or permuted.

With its ability to examine permuted partial terms, STERILE permits the analysis of terminology based upon word roots, upon operation-denoting suffixes, and upon prefixes and other delimiters. As such, it is especially suitable for third-generation digital computer-based information retrieval systems. ■

BY GEORGE N. VASSILAKIS/TRW ART: ALLAN DOUGHERTY/TRW



TROUBLE-TRAN for fun and profit

Send your ANSWER to the problems posed here in each issue to:

TROUBLE-TRAN EDITOR

software age

1020 Church St., Evanston, Illinois 60201

The correct answer bearing the earliest postmark will net the reader submitting it $25.00

The second correct answer with earliest postmark wins $15.00

You can also profit by submitting PROBLEMS for this feature. If your problem in FORTRAN programming is selected for use in this feature, you will receive $25.00

Contest Rules:

1. ASA standard FORTRAN IV is assumed.
2. The IBM 7094 FORTRAN IV compiler is used in verifying answers.

XTRAN'S ANSWER TO LAST MONTH'S PROBLEM

DID YOU SAY THAT ONLY SUBROUTINE YTRAN WOULD COMPILE CORRECTLY? YOU ARE RIGHT.

THE FORTRAN COMPILER WILL CLASSIFY STATEMENT 10 AS AN ARITHMETIC STATEMENT FUNCTION. SUCH FUNCTIONS HAVE TO APPEAR BEFORE ANY EXECUTABLE STATEMENTS.

MOST COMPILERS TRY TO CLASSIFY EACH STATEMENT FIRST AS ARITHMETIC AND IF THIS FAILS THEY SEARCH FOR THE PROPER NON-ARITHMETIC.

ENCODING FROM ALPHA-NUMERIC NUMBERS

(Continued from page 32)

such a tree for the proposed application is that it is not well adapted to the use of variable length index items.

Conclusions

The choice of any scheme must depend on the application. The purpose of this paper has been only to review briefly some of the available techniques. The results are summarized in tabular form.

Acknowledgement

The author wishes to thank Dr. Morris Rubinoff, Principal Investigator of The Moore School Information Systems Laboratory, University of Pennsylvania for his suggestions and encouragement. ■

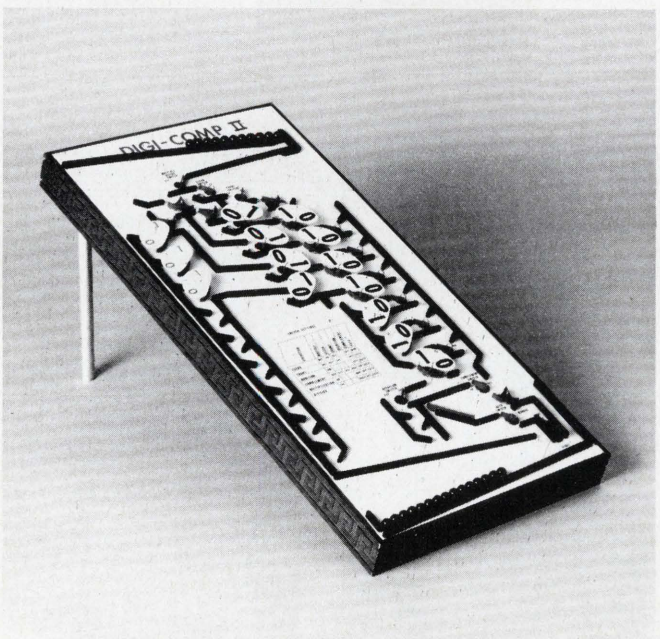
BIBLIOGRAPHY

1. Bowlden, Henry J.: A List-Type Storage Technique for Alphameric Information, *Comm. ACM,* 6:433, 1963
2. Cheyleur, Benjamin F.: Dimensioning In An Associative Memory, *Vistas In Informative Handling,* Paul W. Howerton and David C. Weeks, ed., VI. London: Cleaver–Hume Press, 1963
3. Fisher, Stephen B.: *An Executive Control System For Information Retrieval Via A Remote Console,* M.S. Thesis, The Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, 1966
4. Fredkin, E.: Trie Memory, *Comm. ACM,* 3:490, 1960.
5. Landauer, Walter I.: The Balanced Tree and Its Use In Information Retrieval, *IEEE Trans. Electronic Computers,* EC12:863, 1963
6. Laudauer, Walter I.: *The Tree As A Strategem For Automatic Information Handling,* Doctoral dissertation submitted to the University of Pennsylvania, Philedalphia, 1962
7. Peterson, W. W.: Addressing For Random-Access Storage, *IBM J. Research and Development,* 2:130, 1957
8. Rubinoff, Morris: *Information System Design For The Information Processing Field.* Proposal submitted by The Moore School of Electrical Engineering, University of Pennsylvania, to the U.S. Air Force Office of Scientific Research. Philadelphia, 1965
9. Scidmore, A. K. and Weinberg, B. L.: Storage And Search Properties of a Tree-Organized Memory System, *Comm. ACM,* 3:28, 1963
10. Steward, Joseph Robert: *The Use of Abbreviated Tables for Addressing Random Access Storage,* M.S. Thesis, The Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, 1964
11. Sussenguth, Edward H., Jr.: Use of Tree Structures for Processing Files, *Comm. ACM,* 6:227, 1963

Now . . .

The "second generation" of the famous DigiComp,

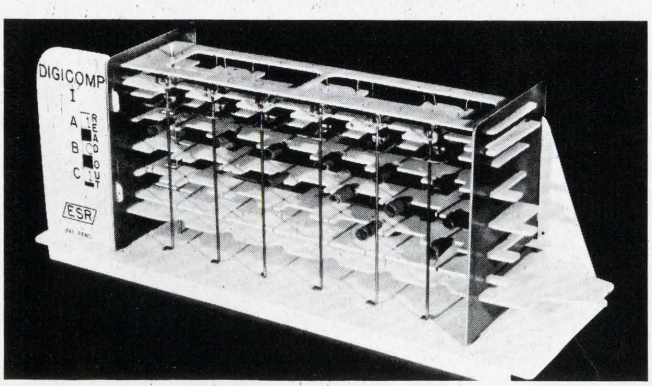
enormously more "sophisticated" as additional "switches" and "core positions" permit many more decisions and "branching" possibilities.



Like its predecessor the DigiComp-I (see below) it is entirely mechanical in operation but affords the functions common to electronic computers in a form far more comprehensible. The best way ever to learn (or to teach) the principles of modern data processing.

$16.95 Postpaid

DIGITAL COMPUTER

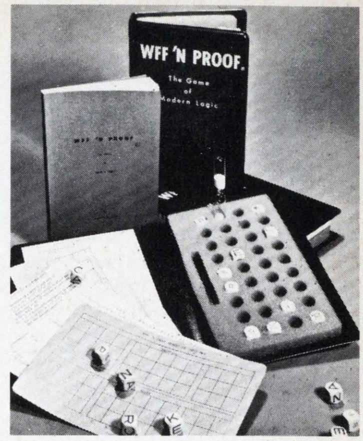


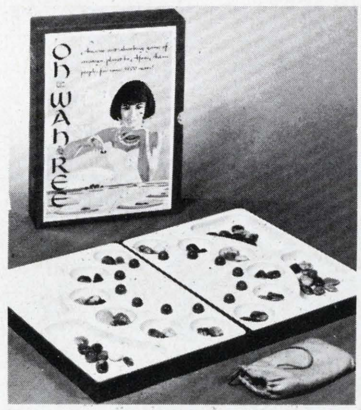
Ingenious mechanical equivalent of an electronic digital computer. Can be "programmed" to perform, individually, every operation of a digital computer. Best thing ever for teaching binary math, how a computer operates, how and why problems must be broken down for machine solution. In kit form—1 hr. to assemble. $5.98 Postpaid

WFF 'N PROOF

Developed in the course of the ALL Project (Accelerated Learning of Logic) Wff'nProof incorporates 21 games of progressive difficulty; has been used to teach propositional calculus to elementary school children but challenges the intellect of professional logicians. Complete with 224-page manual which is an introduction to the world of modern logic; filled with valuable tips to clear thinking.

$6.25 Postpaid





OH WAH REE

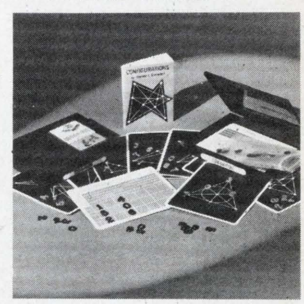
The ancient game of mathematical strategy which has intrigued millions for more than 30 centuries. A game at once so simple in its basics that a pre-schooler can enjoy it yet so complex in its possibilities that its fascination multiplies with the mathematical grasp of the player.

$7.95 Postpaid

CONFIGURATIONS

Based on concepts from the geometry of incidence, CONFIGERATIONS is a series of intriguing mathematical and geometric puzzles that will challenge and delight those who enjoy careful reasoning. It is authored by Professor Harold L. Dorwart, Chairman of the Mathematics Department, Trinity College, Hartford.

$4.95 Postpaid



TWIXT

Another famous "bookshelf" game, package matches that of Oh-Wah-Ree. A stimulating game with a fascinating chess-like strategy of move and countermove. Two or four players.

$7.95 Postpaid

SCIENCE/SYSTEMS P. O. Box 1176 Evanston, Ill. 60204 **SA-4**

Please send me at the address below:

. . . .

. . . .

. . . .

. . . .

Save $1 by ordering any 3 items or $2 on any order for 4 or more of the above

(Games are shipped from separate warehouses)

____
(name)

____
(address)

____
(city) (state) (zip code)

Illinois residents add 5% OES

new products

A new **proprietary software package called FORCE-III** has been announced by Honig, Time Sharing Associates, Inc. FORCE-III is said to operate out of a partition of OS/360 MFT. The subsystem contains a command language interpreter and file maintenance subsystem, a line editor for creation and correction of files, a FORTRAN IV (G level) compiler, a subroutine library, and scheduling and execution control monitor. Using FORCE-III, object programs may be catalogued and filed, thus permitting a library of programs which may be used repeatedly without recompilation. Files may be translated to, or generated from, catalogued OS data sets. Object programs, created and debugged under FORCE-III may be run under OS in the batch environment and, finally, OC can run simultaneously via multi-programming in the background, and provide normal batch processing at the same time. The system is designed for a 360/Model 50 or greater.

For more information circle No. 80 on the Reader Service Card

* * *

United Computing Corporation has announced two new **software simulators** for the SDS SIGMA 5 and 7 computers. UNITE II simulates the SDS 9300 series equipment. UNITE III simulates the CDC lower 3000 series computers.

The UNITE programs execute direct object code for the computers being simulated. This allows codes written for these computers to be run directly on SIGMA without modification. These simulators handle all hardware operations such as: boot strap-loading, I/O operations, interrupts, and console display and switch functions.

For more information, circle No. 79 on the Reader Service Card

* * *

ESP—a comprehensive **system for the evaluation, selection and application of forecasting techniques**—was announced today by Software Resources Corporation.

ESP provides, for the first time in one unified package, a means of selecting the most suitable forecasting technique from the many powerful methods developed in recent years. According to Software Resources, the new system will appeal to market researchers, market planners, statisticians and others concerned with reliable forecasting of consumer demand patterns. ESP is also useful in sales analysis and evaluating the history of product demand.

Written in FORTRAN IV for the IBM 7090 and System/360 computers, output is available in both tabular reports and computer prepared graphs, Software Resources stated.

Among the techniques available under ESP are exponential smoothing, moving average, adaptive tracking, and rolling averages. In all, more than 20 different statistical and mathematical techniques are available to system users.

For more information, circle No. 78 on the Reader Service Card

* * *

Autoflow, a **computer documentation proprietary software product,** is now available for the Honeywell H-200 series. It was announced by Applied Data Research, Inc.

On the H-200 series, Autoflow directly accepts Honeywell's COBOL and Easycoder languages. It is available to H-200 users on a lease or service bureau basis and was placed recently on the GSA schedule.

Autoflow is a proprietary software product that produces high-quality two dimensional flowcharts. All statement analysis, page allocation, line drawing and rearrangement of program flow is performed automatically by the Autoflow system.

The /360 Autoflow system has been extended to accept seven additional Assembly Languages: Sleuth (Univac 1107, 1108); Symbol, Meta-Symbol (Scientific Data System 910, 920, 930); Compass (Control Data 3200, 3300, 3400, 3600); DAP II (Computer Control 24, 124, 224); Art (Univac 418); and Autocoder (IBM 7070, 7074).

For more information, circle No. 77 on the Reader Service Card

* * *

A new, compact, write only, **magnetic incremental tape recorder** for recording data in industry-compatible format for immediate data processing has been announced by Potter Instrument Company, Plainview, New York.

The ME-4210 can be connected to typewriters, keyboards, adding machines, accounting machines, cash registers, etc. Recorded data can be computer-processed without intermediate conversion. Applications include data logging, data transmission, and office machine output.

The ME-4210 records data in one-character increments on magnetic tape. Asynchronous data transfers from 0 to 60 characters-per-second can be accepted. Recording accuracy is better than 1 in 107. Data is recorded in NRZ-I format on ½", 7-channel tape at a packing density of 200 bpi. The 5½" diameter reels hold 140′ of tape.

For more information, circle No. 76 on the Reader Service Card

* * *

Remote Switching Control Module, RSCM #101, has been designed to offer remote control switching with ease and economy of installation as well as safety. The remotely located master switch may be installed as far as 3000 feet away from the RSCM and may be connected by low voltage (16 to 24 gauge) surface wiring (bell wire, etc.).

The RSCM is said to be ideal for permanent or temporary remote control applications, some of which include: central switching, instrument panel wiring, safety control, security control, shock hazard elimination, intrusion—burglar alarm and process supervisory control.

For more information, circle No. 75 on the Reader Service Card

* * *

NAVCOR's new 1050 Series **keyboard assemblies** offer flexibility in meeting today's man/machine interface requirements. Completely modular in construction, the keyboards are available in a wide variety of standard configurations, and can be adapted to special applications with a minimum of time and cost. All-electronic design insures reliability and long life in the field. Each key has a life expectancy of over 100 million operations, and repair or modification of a keyboard involves only the removal and replacement of a modular component.

A basic keyboard consists of an array of NAVCOR KRM keys mounted directly on an etched-circuit base panel. For coded outputs, a diode matrix is available either on the same etched-circuit board which holds the keys or as a separate board below and parallel to the keyboard. Optional buffer and timing circuitry, contained on a third etched-circuit board which mounts directly below the matrix, offers full double-strike protection without interfering with normal typing speeds.

For more information, circle No. 74 on the Reader Service Card

* * *

A new **ultra precision rack and pinion assembly** for digital readout systems, especially designed for original machine equipment manufacturers is now available from Spaulding Instruments, Pasadena, Calif.

Key features of the Spaulding rack and pinion assembly include: rack mountings ground four sides for precision alignment drilled and counterbored for 10–32 mounting screws on 6.4 inch centers, hardened and ground 416 stainless steel precision racks, rack assembly sealed from chips and dirt, and gear box and photoelectric encoder assembly with .0002 inch resolution.

For more information, circle No. 73 on the Reader Service Card

* * *

A new **solid state operational element** designed to perform economically and reliably in instrumentation, control, and computing systems as available from GPS Instrument Company, Inc., Newton, Massachusetts.

This new form of multiplier, formerly available only in complex instrument form or as part of high performance analog computers is said to open new horizons for the systems designer. It may be used in many applications such as squaring, multiplication of variables and division. Together with other COE-Line products, it can be used in a vast variety of systems for generation of special functions; correlation functions; convolution integrals; mixing of signals; ratio measurements; suppressed carrier modulation; VGC; resolvers; trignonometric computation; sampling; power measurement; and many more.

For more information, circle No. 72 on the Reader Service Card

* * *

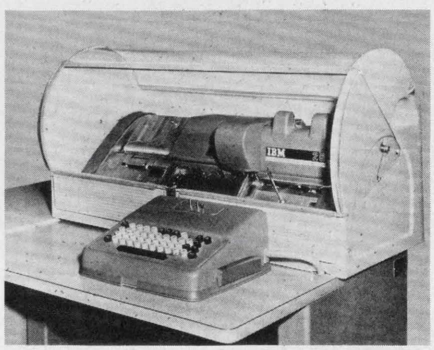
Honeywell EDP's Special Products Division have made its Model M4005 **disk pack** compatible with every computer disk drive now on the market, company officials said.

An M4005 disk pack comprises six 14-inch-diameter disks mounted on a common spindle assembly. Data is recorded on the 10 "inside" surfaces (the upper surface of the top disk and the lower surface of the bottom disk act as protective covers). A pack weighs about 100 pounds.

The 0.050-inch-thick substrates are coated on both sides with a 3.5-inch-wide circular band of magnetic iron oxide averaging 0.00023 inches in thickness. The actual recording surface is a two-inch-wide strip along the band's center line.

For more information, circle No. 71 on the Reader Service Card

* * *



A new low-cost, low-profile **silencer cover** for IBM keypunches is introduced by Periphery, Inc., Framingham, Mass.

The AWECO Silencer Cover reduces noise levels by 75% and is 50% lower in cost than comparable models. The low, compact design eliminates operator objections to the prison-like appearance of other units.

The AWECO requires no extra lighting, since all sides are completely transparent. A rotating front window permits full access to the keypunch with removal of the cover.

For more information, circle No. 70 on the Reader Service Card

* * *

Computer Accessories Corp., Huntington, N. Y. has announced a new, portable **Loader-Winder** to complement its line of winders and rewinders capable of handling perforated tapes.

Designated Model 204 LW, the loader box holds up to 1,000 ft. of one inch tape; another box is also available for 11/16" tape.

It offers variable speed control and adjustable tensioning to enable winding to any desired density. Speeds up to 1,000 fpm are possible and an adjustable slip clutch allows for high speeds or intermittent winding without tearing the tape.

For more information, circle No. 69 on the Reader Service Card

* * *

The Geo-Space CS-900-A **Channel Simulator** is a compact, portable test unit used for checking equipment designed to connect to the IBM System/360 1/0 interface. Originally designed to meet the in-house needs of Geo Space Computer Division for checking-out the DP-203 Digital CRT Plotter, CP-205 Magnetic Tape System and the 1044 Hybrid Linkage Unit, the CS-900-A may be used for a wide range of peripheral devices designed to attach to the IBM System/360. In addition, the CS-900-A is a self-contained unit including power supplies.

For more information, circle No. 68 on the Reader Service Card

* * *

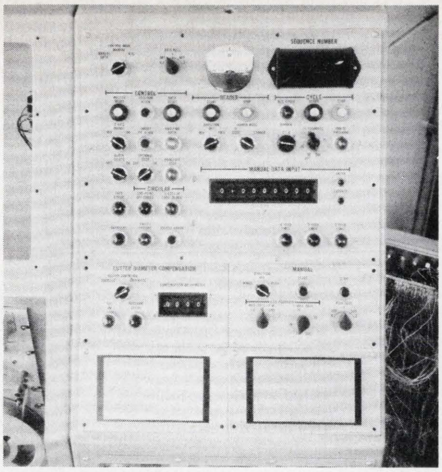
Development of a **data compressor** that could cut the computer costs of data acquisition systems by 90 percent has been announced by Stellarmetrics, Inc., Santa Barbara, Calif.

The new data acquisition sub-system is designed to take the output from high-speed analog-to-digital converters at data rates up to 100 KHZ. The device scans all information channels, but forwards only data that has changed between data samples. Read-out is in digital words that can easily be computer-translated into numerical form.

For more information, circle No. 67 on the Reader Service Card

* * *

Maintenance free **thumbwheel control switches** are being used in complex com-



puterized milling machines to insure that settings of start, stop and cutting depth are rigidly maintained.

Developed and manufactured by The Digitran Company of Pasadena, Calif., the Digitran Model 9000 sealed switches function in an environment of metal chips, oil, dirt and smoke, with no equipment down time due to switch malfunction.

The switches control the computers that operate the milling machines, by maintaining the exact parameters programmed into the system.

Other applications of Digitran switches include military and commercial airborne and ground communications, test equipment, computer control, and data handling systems.

For more information, circle No. 66 on the Reader Service Card

* * *

A new **patient monitoring system** that bridges the gap between transducers—which measure temperature, blood pressure, and other variables—and a general purpose digital computer has been introduced by Redcor Corporation, Canoga Park, Calif.

A complete network of 64 transducers and 12 display stations, in various hospitals, are connected to the Redcor system via various communication media.

Under control of a general purpose digital computer, the system monitors physical variables and events in each of the 12 monitor stations. As a result of computer analysis, the system distributes information for display at the various stations. The displayed information assists hospital personnel in diagnosis and treatment.

For more information, circle No. 65 on the Reader Service Card

* * *

A series of **circuit card guides** specifically designed for miniature integrated circuit cards has been introduced by Scanbe Manufacturing Corp., Monterey Park, Calif.

The small card guides—designated as the Scanbe M series—are for miniature cards from one inch to 2.75 inches long. With a ⅜-inch minimum card spacing, the M Series is capable of accommodating more than 500 integrated card configurations.

Incorporating only one T or H-shaped mounting bar, the M Series provides a small guide profile that permits higher density packaging and better cooling.

For more information, circle No. 64 on the Reader Service Card

* * *

A new, low-cost, **direct access storage system** designed for the UNIVAC 9000 series computers was announced today by Sperry Rand's UNIVAC Division.

Known as the UNIVAC 8410 Disc File, the new system enables users to bridge the costly gap between punched card and direct access processing, and offers unique advantages within a wide range of data processing applications.

Suitable for field-installation on any UNIVAC 9200/9300 System, the 8410 Disc File provides removable direct access storage of 3.2 to 12.8 million bytes, or 6.4 to 25.6 million digits in packed decimal format.

A basic 8410 System includes a master dual disc drive expandable one drive at a time to a total of eight drives. Each drive holds a reversible disc cartridge with two storage surfaces; one of which is on-line. By interchanging disc cartridges, unlimited storage is provided for applications which require serial processing.

For more information, circle No. 63 on the Reader Service Card

* * *

Data-Control Systems, Inc. announces the availability of a new **high frequency Up/Down Translator** for handling data translation up to 5 MHz. Operational mode is selected through interchangeable, plug-in tuning units.

Typical uses are multichannel and constant bandwidth applications where a multiplex is translated to a higher band and summed with an untranslated multiplex for transmission or recording, or where six channels must be transmitted over a 5 MHz line.

Reference signals may be supplied by an internal crystal oscillator or an external source.

For more information, circle No. 62 on the Reader Service Card

* * *

A **Remote Recording feature,** enabling one IBM Magnetic Tape SELECTRIC Typewriter to send information to another, has been announced by the IBM Office Products Division. A telephone data set is utilized in the system for sending exact duplicates of typewritten documents at speeds of up to 150 words per minute.

For more information, circle No. 61 on the Reader Service Card

* * *

Honig Time Sharing Associates, Inc. announced the availability of an **Interface System** for use with an IBM 1130 computer which expands the capabilities of the 1130 to provide a true on-line and terminally oriented system. With this interface, users can communicate to their 1130 computer over standard voice grade telephone lines and commercially available terminals.

This interface attaches to an 1130 via the IBM Storage Access Channel, feature code 7490. Computer instructions necessary to operate devices attached to the interface are standard 1130 I/O instructions and no additional modifications are necessary within the 1130 system. This interface will allow users to attach teletypes, IBM 1050's, or IBM 2741's to an 1130 in numbers up to 15.

For more information, circle No. 60 on the Reader Service Card

software age

CONFIDENTIAL INQUIRY

Your original copy of this form will be retained at the offices of SOFTWARE AGE and will be used for no other purpose than to notify the specific firms which you have checked (on the reverse side) of your interest.

THIS INQUIRY IS IN DIRECT RESPONSE TO YOUR ADVERTISEMENT IN SOFTWARE AGE MAGAZINE

TYPE OR PRINT CLEARLY FOR PHOTO REPRODUCTION

JOB DESIRED: ______

List computer hardware knowledge (names of systems, tape, disk, terminals, etc.): ______

Programming specialties and years of experience (commercial, scientific, theoretical, experimental, analog, etc.): ______

Systems programming on which you have had development experience (compilers, assemblers, executives, monitors, O.S., etc. Indicate for what computer): ______

Programming languages used and extent of experience (COBOL, FORTRAN, etc.): ______

Applications programmed (aerospace, banking, insurance, math subroutines, compilers, etc.): ______

Systems analysis experience (card design, flow charting, operation analysis, etc.): ______

EDP management experience (include years and number of people reporting to you): ______

SALARY: ______ (current) ______ (desired) DATE OF AVAILABILITY: ______

EDUCATION: Indicate major as well as degree unless self-explanatory.

Degrees			
Years			
Schools			

EMPLOYMENT: Indicate present employment and previous jobs below.

Employer			
City			
Years	to	to	to
Title or Function			

Name ______ Age ______

Home Address ______ Home Phone ______

______ (city) (state) (ZIP code) U.S. Citizen? ______

Security Clearance ______ Location Preference ______

Marital Status ______

BE SURE YOU HAVE CHECKED ON REVERSE SIDE THE COMPANIES YOU WANT TO SEE THIS INQUIRY. PUT FORM IN STAMPED ENVELOPE AND MAIL TO:

software age
MAGAZINE
1020 CHURCH ST., EVANSTON, ILL. 60201

AFTER JUNE 15, THIS FORM WILL NOT BE PROCESSED
WRITE ADVERTISERS DIRECTLY OR WRITE US FOR CURRENT S/A ISSUE

check your interests here

Fill in the confidential inquiry form on the other side of this sheet. *(Be sure to use this form. We cannot process multi-page resumes; and this form provides all the information advertisers require to screen applicants. If further information is desired, you will hear from the advertiser direct.)* Then, circle below the numbers of those companies to which you want copies of your form sent. Mail to SOFTWARE AGE, 1020 Church Street, Evanston, Illinois 60201.

☐ **I do not now receive S/A. Please enter my FREE subscription.**

....
My Specialty

....
Prime Experience in What Industry

☐ Technical Degree ☐ Non-Technical Degree ☐ No Degree

.... ☐ I Have Analog/Hybrid Experience
Year Born

software age
1020 Church Street
Evanston, Ill., 60201