December is a good month for you to think about the years ahead

Job Openings

Engineering Development Programmers
Minimum of one year experience with remote terminal devices is especially useful, although not essential. Duties: software to support state-of-the-art remote terminal design, development and testing, and program design for latest technology equipment. Programs will be written in several languages and executed on various computers.

Environmental Engineer for Computer Installations
As a site preparation engineer for computer installations you will have full responsibility for all site work. Travel only during the week. Engineering degree required. A minimum of two years' experience in air conditioning, heating and refrigeration preferred.

EDP Programming Writers
Degree plus 2-3 years' experience in EDP writing. Duties would include the preparation of technical manuals and sales material on new or modified systems, devices, equipment or installations. The material is to be used primarily by Customer Service Technicians and Field Personnel for training, reference, education and maintenance.

Applied Programming Development
These positions are for programmers and systems analysts experienced in on-line, commercial, industrial, financial or retail applications programming. They involve working on third generation equipment. A minimum of one year programming or systems experience is required.

Competitive Product Evaluation
Your background should include three to six years' experience in marketing, either in sales or systems with a vendor, or experience directly in the evaluation area. Duties would include preparation of reports concerning the capabilities of competitive computer hardware and software; technical sales assistance to the NCR Field Marketing force; analysis of the computer marketplace of the future and development of strategies to take advantage of the market.

Field Systems
The position is in hardware/software technical support of third generation mixed tape and disc. You shall perform as intermediary in all major software problems from initial presale demonstration through the debugging stages. The position will carry full responsibility for the software support of the new Century Series and 315 Systems. Experience in magnetic file computer most helpful.

Executive Software Systems Programmers
Prefer a BS degree or two years' or more systems software experience with large scale hardware. Work is in the following areas: design and implementation of advanced time sharing, multiprogramming, and multi-processor operating systems executives. Design and implementation of on-line languages, file systems and compilers to operate exclusively in a time shared environment.

Management Information Systems
Academic background should include a broad business education with an MBA degree preferred. Experience in MIS and some knowledge of business systems as they can be automated is helpful. The responsibilities will include the determination and co-ordination of user specifications for the development of such an integrated computer-based system.

Planning Analyst
Duties would involve work on high level planning and anlysis problems. A technical degree is required. An MBA or some graduate level business training also desirable.

Address: Mr. William Stephan
Dept. SA-12
Executive and Professional Placement
The National Cash Register Company
Dayton, Ohio 45409

NCR

We are an Equal Opportunity Employer M/F

How do you know we don't need an EDP professional with your unique qualifications?

If you go on to the next advertisement, you never will.

There's no reason to be coy abou reason for this advertisement.

RCA is looking for individuals wh can participate in the vigorous gro we're making in Information Syst

Growth in total systems architectu of time sharing, multi-function, multi-user operating systems.

If you've carefully avoided narrowi your interests and your ingenuity and vision have moved you steadily toward frontier work in total syster design of hardware and software — this is the place to be.

You'll work with small group conce that give you more responsibility — but with a wide range of technologic assists from the experts at RCA.

That's our reason for this ad. If you reason for reading it is a search for projects that involve total system design — with action and rewards to match — please write us today. Experience should be in any of the following: programming, language processing, control systems, operating systems, utility systems, communicati systems, micro-programming, field sales and systems support, education, or product planning.

Write to: Mr. T. A. Beckett, RCA Information Systems Division, Bldg. 202-2, Dept. SW-27A, Camden, New Jersey 08101. We are an equal opportunity employer.

RCA

computer then uses this information to help the scientists decide which mice should be mated to produce the next generation. Through this form of selective breeding it is felt that it will be possible to develop many lines of mice, each with a different gene or genes which produce different antigens. The computer is especially helpful because the combinations of genes that can cause tissue rejection come to an estimated total of at least one trillion.

At the Lowell observatory in Flagstaff, Arizona astronomers are using computer techniques to answer questions about the universe.

Information gathered at the telescopes is electronically coded onto paper tape and sent to an IBM 1130 in the Planetary Research Center. The desk-sized system reduces these seemingly unrelated numbers to tangible results, saving thousands of pencil-and-paper calculations. In the future they would like to set up a microwave link from the telescopes to the computer and display information as it is processed.

Before the recent Mariner fly-by missions, the computer had helped analyze some 10,000 photographic plates of Mars collected since 1900. Every plate was catalogued and analyzed with the aid of the computer. The huge data reduction job was later proved to be quite accurate by the Mariner photographs.

The observatory was founded in 1894 by Dr. Percival Lowell, who predicted the existence of an unknown planet and supervised the calculations of five people who worked seven years to establish its probable position. Sixteen years after Dr. Lowell's death astronomers detected the planet with an improved telescope within 1° of where Dr. Lowell had said it should be.

Kansas City police have announced the successful testing of a computer-based crime analysis program (Law Enforcement—Manpower & Research Allocation System—LEMRAS) that will forecast when and where patrolmen will be needed to combat crime and answer calls for assistance. The Chief of Police in Kansas City has indicated that metropolitan police solve two-

An Information System for Vehicle Scheduling*

by
R. Noonan
and
A. Whinston

* Research supported in part by the Office of Naval Research and Army Research Office.

Introduction

- Our shipping department is a job sink; once an order has entered it there is no way of determining what has become of the order.
- Our dispatcher is a real genius; sometimes the routes are too long and sometimes too short, but they average out to be just about right.
- You can't use a computer to schedule trucks. What are you trying to do, tell me how to run my business?

These statements were representative of comments received from managers of companies which were either partially or wholly in the trucking business. They indicated a lack of awareness for the type of aid that a good management tool could provide.

In this article we will discuss how a modern, computer-based vehicle routing system can be an effective tool in optimally using a vehicle fleet and in providing good management information concerning fleet utilization. We were also interested in combining such a system with customer billing, sales analysis, inventory control, and other systems specific to each user. Although the system was designed specifically for scheduling trucks, it is applicable wherever a routing based principally on time constraints must be determined.

There have been a number of theoretical papers devoted to the so-called scheduling problem.[1,2,3,5] Most of these were unable to solve large problems in a reasonable amount of time. Only one approach has been implemented[4] (as far as was known by the authors) in a practical system for use in solving real-world problems. Thus, although much study has been devoted to these problems, little has been done to develop a system as a good management tool in the area of vehicle routing.[6] As a result, companies were completely dependent on the *genius* of the traffic manager and his manpower and fleet needs.

This article is intended to bridge the gap between the theoretical work in this area and the practical needs of trucking companies. We will first look at the role of the vehicle routing system as an integral part of a total management information system. Then we will examine the capabilities of the vehicle routing system and the algorithm which underlies it. Finally, we will present an actual example.

As a versatile, management tool

The vehicle routing system should be integrated with other existing computer-based systems, such as customer billing, inventory control, etc. This means that the files involved should be set up so that they can be cross-referenced by the various program modules. In the ultimate system a single entry, the customer order or bill of loading, as entered from an on-line terminal, would generate all the updates necessary to the accounts receivable file, the inventory file, the customer order file, etc. These, in turn, would generate, at the proper time: the customer bills, inventory purchase orders, the vehicle routings, etc.

In addition to the mere day-to-day routing of vehicles to customers, the vehicle routing system automatically provides management control information. First, it generates a written record of what orders were shipped on each truck, when they are scheduled to arrive, and the percent utilization of the fleet. Such information could be used to determine when increases or decreases in vehicles or men are appropriate; it could also be used seasonally to determine good vacation periods. Finally, the information could be used to verify that each vehicle is loaded with the correct amount of goods requested by each customer. This control information provided by the vehicle routing system should be invaluable in allowing higher management to apply good management control to the traffic department.

10. Multi-dimensional capacity restrictions can be allowed.
11. An average delay time can be specified for the fleet and/or individually for each customer.
12. The vehicle fleet can be broken down into types based on capacity and product type.
13. The system is written in a higher-level computer language and is designed to be computer-independent.
14. The system is capable of solving relatively large problems on relatively small machines.

Nr. of Orders	Memory Size (in bytes)
100	65 K
200	65 K
500	96 K
1000	131 K

Larger problems can be run on the smaller machines if direct access disk storage is available.

15. The system allows the suppression of the printing of individual route information, with only summary vehicle utilization information being printed.

As an algorithm

Underlying the vehicle routing system is a classification algorithm. In principle, the problem with the restrictions noted above could be formulated as an integer programming problem.[7] However, the problem formulated would involve too many equations and variables to allow it to be solved. One could solve the problem using the simplex method and simply round to the nearest integer solution, but the formulation would involve millions of variables for even a fairly simple 200 customer problem. This is still too large a problem to solve using present, standard linear programming codes. Finally, one could formulate the problem as a period-transportation problem, but a 200 customer problem with 50 periods would involve 1 million variables. Although a problem of this size is solvable on a medium-to-large scale computer with disk capacity, it does not allow for the other complexities which we have introduced and would require at least 5 times the amount of computer time.

Basically, we chose to formulate the problem as a classification problem, following earlier work done along these lines by Clarke and Wright.[3] In this way we were able to enlarge the formulation of the problem to allow arbitrary complexities without significantly increasing the size of the problem. One immediate advantage of this type of formulation was that unlike a mathematical programming model, the problem need not have a feasible solution, that is, one in which all orders are delivered in the time period being scheduled.

After the input phase, the algorithm begins by sorting the customer orders according to priority and amount. A subset of the total number of orders is assigned to individual routes. For each pair of existing routes, a measure of association is calculated as a function of their nearest points, the distance from the warehouse, and the time windows involved (the early and the late delivery times). Two orders are never assigned to the same route if the products involved cannot be shipped on the same type of truck.

The merging of routes is then attempted by order of the greatest association measure; this process continues as long as truck availabilities and size and total route distance permit. Then a new subset of orders is taken and the entire process is repeated (see figure 2).

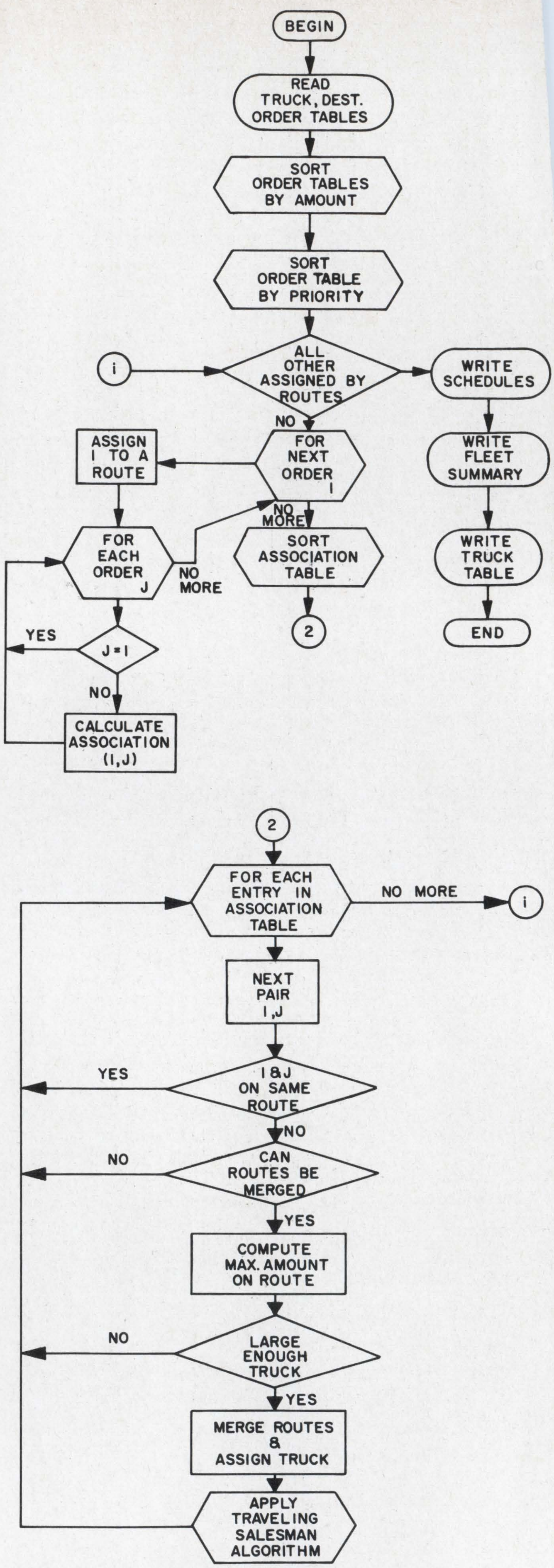


FIG. 2 VEHICLE ROUTING PROGRAM

TEST PROBLEM NO. 5

PARAMETERS

FLEET EARLY START TIME	8.00
FLEET LATE FINISH TIME	17.00
AVERAGE DELAY TIME/STOP	.06
SPEED MODIFICATION FACTOR	1.00
NUMBER OF WAREHOUSES	1

ROUTE NUMBER	5
INITIAL AMOUNT	3878
TRUCK TYPE	1
WAREHOUSE	1
TOTAL TIME	15.36

Grid Nr	Cust Nr	Customer Name and Address	Product Number	Amount	Early Time	Late Time	Arrival Time	Delay Time
1509	45		6032	520	8.00	17.00	8.36	.06
1711	20		8943	777	8.00	17.00	8.54	.06
1909	59		5963	47	8.00	17.00	9.12	.06
1807	48		8569	888	8.00	13.00	9.30	.06
1508	5		2824	900	8.00	13.00	9.54	.06
1209	66		5000	746	10.00	15.00	10.18	.06
1	1		0	2774	8.00	17.00	10.36	.06
1809	40		6858	1108	8.00	17.00	11.30	.06
1907	79		5424	1121	8.00	13.00	11.48	.06
1907	61		2340	511	8.00	17.00	11.54	.06
1409	28		1183	34	8.00	13.00	12.30	.06
1	1		0	3539	8.00	17.00	13.00	.06
1706	9		3682	1030	12.00	17.00	13.54	.06
1806	63		1640	1066	12.00	17.00	14.06	.06
1806	44		4983	1137	10.00	15.00	14.12	.06
1907	54		2840	125	8.00	17.00	14.24	.06
1908	42		3814	181	12.00	17.00	14.36	.06

SLACK TIME 0.00

ROUTE NUMBER	6
INITIAL AMOUNT	3051
TRUCK TYPE	1
WAREHOUSE	1
TOTAL TIME	15.36

Grid Nr	Cust Nr	Customer Name and Address	Product Number	Amount	Early Time	Late Time	Arrival Time	Delay Time
102	34		3943	705	8.00	13.00	9.18	.06
102	11		1271	563	8.00	17.00	9.24	.06
105	16		9635	268	8.00	13.00	9.48	.06
205	6		90	945	10.00	15.00	10.00	.06
207	29		5131	56	10.00	15.00	10.18	.06
308	80		7260	514	10.00	15.00	10.30	.06
1	1		0	3787	8.00	17.00	11.18	.06
401	68		1783	533	10.00	15.00	12.24	.06
500	71		2909	972	8.00	13.00	12.36	.06
701	3		8107	213	10.00	15.00	12.54	.06
900	24		5624	684	12.00	17.00	13.12	.06
901	49		3618	323	8.00	17.00	13.24	.06
804	25		8387	873	12.00	17.00	13.48	.06
907	51		7570	189	8.00	17.00	14.12	.06
1	1		0	875	8.00	17.00	14.36	.06
1106	53		4742	875	12.00	17.00	15.06	.06

SLACK TIME 0.00

Figure 3

An Experimental Approach to Recursive Fortran Subroutine Programming under Operating System 360

by
Herbert C. Kugel

Recursion is the name given to the process of defining a function or relationship in terms of itself.[1] A typical recursive relationship is the factorial function for the positive integer numbers:

Factorial(n) = n*Factorial(n—1), n greater than zero,
Factorial(0)=1

Recursive programming is an area that has long fascinated many people. Many algorithms have been defined and much thought given to the recursive computer program. Although the results have not always been spectacular, the problem is far from academic since recursion is very much a factor in such diverse areas as operating system generation and compiler writing.

FORTRAN has traditionally been an area where recursive techniques have had only limited success. IBM Operating System 360 FORTRAN, however, does allow the compilation of a recursive subroutine. That is, a subroutine that can call itself in exactly the same manner it might call any other external subroutine. Although the method that will be outlined is quite 'non-standard,' its implementation can lead to a fuller awareness of the fundamental concepts of software development, and, as such, is well worth some thought and consideration.

For the purposes of this article it is feasable to begin with the factorial function itself since certain of its properties are common to all recursive relationships. Each Factorial (n) is defined in terms of Factorial (n — 1) and so on back to the initial Factorial(0). The initial Factorial (0) is explicitly defined as one. There is no calculation involved, only a formal definition from which all other values can be derived. There may be more than one explicit definition. The Fibonacci sequence, for example, has two such definitions. It is the series of integers, x1, x2, ... x(n), such that every x(n) is the sum of the two preceding values; x1 and x2 are explicitly defined as one:

1,1,2,3,5,8,13,21,34,55,89............

If the name Fibonacci (n) is arbitrarily assigned to x(n), the process becomes:

Fibonacci (n) = Fibonacci (n — 1) + Fibonacci (n — 2)
Fibonacci (1) = Fibonacci (2) = 1

The same rules apply to the Fibonacci sequence as the factorial function. There are explicit definitions of the initial arguments and all other values are derived therefrom.

Thus it becomes reasonably apparent that, if the recursive process is to be programmed, the following factors must be carefully considered:

- The initial explicit definitions
- The recursive relationship itself
- The number of times this recursive relationship is to be applied.

The actual programming, obviously, must also be given careful examination. Logically, if a recursive function is defined in terms of itself, then the coding of a recursive subroutine to process that function should involve the creation of a subroutine that has the ability to call itself. For instance, if a subroutine, IFACT, is defined to calculate Factorial (n), the compiler must be able to process a structure of the form:

```
SUBROUTINE IFACT (IY,N)
      .
      .
CALL IFACT (IY,K)
      .
      .
RETURN
END
```

[1] *Recursive Techniques in Programming, D. W. Barron,* Macdonald & Company, London, 1968.

IZ(n), are defined as data constants, their location within the subroutine is completely unrelated to the condition of the general registers at any given time. Hence, they are also independent of the number of times the subroutine calls itself. Now, since the logic of the recursive function is such that the dependent variables replace the independent variables, it can be seen that this is quite easily accomplished by having all recursive calculation input and output go to and from these IZ(i). Thus, when the recursion is complete, the desired results are in IZ(i). They can be transfered to the calling program and the IZ(i) are then reset to their initial explicit definitions if the subroutine is to be used again.

This is all relatively simple, as will be seen from the following example. For the moment assume the existence of two assembly language subroutines, SET and RESET, that deal with the problem of saving and restoring the initial contents of the general registers. Now consider the following subroutine IFACT. The purpose of this routine is to set the value IY equal to the Factorial of integer (n).

```
      SUBROUTINE IFACT (IY,N)
      INTEGER IZ1/1/
      CALL SET
      K=N
      IF (K) 4,4,3
    3 IZ1 = IZ1*K
      K=K-1
      CALL IFACT (IY,K)
    4 IY=IZ1
      IZ1 = 1
      CALL RESET
      RETURN
      END
```

In IFACT, the INTEGER statement is used to define IZ1 as the explicit Factorial (0) value of one. The K = N statement takes, during the first call to the subroutine, the externally supplied N and initializes it as an internal counter, and, in this case, operand, K. This K is compared to zero. If it is not zero, a recursive calculation is performed and IZ1 is set equal to IZ1*K. The value K is then decremented and IFACT is called recursively, this time with K as an argument instead of N. This is done with all recursive FORTRAN subroutines. In fact, if each recursive call is thought of as being made to an external subroutine, say IFACT2, the above logic would reduce to the quite familiar:

```
      SUBROUTINE IFACT (IY,N)
          .
          .
      K=N
          .
      K=K-1
      CALL IFACT2 (IY,K)
          .
```

The only difference is the call. Any recursive subroutine is called once with an externally supplied N. The subroutine then calls itself (n — 1) times using an internally generated counter K. Each call results in exactly one interation. After N such calculations the process is complete, and, in the case of IFACT, IY is set equal to IZ1, which now contains the Factorial (n). It is quite important to stress that this logic is completely independent of whether N or K is involved in the arithmetic calculations. It need not have been, as is shown in the next examples.

It is simple to extend this approach to several variables. All that is required is that a unique IZ(i) be associated with each variable. Consider, for example, the Fibonacci number sequence defined above. It is possible to create a subroutine, FIBNCI, which, for any integer n, determines the three values Fibonacci (n), Fibonacci (n + 1) and Fibonacci (n + 2). The coding for this is straightforward:

```
      SUBROUTINE FIBNCI(IYN,IYN1,IYN2,N)
      INTEGER IZ1/0/,IZ2/1/,IZ3/1/
      CALL SET
      K=N
      IF (K) 3,3,4
    3 IYN = IZ1
      IYN1 = IZ2
      IYN2 = IZ3
      IZ1 = 0
      IZ2 = 1
      IZ3 = 1
      CALL RESET
      RETURN
    4 IZ1 = IZ2
      IZ2 = IZ3
      IZ3 = IZ2 + IZ1
      K=K-1
      CALL FIBNCI (IYN,IYN1,IYN2,K)
      END
```

This logic is exactly the same as in IFACT. The only difference involves the initial explicit definitions and the arithmetic calculations. The integer values at IZ2 and IZ3 are the initial explicit definitions for Fibonacci (1) and Fibonacci (2) respectively. When FIBNCI is first entered, IZ1 is set equal to IZ2, which, in turn is set equal to IZ3. IZ3 is then defined as IZ1 + IZ2. These values, after the first call, are Fibonacci (1), Fibonacci (2), and Fibonacci (3) respectively. This process is then repeated recursively (n — 1) times. During each such call Fibonacci (j + 1) and Fibonacci (j + 2) are moved down to Fibonacci (j) and Fibonacci (j + 1) respectively. A new Fibonacci (j + 2) is calculated and placed in IZ3. After this process has been repeated n times, IZ1, IZ2, and IZ3 contain Fibonacci (n), Fibonacci (n + 1), and Fibonacci (n + 2) respectively. These values are delivered to the calling program and the IZ(i) are reinitialized, as was IZ1 in IFACT.

The only programming constraint in this method is in the inherent nature of any subroutine generated by the FORTRAN compiler. These modules are designed to obtain all the arguments supplied to them by the calling program, modify some as called for by their coding logic, and then return all these arguments, both modified and unmodified, to the calling program. Since the recursive subroutine repeatedly calls itself with a decreasing variable, K, it replaces the previous K during each recursive call. Finally, at K = 0, when all arguments are returned to the calling program before final exit, this zero value replaces the initial value of N. Because of this, N must be saved if it is to be used again. This is the only major constraint that must be defined in recursive usage.

There is no problem with unmodified arguments. The recursive subroutine passes unmodified arguments to itself in exactly the same way it would pass them to any external subroutine. Consider, for instance, a subroutine, SQRTI, that uses Newton's Method to set Y equal to the square root of X. Newton's Method is defined in almost any text on numerical analysis, but for the purpose of this discussion it is sufficient to state that the method uses the recursive relationship:

$$Z(n+1) = Z(n) - ((Z(n)*Z(n)) - X)/(2*Z(n))$$

where each Z(n) is normally the same or a closer approximation of the square root of X. (Z(n) and Z(n + 1) should eventually be the same if X were a perfect square.)

S O F T W A R E A G E

R E A D E R S H I P S U R V E Y

READERS:

Help us bring you editorial material that is in your specialty.
Just fill out the coupon below and return it to us.

Return to: **SOFTWARE AGE Magazine**
Box 2076, Madison, Wisconsin 53701

I already receive SOFTWARE AGE ☐ Yes ☐ No

Enter my subscription. ☐ Yes ☐ No

The information below gives my professional status.

Name ____________________

Company ____________________

Title ____________________

Home Address ____________________

City ____________________ State __________ Zip __________

Signature ____________________

Circle the most appropriate items below to describe your present position and experience.

MY SPECIALTY:
1—Systems Analysis
2—Math/Math Analysis
3—Operations Research
4—Scientific Programmer
5—Business Programmer
6—EDP Management
7—EDP Instructor
8—EDP Consultant
9—____________________
other, incl. EDP student, operator, technician, etc.

PRIME EXPERIENCE IN:
1—Aerospace/Defense Industry
2—Non-Defense Industry
3—Commercial/Finance
4—Research Organization
5—Govt./Military
6—EDP Consulting
7—Transportation/Utilities
8—Education/Library
9—____________________
other, specify

I WAS BORN:
A—1920 or before
B—1921-25
C—1926-30
D—1931-35
E—1936-40
F—1941-45
G—1946-50
H—1951-55

I HAVE:
☐ Technical Degree
☐ Non-Technical Degree
☐ No Degree

RENTAL CHARGE ON THE EQUIPMENT I WORK WITH:
☐ $1,500 or less, monthly
☐ $1,500-2,500
☐ $2,500-5,000
☐ $5,000-10,000
☐ $10,000-20,000
☐ over $20,000 monthly

Suggestions for SOFTWARE AGE

____

____

____

____

b) Provide rapid responses to agent inquiries regardless of their geographical location.

c) Perform the varied accounting functions necessary in a reservation system.

It is extremely important to distinguish problem symptoms from the real needs. In order to design a good system, the problems must first be accurately defined. In many cases, the stated problems are only symptoms of an underlying problem of gathering, analyzing, classifying and disseminating information. The system must be designed around the problem instead of around the symptoms of a problem.

Determination of the Problem

Problems which require an on-line real-time system solution usually involve many information senders and receivers. A list must be compiled of all senders and receivers of information and their geographical locations. The amount of data being received and transmitted from each location must be determined. This data should be in the form of messages per minute, hour, or day, the peak number of messages per unit time, message length, and the duration of the peak data transmission or traffic period. One example of data senders and receivers is in an airline reservations system. Here, the data is sent and received by a number of persons at different locations. There may be several agents at city A, transmitting hundreds of transactions per peak hour. There may be many agents at city B, transmitting dozens of transactions per peak hour, etc.

System Cost

The system is normally designed to handle the peak traffic projected several years hence, rather than projections of the current average traffic. One factor which may lead to a decreased system cost is that traffic peaks may occur at different times in different portions of the system. This information should be used in designing the system. Neglect of this factor may require more equipment than necessary, causing a substantial increase in the price of the proposed system. There is little difficulty in defining a real-time system which will work. However, the problem is defining one which will operate efficiently and is optimized for minimum cost in the particular application.

Response Time

The response time delays which can be tolerated by each user should be documented. Usually the delay which can be tolerated determines the speed and type of communications network required. If the messages are short and the response time is short, a low speed communications network may be used. If the messages are long and short response times are desired, a high speed communications link may be desirable. Transmitting a message from a remote terminal to another point in the system can account for an appreciable part of the response time.

Data Transmission Characteristics

The nature of the transmitted data must be defined before designing and predicting performance of a system. For example, it may be numbers, letters of alphabet, punched card data, or other types of inputs, such as signals from counters or digital converters. In a traffic control system, for example, the data is essentially a *car* or *no car* signal between the traffic system and the computer system. On the other hand, for an airline reservations system, the transmitted data consists of alphanumeric character input and output messages called transactions. Fifty percent of the transactions may have a 40 character input message and a 120 character reply; 30% of the transactions may have an input message of 75 characters and have an average reply of 50 characters with a range between 25 and 300 characters. Ten percent of the transactions may be input messages of only 100 characters. Ten percent of the transactions may be output messages of 100 characters average, ranging from 50 to 500. It is in this manner that transactions or types of messages must be described for each geographical location.

Communications

In many systems, the communications network may already be in operation. In these cases, it may not be possible to change the network drastically. An example might be an inventory/order entry and reply system using a full-duplex communications network with over 200 cities involved. It would be impractical from a cost standpoint to alter the existing communications network. In this case, additions to the system would have to be designed around the existing remote communications facilities.

It may be necessary for the system supplier to choose the communications, particularly when the system requires very short response times. In an airline system, an average response time of two seconds may be required by some agents. Since the computer could be installed many miles from the remotes, it may be necessary to use several high speed lines to meet response time requirements and minimize communication costs.

The determination of system requir ments includes analysis of transactio and terminals to establish priority stru tures. A first-come, first-served priorit may suffice, but transaction volume and terminal types may dictate som other priority structure.

Data Files

Usually, transactions (which consis of system input and output messages) are related and require references to a Master File. In a traffic control system, the Master File may contain the present status of all traffic signals and traffic detector counts, or the model which is being used to mimic the traffic on the streets. In an airline reservations system, the Master File is a record of all seats on all flights to be sold from the present to six months hence. For a mail order company, the Master File would be the inventory record. For a manufacturing company, the Master File might be the product file.

In a real-time system, this file must be connected on-line to the computer and be available to all users directing inquiries to the file, changing it, adding items to it in accordance with designated procedures. It is important to estimate the size of this file and the number of references to it in a given time period. This ultimately determines the type of on-line random access storage required. Each file record may be of fixed or variable length.

An index to the Master File may be required due to the frequency of references to the file. The index provides ease in updating the file and helps avoid file searching. Because this index may be relatively large, its size and nature should be estimated. The nature of the file should be stated, for example, in computer words or characters to avoid confusion in estimating its size. Variable length records in the file could complicate the structure of the file index.

Multi-Programming

Sometimes, along with the real-time requirements, there is a requirement for concurrent batch (background) processing. This may be defined in terms of the characters or computer words of input and output and the type of processing involved. For example, 3,000 cards per hour input and 4,500 lines per hour of high speed printing may be desired as a card-to-printer conversion action to occur concurrently with the real-time activities.

At this point in the design process, the data necessary to determine the problem scope has been gathered. The next step in the process of designing the system and understanding the problem is to construct a trial system block diagram.

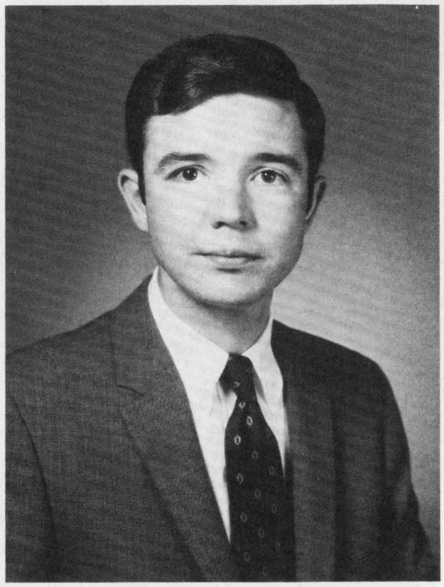
11. Determine the maximum computer I/O rate.
12. Estimate the capability of the computer.

Of the items in this list, the most important is the first: "List the major system objectives". Many times computer firms receive requests for proposals to supply systems defined in terms of desired equipment characteristics rather than in terms of the system objectives. For example, the person commissioned to buy the equipment may have determined that the system must have a punch card reader that reads 200 cards per minute. Perhaps he needs a certain amount of equipment reliability, and believes the only way to get it is equipment he specifies. In fact, one of the system objectives should be a stated level of reliability.

Most often there are many solutions to a problem. Computer manufacturers should be asked which pieces of their equipment will best satisfy the system objectives. For example, if the system objective is to locate various freight cars of interest to shippers, one solution to the problem is a computer system including a high-speed printer. Another solution is to utilize a real-time computer, communications lines and low speed teletypewriters. Another example is in an inventory application. With a batch processing solution, high speed magnetic tape units are desirable. With a real-time solution, magnetic tape capability is of minor importance.

At this point, the function of the systems engineer is complete and his design, fully documented, can be turned over to other company operating groups to be programmed, installed, debugged and put on the air. ■

James L. Rowse



Robert D. Jenkins

Thomas G. Roberts

James L. Rowse, *partner,* has a B.S. in Electrical Engineering. His experience includes research, central processor communications and subsystem design, systems analysis, mathematical models, simulation, communications network philosophy and technical marketing support of large-scale real-time multiprocessor computer systems.

Robert D. Jenkins, *partner,* also has a B.S. in Applied Mathematics. He has extensive experience in digital systems simulation, mathematical modeling, software design, and system evaluation. He also has experience with the development of computerized business systems for management.

Thomas G. Roberts, *partner,* has a B.S. in Applied Mathematics. He has experience in technical management, software systems design and evaluation, and simulation systems for business, military and scientific applications.

the other usual components of operating systems not included in the system control program class. Most compilers, utility routines, subroutine packages, and programs diagnostic routines will probably be included. Most old Type I programs not assigned as SCP will be assigned as class A program products (PP).

Class B programs probably will be application oriented programs and subroutines. Many of the old Type II programs will be reassigned to this class.

Class C programs are all other programs. These might be programs contributed by users of IBM hardware, and by IBM employees. Most of the reclassified old Type III and IV programs will be assigned here, as well as many new programs.

All new system control programs will continue to be provided and installed free of charge, but only once for each computer for each user. Maintenance (including updates), debugging, and temporary patches will also be provided free of charge as long as the user keeps up to date. These provisions will also apply to any old Type I, II, and III programs reclassified as system control programs.

For program products (programs in classes A, B, and C), the picture is quite different (see Table I). To obtain these programs, the user must make a monthly payment for each computer on which he is licensed to use the program. Installation and maintenance will sometimes be included in the license price for classes A and B. For class A programs some debugging and temporary patches will also be provided, but not for classes B and C. Also debugging on live data will probably not be provided for any program products. Class C programs will usually be eligible for no free service.

Eligibility for free service will also depend on the up-to-dateness of the software. A program otherwise eligible for free service will apparently not be eligible if it is not the most recent version, if it does not include all updates and modifications, or if these updates or modifications were not performed by IBM personnel for the user promptly after their official release. The user must permit IBM to keep his software current, to keep it qualifying for any free services for which it might be eligible.

This same concept applies also to the installation of new programs. Assuming the program is eligible for IBM-provided installation service (a system generation for an operating system, for example) then the work must be done by IBM and the result must conform to IBM's normally established practices. The user will not be able to have IBM customize a program for him and expect it free of charge. Further, if the user does have a piece of IBM software customized, then it may fail to qualify for some subsequent free services.

Starting in 1970, the existing Type I, II, III, and IV programs will be subject to the same installation, maintenance, debugging, and temporary patch policies as the new programs under the new classifications. The eligibility for free service will depend upon the service classification assigned to each of the old programs, even though IBM will continue to provide the programs free of charge.

Software License Agreement

IBM is taking a number of steps to protect its software and its revenues. One basic step is the copyrighting of the programs to establish IBM's legal right to try to control the terms on which anyone can use the software. The copyrights give IBM access to the courts, if necessary, to enforce its ownership claim on its software, and they help demonstrate IBM's interest in its software.

IBM is taking a second step. To gain revenue from the software, and to protect IBM from unauthorized use of the software, IBM has drawn up a "License Agreement i IBM Program Products." IBM assumes all who use program products agree to the terms IBM sets forth· acceptance of a program is equivalent to signing the L cense Agreement. In brief summary, the major provisioı of the License Agreement are these:

1—Use is limited to a single specified CPU.
2—The user may make no more than five copies c the program or documentation.
3—The user must not make copies available to any one, of either the program or the documentation.
4—The user is to pay a monthly charge.
5—The user may also have to pay an initial charge.
6—IBM may change the monthly charge at any time, with notice.
7—If the user modifies a program, he forfeits the right to free debugging service from IBM, a consequence of the modification.
8—Use of the program is entirely the users responsibility.
9—IBM does not guarantee the results of using a program.
10—IBM may discontinue the license upon six months' notice.
11—The user may discontinue a license with one month's notice, after the first two months.
12—When either IBM or the user discontinues the license for a program, the user must destroy all copies of the program and documentation.
13—IBM may modify the Agreement but the user may not (the user may discontinue all licenses).

System Engineering Agreement

IBM is taking a third step. As a result of its unbundling, IBM will cut off, as of January 1970, all free systems engineering services which it deems nonessential for the support of its own sales force. It is thus putting systems engineering onto a pay-for-service basis. IBM customers who wish assistance in the installation and use of data processing products can now purchase it from IBM.

To facilitate this, IBM has drawn up and expects its customers to sign an "Agreement for IBM System Engineering Services". Some of its provisions, in summary, are:

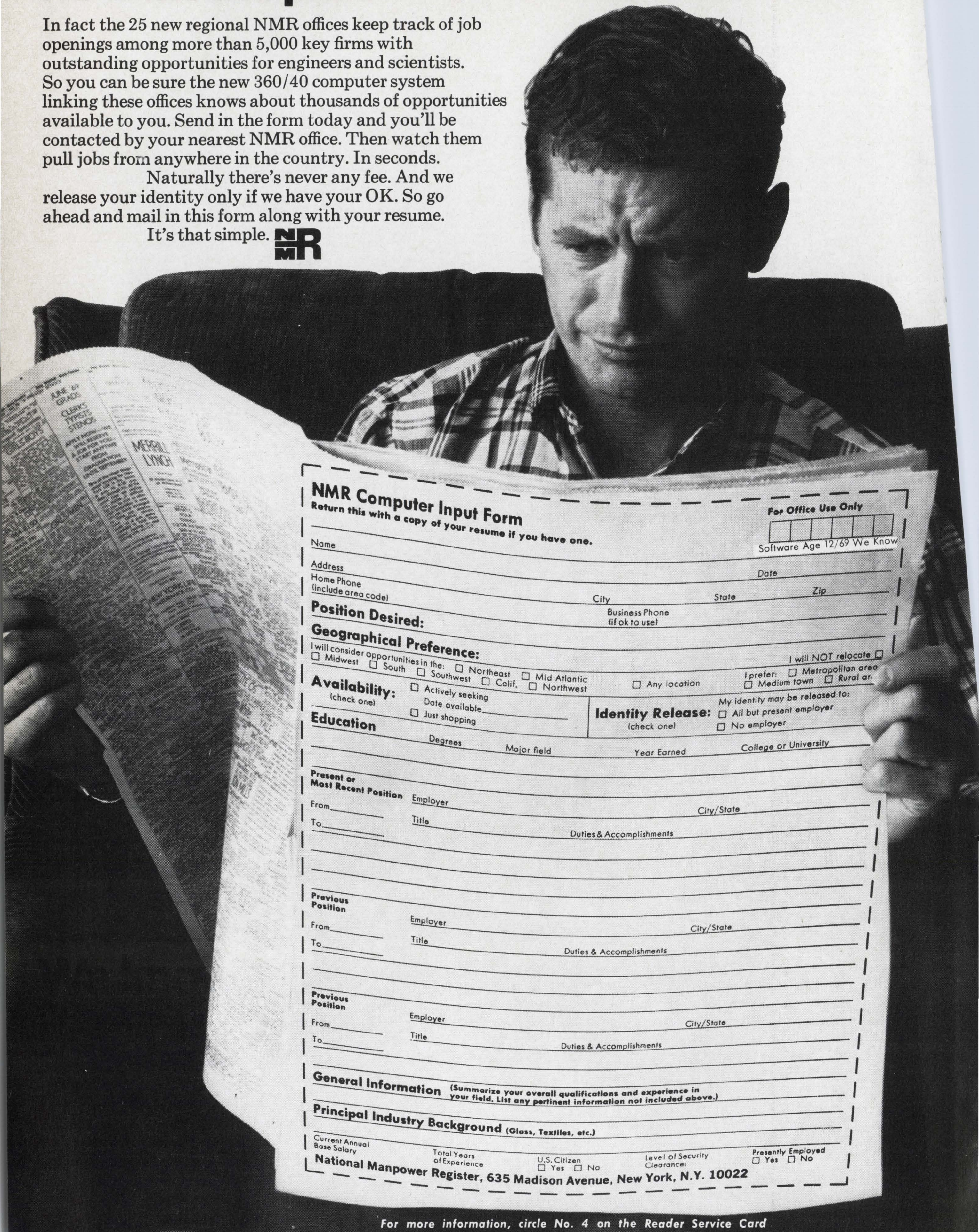
1—The price is based on time and materials. It is never a fixed price, even if a "quotation" has been made.
2—The rates for personnel services and the specification of the work to be done are not part of the agreement; they are not even incorporated by reference.
3—IBM invoices at its current rates, and may change them after notice (see Table II).
4—IBM determines which personnel to assign.
5—The user assumes responsibility for the supervision, management, and control of the work of the personnel IBM assigns.
6—IBM does not warrant the fitness of the work of its personnel in meeting the user's need.
7—IBM is not liable for failures to meet schedules.
8—IBM may modify the Agreement but the user may not (the user may terminate the entire Agreement).
9—IBM will ". . . instruct its personnel . . ." to treat the user's financial and statistical data as confidential, and does not bind itself to so treat such data.
10—IBM retains most rights to all original materials it produces in whole or in part under the Agreement, including ideas, concepts, data, discoveries, inventions, techniques, and the like.

We know where the jobs are and you don't. It's that simple.

In fact the 25 new regional NMR offices keep track of job openings among more than 5,000 key firms with outstanding opportunities for engineers and scientists. So you can be sure the new 360/40 computer system linking these offices knows about thousands of opportunities available to you. Send in the form today and you'll be contacted by your nearest NMR office. Then watch them pull jobs from anywhere in the country. In seconds.

Naturally there's never any fee. And we release your identity only if we have your OK. So go ahead and mail in this form along with your resume.

It's that simple. NMR



NMR Computer Input Form

Return this with a copy of your resume if you have one.

For Office Use Only

Software Age 12/69 We Know

Name

Address — Date

Home Phone (include area code) — City — State — Zip

Position Desired: — Business Phone (if ok to use)

Geographical Preference:

I will consider opportunities in the: ☐ Midwest ☐ South ☐ Southwest ☐ Northeast ☐ Calif. ☐ Mid Atlantic ☐ Northwest ☐ Any location

I will NOT relocate ☐

I prefer: ☐ Metropolitan area ☐ Medium town ☐ Rural ar.

Availability: (check one) ☐ Actively seeking — Date available — ☐ Just shopping

Identity Release: (check one) My identity may be released to: ☐ All but present employer ☐ No employer

Education — Degrees — Major field — Year Earned — College or University

Present or Most Recent Position — Employer — City/State

From — To — Title — Duties & Accomplishments

Previous Position — Employer — City/State

From — To — Title — Duties & Accomplishments

Previous Position — Employer — City/State

From — To — Title — Duties & Accomplishments

General Information (Summarize your overall qualifications and experience in your field. List any pertinent information not included above.)

Principal Industry Background (Glass, Textiles, etc.)

Current Annual Base Salary — Total Years of Experience — U.S. Citizen ☐ Yes ☐ No — Level of Security Clearance: — Presently Employed ☐ Yes ☐ No

National Manpower Register, 635 Madison Avenue, New York, N.Y. 10022

For more information, circle No. 4 on the Reader Service Card

Financial Currents

I.R.A. Systems, Inc., a U.S.M. Corporation affiliate with headquarters in Waltham and manufacturing facilities in Whitinsville, has changed its name to Sprias Systems, Inc. The name change was announced by Walter Anderson, President of the systems development firm, which manufactures the Spiras–65 stored program controller/processor, the IRAscope CRT database editing display, and related software and peripheral equipment.

* * *

Foto-Mem, Inc. of Natick, Massachusetts has acquired Wilkinson Computer Sciences, Inc. Foto-Mem now plans to sell the Wilkinson computer alone or in conjunction with Foto-Mem's FM-390 or RISAR Systems. With the Wilkinson computer, they can now deliver a complete "turn-key" installation and take full responsibility for all components of the system.

* * *

Data Bank Corporation, New York, N. Y. is a new company that has been formed by J. Paul Chin, Edward Mathews and John Kong. The company is entering the information storage and retrieval business utilizing a photo-optical random access mass memory, with multi-billion bit or multi-million document capacity. Using its own designed MARS© (Microfiche automatic Retrieval and Storage) System, the corporation will be offering the following services: Facility Management or Turnkey Operation, Time-Sharing Services, and System Consultation.

* * *

Datadial, Inc. of Smithtown, New York has purchased 51 percent of the stock of Associated Computer Management, Inc. of Rego Park, New York. Datadial is primarily engaged in facilities management, for general insurance agencies, permitting immediate access to Datadial's time sharing data processing facilities, and ACMI markets special programs developed for medical centers. The merger of ACMI complements Datadial's plans for expansion and growth in developing other proprietary programs aimed at specific professional and business groups.

* *

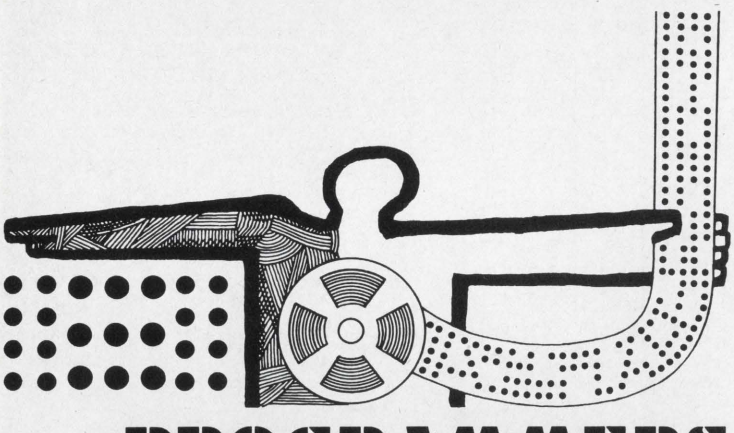
DATA 100 Corporation, New York, N. Y. has agreed, in principle, with Radale Electronics, Inc. for the acquisition of all the assets of Rodale Electronics, Garden City and Westbury, New York. Rodale Electronics currently manufactures sophisticated electronic equipment for military and commercial use, including: airborne computers, radar transponders, anti-submarine warfare and direction finder equipment, telemetering transmission equipment and radar altimeters. Data 100 Corporation, headquartered in Minneapolis, Minnesota, is a designer and manufacturer of computer terminal equipment and systems, which provide computer users with the capability of transferring information between centralized computer centers and remotely located offices and plants. Data 100 also manufactures metal cabinets for computers and computer peripheral equipment through a recently acquired subsidiary.

* * *

Digital Applications, Inc. New York, N. Y. (O.T.C.) has announced the acquisition of CDS Electronics, Inc. of Houston, Texas. CDS is engaged in the business of manufacturing electronic Systems based upon either their own design or for the detailed specifications of their customers. CDS facilities include the capability to complete the manufacture of printed circuit boards, to wire up complete modules, and to carry out the mechanical fabrication of electronic housings utilized in a complete range of instruments, units and data acquisition systems. The acquisition of CDS thereby give Digital Applications, Inc. the in-house capability of manufacturing its whole range of Data Acquisition and control systems.

* * *

Diversified Technologies, Inc., New York, N. Y. and Brogan Associates, Inc., Long Island, N. Y. have signed an



scientific and technical fields, for use in research and development, and in legal, general management, and public relations work.

* * *

Autocomp, Inc., Bethesda, Maryland and Autocode, Washington, D. C., have agreed in principle to the acquisition of Autocode by Autocomp. Autocomp specializes in the application of computer technology to the codification of municipal, county, state and national codes and statutes and offers a wide range of computer and other services to governmental units and private industry. The acquisition of Autocode is projected to give Autocomp the capability of providing a total range of computer and other specialized services to assist all levels of government in solving the urban crises. The Autocomp–Autocode personnel will represent a blending of talents in the areas of computer and photocomposition technology, government and law.

* * *

Tracor Computing Corporation (TCC) Austin, Texas and Insurance Information Exchange, Inc. of Dallas have announced that the two companies have reached an agreement in principle under which TCC will acquire the Dallas Computer software company. Insurance Information Exchange provides consulting, programming assistance, training and "software packages" to insurance companies while Tracor Computing Corporation offers a full range of computing capabilities, including hardware product development. TCC specializes in systems and facilities management services. under which they takes total responsibility for a customer's computing requirements.

* * *

University Computing Company, a multi-national computer service organization headquartered in Dallas, has acquired Computer Data Sciences Inc. of Cleveland, a affiliate of Curtis Noll Corporation. Computer Data Sciences was formed in Cleveland in late 1967 and has specialized in computer programs and operating systems, and the development of a cathode ray tube type-setting service for the printing and publishing industries. CDS will remain intact as a wholly-owned division of UCC's Applied Science Group. UCC International, Inc. has also organized a new subsidiary, University Computing Canada, Ltd., headquartered in Toronto. UCC operates a nationwide computer Utility Network, powered by computing centers of varying capacity in some 15 cities, and a similar network in Great Britain and western Europe.

FREE PROGRAM LISTINGS IN SOFTWARE AGE

(Procedures and Practices)

COST . . . . There is no charge for computer programs listed in SOFTWARE AGE.

PUBLISHED . . . . . The listings will be published in the first available monthly issue of the magazine.

All listings will be re-published in semi-annual summary issues of SOFTWARE AGE during the months of the S.J.C.C. and F.J.C.C.

BROKERAGE . . . . SOFTWARE AGE does not broker computer programs nor act as an agent. As an unbiased third party, SOFTWARE AGE is acting, solely, as a pipe-line of communications between buyers and sellers of computer programs.

EVALUATION . . . . SOFTWARE AGE does not compare or evaluate computer programs. The degree of seller on-site support and the extent of buyer capability virtually make the evaluation of any program meaningless.

REQUIREMENTS . . . A prerequisite for any listing is that it be documented or supported to the extent of being operable by the buyer.

ANONYMITY . . . . A cross-coded reference number will be assigned to any program whose originator wishes to remain anonymous. Buyer inquiries will be forwarded to the originator. If the originator still prefers anonymity, buyer will be notified by SOFTWARE AGE.

GROUPING . . . . . Listings will be grouped according to keyword, title, mainframe application and compatability, industry, etc. As listings grow, more categories and definitions will be added.

software age

2211 FORDEM AVENUE
MADISON, WISCONSIN 53701

FREE PROGRAM LISTINGS IN SOFTWARE AGE

Additional Description		Additional Listing Forms Required	

RETURN TO: PROGRAM LISTING EDITOR • SOFTWARE AGE • 2211 FORDEM AVE. • MADISON, WIS. 53701

Additional Description		Additional Listing Forms Required	

RETURN TO: PROGRAM LISTING EDITOR • SOFTWARE AGE • 2211 FORDEM AVE. • MADISON, WIS. 53701

the printing of special and regular reports. MMFILE provides file creation and maintenance capability. The MINI-MIS programs are part of SBC's program library under the CALL/360: BASIC System. (SBC offers time-sharing services in 35 cities across the nation through six computer centers.)

For more information, circle No. 10 on the Reader Service Card

* * *

Ancom Systems, Los Angeles, Calif., has developed a new General Ledger System to provide a flexible and convenient method of forecasting and measuring financial progress. The system integrates forecasts, budgets, and actual performance into an analytic system containing comprehensive reports to isolate an "out of control" situation requiring corrective action. The system offers the capability of comparing results to both forecast and budget. It is capable of measuring and reporting financial data at the multiple levels for multiple companies and it can consolidate the information at higher levels. As a result, analysis can be performed at the department, division or operating company level. In addition, 3 levels of consolidation above the operating company are available.

For more information, circle No. 11 on the Reader Service Card

Dialog Computing, Inc., Fairfie Conn. has recently released the seco in a projected series of Time/Shari oriented software packages. Dialo new interactive CPM/PERT Syste provides dramatic improvement in tot project management. This newly d veloped system will provide segmente data for evaluation during any stage (the project progress. New cost factor. personnel deployment and interir scheduling can be achieved throug the segmented evaluation process af fected with the Dialog CPM/PERT System.

Major technical features of the new system include: Data input editing before the data is entered into the calculating portion of the system; free-form input of data of variable field length; multiple starting and/or ending nodes ability; complete network error analysis including "loop" detection; and machine independent Fortran.

For more information, circle No. 12 on the Reader Service Card

* * *

Computer Interaction, Inc. of Great Neck, New York, has announced the availability of its new CRAM/360 proprietary program for compression, retrieval and maintenance of data. CRAM/360 is said to be capable of increasing external storage capacity in most major manufacturers' hardware by an average of 40%.

The system is supposed to provide the capability of retrieving and updating source data stored in a special direct access library. In addition to the compress and decompress techniques, a new fragmented data set concept is utilized for access and retrieval. The compress routine scans each input source record and eliminates all blank columns and only the valid characters remaining are compressed.

The decompress routines restore each compressed image to the original 80 columns. The CRAM/360 system was designed for use in the following application areas:

1. name and address files (mailing lists)
2. stock inventory systems
3. source language files containing programs written in Assembler, Cobol, Fortran, RPG, and PL/I.

For more information, circle No. 13 on the Reader Service Card

* * *

Communication Dynamics Systems, Inc., Westchester, Illinois, has developed what is said to be a "totally new approach" to Cobol Cross-Referencing problems. The mini-program, REFBACK, uses one card to call a subprogram off the core image library at compilation time. The subprogram generates a Cobol Cross-Reference list of all

I.A.C.P.

A professional organization to represent the interests of programmers throughout the world has recently been formed. The Interntional Association of Computer Programmers, Inc., or I.A.C.P., a non-profit organization is presently encouraging the formation of local chapters by active, affiliate and student members throughout the United States.

The schedule of the new association calls for

- a monthly magazine containing articles and departments of interest to the computer programmer
- state, regional, national and international meetings of the membership
- educational seminars, lectures and exhibits
- publication of educational materials
- and many more programs of interest to the members of the programming community.

If you are interested in further information on how you may form or join a local chapter in your community please write to Richard Lynch, % Software Age Magazine, Box 2076, Madison, Wisconsin, 53701.

data names and procedure names giving all page and line references used by data and procedure division names. The Cross-Reference list indicates any fields which are used as subscripts, list indicates any fields which are used as subscripts, qualifiers, or data files.

Previous to the introduction of this unique mini-program it was impossible to generate a Cross-Reference list from an IBM Cobol compilation.

For more information, circle No. 14 on the Reader Service Card

* * *

MIRACL/CPG, a computer software product directed toward reducing turnaround time for writing and testing programs has been announced by Republic Software Products, Inc., a subsidiary of Republic Systems, Inc. of East Orange, N.J.

MIRACL/CPG (Cobol Program Generator) generates complete, debugged Cobol programs with far less effort than through the use of Cobol. The MIRACL system improves on the efficiency of programming languages by eliminating the repetitive and time-consuming aspects of detailed coding. As many as 30 or more Cobol statements are replaced by one MIRACL statement. This allows programmers to spend a higher proportion of their time on program and systems logic instead of clerical details. An additional feature makes it possible for non-programmers to use MIRACL and generate reports.

For more information, circle No. 15 on the Reader Service Card

* * *

Information Science Incorporated of New York City, New York has developed a new programming tool for the creation and maintenance of computer files that includes complete edit and update capabilities. Termed the General Maintenance System (GMS) the new concept is a general purpose system that creates a complete range of edit and update programs, including file creation, transaction editing, file maintenance and optional audit–trail and/or error routines. The programs are created for each file by the GMS "generator" and all are written in COBOL, with provision for incorporation of unique individual user routines. Minimum machine configuration specifications for installation are: System/360–32K DOS or OS; Reader Punch; Printer; Disk Drive for system residence and one additional direct access device or four tape drives; or equivalent COBOL support configuration.

For more information, circle No. 16 on the Reader Service Card

CONFERENCE COUNTDOWN

DECEMBER

18–20 Third International Symposium on Computer and Information Science, Bal Harbour, Miami Beach, Fla. Contact: Dr. Julius Tou, Dept. of Electrical Engineering, University of Florida, Gainesville, Fla. 32601.

27–28 Annual Meeting of the Association for Symbolic Logic, New York, N. Y. Contact: Dr. Barwise, Dept. of Mathematics, Yale University, New Haven, Conn. 06520.

JANUARY, 1970

12–16 Data Transmission Facilities exhibition, London, England. Contact: U.S. Dept. of Commerce (BIC-942), Washington, D.C.

14–16 1970 International Conference on System Sciences (IEEE), Honolulu, Hawaii. Contact: Dr. Richard Jones, 2565 The Mall, University of Hawaii, Honolulu, Hawaii 96822.

20–22 Computer Software and Peripherals Show and Conference, Eastern Region, New York, N.Y. Contact: Show World, Inc., 37 West 39th St., New York, N.Y. 10018.

FEBRUARY

16–19 Annual Computer-Aided Circuit Analysis and Design Institute, Tampa, Fla. Contact: Dr. George Zobrist, Department of Electrical Engineering, University of South Florida, Tampa, Fla. 33620.

17–19 Computer Software and Peripherals Show and Conference, Midwestern Region, Chicago, Ill. Contact: Show World, Inc. 37 West 39th St., New York, N. Y. 10018.

23–25 Data Processing Supplies Association, Winter General Meeting, New Orleans, La. Contact: DPSA, 1116 Summer St., P.O. Box 1333, Stamford, Conn. 06904.

27–28 National Association of Computer Assisted Analysts (NACAA) Multiposium, Los Angeles, Calif. Contact: NACAA, Multiposium, Mary Bragg, P.O. Box 2802, Fullerton, Calif. 92633.

MARCH

23–25 INFO–EXPO–70, Washington, D.C. Contact: Paul Zurkowski, Information Industry Association, 1025 15th St. N.W., Washington, D.C. 20005.

23–26 IEEE International Convention and Exhibition, New York, N.Y. Contact: IEEE, 345 E. 47th St., New York, N.Y. 10017.

APRIL

7– 9 Computer Software and Peripherals Show and Conference, Western Region, Los Angeles, Calif. Contact: Show World, Inc., 37 West 39th St., New York, N.Y. 10018.

Some of the other topics covered in this book are: Content Analysis in Political Research; International Pattern and National Profile Delineation; Numerical Simulation of the Earth's Atmosphere; Artificial Intelligence of International Relations; Pictures, Computers, and Input-Out; and the Transition Toward More Sophisticated Procedures.

The physical sciences have long had the advantage of heavy and initial use of computers in war related research but it seems likely that computers will ultimately be more important to the behavior sciences than the physical sciences. The reason for this opinion is that man is the most complex of mechanisms and the computer is the only device capable of providing any worthwhile simulation or analysis, except of course by man himself.

A GUIDE FOR SOFTWARE DOCUMENTATION. By Dorothy Walsh. Inter-Act Corporation, 437 Madison Avenue, New York, N. Y. 10022. 157 pages. 1969. $16.25.

Good documentation is usually one of the biggest headaches for programming managers since most programmers feel documentation to be an insult to their intelligence. This aversion towards documenting quite often just covers up the fact that program documentation is difficult, time consuming, and not as thrilling as actual programming.

Aside from providing the programmer with clerical and typing help, the manager who wants good documentation should also provide a documentation guide. That's where this book can be useful; it offers the busy manager a chance to adopt a standard guide at little expense and effort on his part.

Fourteen guides are provided for almost all types of programming situations including conversational and non-conversational, internal logic, library users, operations, and operating system guides. The book provides the guides, with little discussion on their use, but the information seems sufficient for successful documentation. The book does suffer from the lack of any completed examples, but the material is clear enough that this omission is not fatal. Rather than each computer installation researching and designing its own software guide it will probably be more efficient for the installations to start with a guide, like this one, and proceed to modify, add, and delete sections as needed.

MAN AND THE COMPUTER: Technology as an Agent of Social Change. By John Diebold. Frederick A. Praeger, 111 Fourth Avenue, New York, N. Y. 10003. 157 pages. 1969. $5.95.

John Diebold obviously belongs to the optimistic school of technology—that is, technology can solve all man's problems. Such a philosophy becomes evident as one reads through the book because such statements appear as: "We today possess the means for achieving virtually any ends we wish, whether it be to travel the galaxy, mine the ocean, replace the human body, or educate, house, and care for the world's population." Unfortunately these optimistic promises do not state explicitly *how* we are to solve the political problems of population control, resource distribution, or how we are to prevent atomic, biological or environmental destruction.

Since John Diebold is a very successful manager (president and founder of the Diebold Group, Inc., an international management consulting company, and chairman of John Diebold Inc., a management and investment company) one would rather expect him to take a manager's point of view, and he does in this collection of speeches. Diebold rightly condemns organized labor for demanding higher wages and supporting featherbedding jobs, but he completely ignores industry's disinterest in employee obsolescence. He also states that labor should be demanding educational facilities for workers caught in a technological squeeze of old work skills and new methods, and while this is a pertinent point, both labor and industry have not been too eager to help with the difficult task of retraining workers. Rather, industry usually prefers to simply lay off obsolete workers with obsolete skills and contribute to unemployment compensation.

Diebold goes on to suggest that a computerized job bank which matches employee qualifications to employer requests will help alleviate unemployment, but this suggestion ignores that fact that an uneducated laborer can not be matched to any available technological position.

While this book will make many highly placed technocrats pleased with their future (Diebold does a good job of describing technological possibilities), the book does not give the answers to the political problems of implementing the solution. By far the best chapter is on Education. Diebold concludes that both the population and information explosion combined with a rapidly changing technology will force major changes in education. Diebold sees all this as an impetus to study how people learn and the introduction of new teaching devices such as computers. Diebold perceptively notes that because of the rapid change in technology future education of a child must address itself to imparting an ability to learn and concern itself less with the content of learning.

Other topics covered are the impact of science and technology, international disparities (European only), and the training of managers. The subtitle of the book more correctly describes the content of this book than the title.

check your interests here

Fill in the confidential inquiry form on the other side of this sheet. This form provides all the information advertisers require to screen applicants. If further information is desired, you will hear from the advertiser direct. Then, check below the boxes of those companies to which you want copies of your form sent. Mail to SOFTWARE AGE, P.O. Box 2076, 2211 Fordem Avenue, Madison, Wisconsin 53701. (Please do not send us your own resume. We will only process this form. A new form must be filled out for each issue in which you are answering ads.)

☐ **I do not now receive S/A. Please send me information on how I may receive a FREE subscription.**

....
Name

....
Street Address

....
City State Zip Code

software age

P. O. Box 2076
2211 Fordem Avenue
Madison, Wisconsin 53701

#10