DATAMATION
94 South Los Robles Avenue • Pasadena, California 91101 • (213) 681-8486
The Magazine of Automatic Information Handling

October 7, 1970

Mr. Bob Bemer
General Electric Company
13430 N. Black Canyon Highway
Phoenix, Arizona 85029

Dear Bob:

We are planning a new feature consisting of profiles of various of the prominent or influential (where the two do not coincide) people in the computer industry and I have been assigned to "do" you.

A certain amount of interviewing will be necessary, of course--we don't intend to do dry, statistical pieces--but time can be saved on the bread and butter questions if you will fill out the enclosed questionnaire and send it back to me so that they're out of the way when we meet. A resume, if available, and a photo also would be helpful.

I'll try to make it all as painless as possible and trust that the profile will benefit both the magazine and you. Your co-operation will be appreciated.

Cordially,

Aubrey

Aubrey Dahl
Associate Editor

AD:agw

Bob Bemer
By all means! Pls let me know how it comes out.
Rec'd
70 OCT 30

Thompson Division of Technical Publishing Company

GREENWICH 06830
35 Mason Street
(203) 661-5400

CHICAGO 60606
205 West Wacker Drive
(312) FInancial 6-1026

MANCHESTER, N.H. 03104
112 West Haven Road
(603) NAtional 5-9498

PASADENA 91101
94 S. Los Robles Ave.
(213) 795-9721
from L.A. (213) 681-8486

PALO ALTO 94306
260 Sheridan Ave.
(415) 321-1440

GENERAL ELECTRIC

Information Systems

Advanced Development and Resources Planning Division

DIAL COMM 8*433 2569 DATE 1969 November 24 MAIL ZONE M2

DEPT. EMIO

ADDRESS • COPIES •

SUBJECT • TRIP REPORT - Second NATO Conference on Software Engineering

TO: Distribution

FROM: R. W. Bemer

1. APOLLO 11 SOFTWARE

This presentation by Joel Aron, IBM Federal Systems Division, was most interesting, loaded with information, of which the highlights follow:

- Ten years ago Vanguard required part of a 704. Apollo 11 used six 360/75s for development and operation and a 360/50 for programming support. Even so, they found they weren't providing enough systems for usage and went to 11 machine hours per programmer month (as opposed to 7 or 8 hours for other IBM FSD work).

- The number of IBM personnel at Houston varied over project life from 300 to 850 people. Of these, a maximum of 300 were programmers; 250 writing applications (primarily in FORTRAN) and 50 building a realtime operating system (primarily in assembly language). This operating system was based upon OS360, yet turned out to be different and incompatible due to schedule requirements (APL please take note). These programmers were supported by less than 50 operators and technicians.

- The major aspect of software design was that output was tailored to human usage via console. Modules assigned to individual programmers ranged from 400 to 1000 machine language instructions. The system was modeled in GPSS. A simulator was built which could operate either with mission controls as black boxes or by running the actual program itself. Thus the program simulated the hardware, the software and the realtime application.

- Software production was done in batch mode, as was debugging, at 1 or 2 shots per day. They found they could adjust to this schedule (but Aron's paper showed very high production costs, almost $50 per instruction). Control was modeled on a guideline for hardware and software projects, named either Air Force 375-1 or NASA 500-1. (I haven't seen this document.) They quit using PERT in the middle of the project; they were too busy to provide inputs, and they thought it did not hurt to drop it.

2. PROGRAM CORRECTNESS

Of the four major groups, I had the bad luck to be stuck chairing the one that discussed this. Nothing else showed so well the disparity between the "computer science--university type" and the "practical software production man". Attachments 1 and 2 bear on this matter, and both were unpopular with the scientists. Dijkstra was the most vociferous, saying that "program testing can be used to show the existence, but not absence, of bugs". He is exploring new ways, including the metaphysical, to solve the latter problem. We had some worthwhile quotes:

Schorr (IBM) - "Apollo 11 acceptance testing took about 2 months, and it was at least 30 days before anything would even start to run in realtime. Bugs were taken out of the software up until the day before launch."

Aron (IBM) - "Testing and integration of Apollo 11 software took 30% or more of the total time, and from 30 to 50% of the cost of the project."

Needham (Cambridge University) - "There are very few bugs in our operating system that weren't put in the last two weeks."

Someone christened the Dijkstra approach "The Power of Positive Programming", but this had to be explained. Apparently Norman Vincent Peale is not well known in Rome.

3. PRODUCTION METHODS

There was general agreement that production was in a mess, but the managers of the messes would not believe the techniques proposed because they had not been proven in sufficiently large "commercial grade" products.

On the pessimistic side, Scalzi said that the OS 360 operating system part was targeted at 300K instructions and went to over a million for only the same capability. Hopkins of IBM quoted production of 600 instructions per man year. I said that from what I knew of their production rates (.2/hour), that meant they were all working 1 1/2 shifts a day.

On the optimistic side, there was a great deal of support for IBM's APL. Sharp of Toronto said they had done a COBOL compiler in 6 man months using APL. Perlis said that "our problem is that finite skulls require concise representations. APL programmers carry assemblers and parsers in their heads!"

4. PROPOSAL FOR NATO SOFTWARE INSTITUTE

This has been in the mill for more than a year, and may be thought to be similar to CERN. In general the Europeans are more skittish about it than the Americans, which seems strange to me. Only a few see it as a major opportunity to bring fragmented European resources together to get some money out of software, now that the chance to do it in hardware has passed.

Presently it is envisioned as a physical institute, and Germany has even let it leak to the press that Munich is under strong consideration. Although there is more sentiment now to have it an umbrella institute (Canada and others would be happier with it this way), this would require reworking the papers already in the slow mills of official governments.

Nothing firm came out of the discussions, as the attendees were anyway not the official spokesmen for their governments.

5. MISCELLANEOUS

About 3 inches of papers were prepared for the meeting, but most of these are just as well read in the condensed report which will appear. There was one of particular interest to me, a paper on zero-time instrumentation of the UNIVAC 1108 by means of a second system tapped into it. The paper abounds in details, and will show just exactly what goes on as FORTRAN compiles at 5300 statements per minute. Available upon request on a circulating basis. The big charts are too difficult to reproduce.

Herb Schorr is apparently quite well up in the IBM hierarchy. He and Marty Hopkins made an interesting remark on the usage of programming languages, saying that the fasting growing usages were, in order--RPG, COBOL and then the others. This admits that PL/I is not gaining ground in competition with COBOL.

Chris Strachey doesn't like the COBOL spelled-out instructions. He shows that "DIVIDE CAKE INTO 3" gets you 3/CAKE, rather than CAKE/3.

R Bemer

po

Attachments: Software Engineering and Computer Science
Needham and Aron

A Note on Implementation Correctness
Bemer

ATTACHMENT 1

R. Needham
J. Aron

Software Engineering and Computer Science

Computer Science aims at defining general principles underlying the design and application of software systems. It regards elegance and consistency highly, and tends to ignore small and awkward corners in its subject matter. Thus the computer scientist is tempted to analyse and treat that fraction of his problem which is amenable - rather in the same way as the differential equation expert treats the vanishing subset of equations which are analytically soluble.

The software engineer wants to make something which works, where working includes satisfying commitments of function, cost, delivery and robustness. Elegance and consistency come a bad second. It must be easy to change the system in ways which are not predictable or even reasonable - e.g. in response to management directives. Theory presently cannot keep up with this kind of thing, any more than they can with the sheer size and complexity of large software systems.

A theory must be independent of implementation. In practice large systems implementation is influenced by many factors: available personnel, management structure, and so on. If theoretical work cannot adapt to this it can at best do no more than give helpful hints, while at worst it is irrelevant or unpractical.

Much theoretical work appears to be invalid because it ignores parameters that exist in practice. Thus a system which depends on failure-free operation is unrealistic. In the sense that medical diagnosis is empirical rather than scientific - because the observable parameters are an insufficient subset of the operable parameters - so will computer science be empirical until better methods are available for describing the structure and behaviour of actual systems.

Theoretical work becomes impractical when it ascribes (mathematically) analytical behaviour to software - or at least to the software design, development, and test process. Present day software is the sum of individual activities operating in discrete and usually asynchronous time steps with a presumed common goal. The time steps depend on such factors as individual productivity, computer availability, predecessor activities, conflicting demands or resources, and quality of task specification. The common goal is "presumed" because semantics cause individuals to interpret the goal and its constituent tasks according to their own background. To contribute further to the random discrete behaviour of the implementation process (looked on as a system) is the necessary iterative nature of the goal itself.

Software engineering managers learn how to make decisions under conditions of uncertainty. Where possible, they reduce the uncertainty by applying theory or standards. In no case do they assume that a course of action will go perfectly and not require redirection. Theorists have not learned to cope with this randomly discrete set of events in an uncertain environment. Therefore, their impact on the engineers is minimal.

If theorists could demonstrate the application-independent value of their conclusions and could guarantee that they would work reliably on another job, engineers might overcome their conservative attitude toward new techniques and try them out. Unfortunately, an adequate demonstration requires two or more different large-scale implementations. This is usually beyond the scope of the theorists' interests or resources.

Pending major theoretical advances, software engineering should concentrate on the development of, and the exchange of experience about, practical tools such as:

diagnostic aids
protected testing facilities
automatic or semi-automatic fallback
aids to continuity during development
etc.

We believe that there is much to be gained by discussion and development in these areas.

27th October, 1969

A NOTE ON IMPLEMENTATION CORRECTNESS

R. W. BEMER

I reject the interpretation of 100% "mathematical" or "logical" correctness for software engineering purposes, for reasons of statistical frequency of exercise and the program-data interaction:

1. If a general purpose computer is programmed to do nothing but Fortran compilation, which itself never exercises a logic flaw in the instruction set, then the combined system may be said (if this is the only flaw in the machine or program) to be correct.

2. It buys little to move a program from 99.8% to 99.9% (statistically) correct operation, if the data is 80% correct. Or is the data problem considered to be outside the realm of software engineering?

It is not known, nor have the computer scientists provided us with the insight, how to simulate and test a large multi-access system by means of another computer program that exhibits the realtime properties of:

1. Any randomly possible selection from the U.S.A. communication system.

2. The U.S.A. population making other demands upon that system.

3. An unpredictable user population, either in loading or arbitrary usage.

In short, correctness to the software engineer means that a system should do the "proper" thing rather than do exactly the actions that were specified with such imprecise knowledge.

I prefer the following interpretations of correctness:

1. Design correctness

   - Efficient utilization of production resources
   - Efficient utilization of system resources during running
   - Maintainable and reliable
   - Constructible
   - Flexible (for change and added function)

2. Implementation correctness

   - Matches the specifications
   - Solves the problem envisioned
   - Free from malfunction
   - Free from hang-up or locking

32 WENT TO FRITZ FOR DISTRIBUTION

1968 August 1

TO: ACM Standards Committee

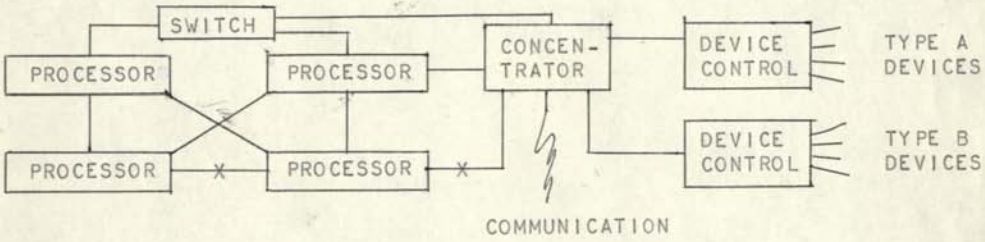
FROM: R. W. Bemer

SUBJECT: Personal Comments on the X3 Meeting, BEMA, New York City, 68 August 1.

1) I/O INTERFACES

The ad hoc Committee on I/O interfaces had made a formal request to become a standing committee of X3. Their latest target area was given as "The I/O interface between the Basic Processor and the Control Electronics"!

After the formal presentation I said that I was still unable to comprehend just what the ad hoc Group meant to do after many months of studying their documents, so would Mr. Poorte kindly place an X on the line in the following diagram where the standard interface is intended?



The X are as added by Mr. Poorte.

It was not coincidental that the organization above has a partial resemblance to Illiac IV and CDC equipment. I asked if CDC and UNIVAC processors should be capable of direct coupling and Poorte answered yes. I then asked if the interface would be independent of system characteristics, that is, the lines would have functions as assigned by the user. Poorte answered no. I asked that this be made a part of the record. I then drew attention, for the benefit of DOD and NBS, to the fact that this was definitely opposite to the government desires for alternate peripheral equipment. One should standardize with care to not destroy the capability for innovation, and always from the outside in, as we had taken interchangeable media and formats first. Attention was drawn to plug-compatible devices in existence, such as tape drives from Midwest, Texas Instruments, Potter, et al.

Jerry Haddad, IBM Vice President for Engineering, Programming and Technology, then stated that this was a most important area, and that if a suitable standard could be achieved for I/O interfaces it would equal in importance all the other work that X3 has done. However, he cautioned that even within IBM, where one had a more rigorous leverage to control than within a general standardizing body, success had not really been achieved in the I/O interface area. One should never underestimate the amount of work required. He then seconded me on not understanding just what the committee proposed, and why couldn't they do it unless they were a standing committee?

The reply was that they could not get the necessary support while in an ad hoc status. Following a vote of 6 yes, 8 no and 3 abstentions on the motion for standing status, Haddad proposed a motion that the ad hoc Committee Chairman not be constrained in forming subgroups to carry out the (still) preliminary work. Passed unanimously.

2) MAGNETIC TAPE LABELLING

In the very last stages of passing this standard up to USASI, a substantive flaw has been found by X3.2.5, which is now in the process of reworking the standard to eliminate it. Inasmuch as the ECMA standard and the ISO/TC97 Draft Recommendation are the same in this area, X3 will inform them immediately of this fact.

3) SHIPPING MAGNETIC MEDIA BY AIR

The attachment is of general interest. Note that the Air Transport Association invites recommendations.

4) FLOWCHART STANDARD

A last minute holdup on the combined symbols and conventions standard. Following the meeting of ISO/TC97/SC7 in Amsterdam on June 7, it appeared that four subsections of the standard might be in conflict with the ISO draft. The chairman of X3.6 wrote X3 suggesting an editorial change to remove these sections from the standard about to be published.

I moved that, to the contrary, they should be removed to an Appendix, where the problem would be described. The body of the standard would still contain the section headings, but have only references to the Appendix.

Clippinger called this proposal hasty and immature. However, a phone call to the X3.6 chairman revealed that they would be delighted with this arrangement but had not hoped to get it. (COBOL RPM precedent).

The sections so treated are:

4.5	Symbol Identification
4.6	Symbol Cross Reference
4.7.2	Cross Referencing Connectors
4.8.3	Cross Referencing of Striped Symbol and Detailed Representation

These sections should probably be used as they are until a definite reconciliation with the ISO is obtained.

5) PUBLISHING OF TUTORIAL PAPER ON RELATIONSHIP OF EBCDIC TO USASCII-8

This is Document X3.2/724, which relates the 256 positions of EBCDIC to the 256 positions of USASCII-8 (even though only 128 are now assigned) by the bridge of the 256 positions of the extended Hollerith card code (all combinations of which are now fixed, except that a certain portion of the relationship is in question internationally).

The vote was 13-5-1 abstention. G.E. voted NO because of its official position that "Relationships of USASCII to non-standard codes are pertinent only if the manufacturers that presently have such codes clearly support USASCII and demonstrate how such a relationship will accelerate the industry to the proper standard (USASCII)."

However, the point might have been stretched by arguing that although IBM does not support USASCII with much enthusiasm, this document may well cause such acceleration by defining clearly the bridge mechanism. Apparently IBM feels this latter point to be true, for IBM also voted NO.

Copies of this document are available from X3.2 upon request.

6) OCR AND USASCII KEYBOARDS

The chairman of X4 reported that the balloting on these drafts had been extended until October, and that more votes and comments had been received from X3 than from X4! He also said that ISO/TC95/SC14 had adopted the TC97/SC2 recommendations and were proceeding on a single coded keyboard with the "QWERTY" arrangement.

A X3/Systems Advisory Committee motion was passed unanimously as:

"1) X3 acknowledges that significant data processing considerations affect keyboard standards and conversely, keyboard standards can affect a data processing system.

2) The Chairman is instructed to advise the X4 Chairman that X3 members are being encouraged to add data processing capabilities to its membership and representation to X4.

3) X3.2 is appointed the responsibility for liaison activity with X4A9."

The main problem in this area lies in the fact that BEMA has separated the X3 (Computers and Information Processing) and X4 (Office Machines) for administrative purposes in such a way that coordination is difficult.

R. W. Bemer

AIR TRANSPORT ASSOCIATION
of America
1000 CONNECTICUT AVENUE, N.W. • WASHINGTON, D.C. 20036 • TELEPHONE 296-5800

July 22, 1968

Mr. James L. Smith
Chairman X3.2.1
I B M Corporation
P. O. Box 390
Dept. B 18, Building 706
Poughkeepsie, New York 12602

Dear Jim:

This letter will provide in writing the information I passed to you by telephone today. As I indicated, we have had an inquiry from an airline regarding the past experiences of the airlines regarding loss of data on magnetic tapes shipped via airline aircraft. We find, on investigation, that one or two airlines have been advised by shippers that they believed the airline might have been involved in the loss of data on magnetic tapes. Instances are scattered and probably, in total, do not represent more than two or three for the entire airline industry so far as we can determine as a result of our inquires.

It would have been rather convenient for me to forget the matter at that point, however, in an effort to be somewhat more thorough, I checked with our tariff publishers and found Tariff No. 855 which reads as follows:

"Magnetized material will be accepted only when:

(a) Devices, such as magnetrons and light meters, have been packed so that the polarities of the individual units oppose one another.

(b) Permanent magnets, where possible, have keeper bars installed.

(c) Outside packages have been plainly marked "MAGNETIZED MATERIAL".

(d) The provisions of Rule 9(f) in Section I can be complied with. In the case of highly magnetized articles such as magnetrons, suitable shielding may be necessary within the package in order to reduce the external field strength to acceptable stowage limits.

Shipper's Certification is not required."

I then tried to determine what magnetic field might be required to "damage" a magnetic tape. Note when I say damage, I am not

referring to the field required to erase the tape -- with essentially 100% sureity, but the field required to modify perhaps a portion of the bits on the nearest side of a magnetic tape placed near to a magnetic field. At this point, information becomes hard to find. I asked Aeronautical Radio (ARINC), an airline industry R & D organization, to run a quick check for me. They took a mag tape (in its normal plastic container) and placed it on top of a 1 megawatt airborne radar package megnetron. The tape was completely erased. A 1 Megawatt megnetron is by no means the largest that is shipped by air.

Interestingly enough, if a shipper of mag tape were to mark the package "Magnetic Material" or "Magnetic Tape" it is possible, even likely, that some very conscientious airline air cargo man would see that it was placed with the other "Magnetic Material" -- which certainly aids "Murphy's Law", which says (unlike the law of probability), "If it can happen, it will happen".

Computer tapes could be packaged in high permeability steel shipping cartons (not unlike those used for shipping film), which might weigh twice to three times the present plastic cases, but would provide a significant increase in protection to the mag tapes.

Obviously, it is also feasible (perhaps desirable) to add something to the Packaging Notes of the tariff of the following general nature:

> "Data Processing tapes which might be damaged by strong magnetic fields permitted by Packaging Note 855 of this Tariff must be clearly and boldly labelled by the shipper to indicate any special handling instructions or distance separation which may be required."

Similar tariff provisions already exist in the case of undeveloped film, which reads as follows:

> "Undeveloped film which may be affected by radiation to the level permitted in Packaging Note 700 of this Tariff must be clearly and boldly labelled by the shipper to indicate any special handling instruction or distances separation which may be required."

Before taking any further steps with the tariff publishers, or otherwise, we invite the recommendations of your committee, which would appear to include the best cross section of authority on data processing tapes. Hopefully, any recommendations you might provide could include some reasonable amount of evidence (perhaps including a few tests with a crated magnetron of various sizes) so that justification would exist for labeling shipments beyond those existing today.

Sincerely,

Frank C White

Frank C. White
Member X3, Representing
Airline Users

cc: Charles Philips - Chairman X3

GENERAL ELECTRIC

Information Systems

Information Systems Equipment Division

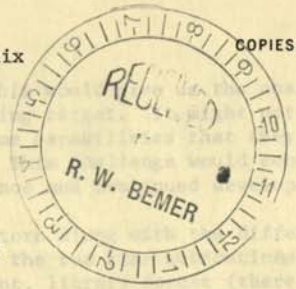
DIAL COMM 8•433 2178 DATE May 5, 1969 MAIL ZONE C-73

DEPT.• Large Systems Department

ADDRESS• Deer Valley Park Plant - Phoenix

SUBJECT•

COPIES• C. W. Dix
R. F. Stevens
L. I. Wilkinson
G. F. Woodward



Mr. R. W. Bemer, Manager
Systems & Software Engineering
Integration

Your suggestion to have a COBOL compiler operating on the 600 and producing code for the 360 is an interesting one. As I have mentioned to you earlier, this would not be an easy task and would certainly be an expensive one.

Obviously, the back end of the 600 COBOL compiler would have to be rewritten to produce 360 code. The analyzers and generators would have to be redesigned and rewritten to reflect the new target machine.

Not so obvious, perhaps, is the amount of redesign and rework that would be required in the Transformer or front end of the compiler. The Transformer is not as independent of the target machine as might be believed. The Pseudo Op Build, Storage Allocator, and File Allocator, for example, are Transformer runs that were designed and written to reflect the 600 operating structure and environment. They, too, would have to be completely redone to produce the 360 type of output.

IBM has special features in their 360 COBOL. Since IBM must continue to provide for their present users, I cannot envision their dropping "super standard" features in their USASI implementation. The Indexed Sequential feature is an example of an IBM special which their COBOL compilers will most certainly continue to accommodate. The number and nature of these implementor specials is not known, but their effect on a 600 compiler would be major. The present internal language would need to be modified and extended and the appropriate compiler modules changed.

With some tremendous effort, we could produce a compiler that matched IBM's as far as features supported when our implementation started and which produced code comparable to that produced by 360 COBOL. IBM would continue to make modifications that change the 360 generated code. These modifications would be done to improve object code, implement new language features, establish new

GENERAL ELECTRIC

run time interfaces, etc. This would give us the challenge of trying to keep up with a moving target. It might not be possible to give the 360 users the same capabilities that they would have from IBM's latest releases. This challenge would continue to be present through the maintenance and continued development phase.

When one considers these factors along with the differences between functions performed by the run time subroutines, file systems, operating environment, library format (there is no standard for data representation on the COPY library), and the need for supporting utility routines, your development cost estimate of $250,000-$500,000 is extremely low. The 360 and 600 computer costs and the quality assurance costs alone would approach these figures.

Money, of course, is just one of the resource considerations. The 600 COBOL project, where most of the COBOL compiler talent is localized, is presently committed through 1970 for its USASI implementation efforts.

Your motto has been "Don't think small!" This would not be a small undertaking. If the potential rewards warrent the costs and risks involved, then we should start project definition and planning. I suggest that step one be a raid on IBM's 360 COBOL compiler group.

G. B. Krekeler

G. B. Krekeler, Manager
Language Systems Engineering

GBK/er

8*223-1042 May 14, 1969

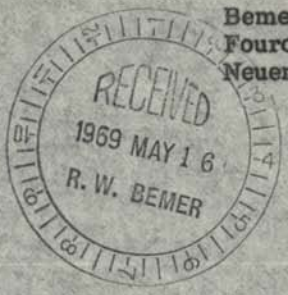
B&MPO

Building 28-EW

Bemer, R. W.
Fourot, J. C.
Neuenschwander, J. W.

COBOL Compilation Centers

Ref.: R. W. Bemer's Letters Dated January 7, 1969 and April 28, 1969



Mr. J. H. Sweeney
OFFICE

R. W. Bemer's suggestion in the referenced memorandum is to provide COBOL compilation centers which would be used by IBM 360 users for the purpose of translating original programs (source code) into object code ready for debugging and running. Discussions with Bob Bemer and (briefly) with Leroy Ellison and John Weil indicate that the idea is a reasonable one and, further, is technically feasible. Estimates for the effort range in the order of $500,000 plus or minus 20%.

An effort is presently underway within the Government to authorize the establishment of centralized compilation centers. We are in a position because of our COBOL compiler on the GE-600 to take advantage of this business opportunity. The opportunity initially would be to provide compilation centers for Government installations which would merely do the compiling. It is anticipated that debugging must be done on the user's equipment since the compiler center will have no way of knowing a precise configuration or options on the user center. Aside from the immediate benefits of providing a useful additional service which would be revenue producing, there are some very specific implications which should be considered regarding APL and unbundling.

The type of service to be offered here should be the same as that which we would plan to offer small computer users within the APL environment. Specifically, the small computer user should be provided with a terminal which plugs into a larger resource on which is available all of the various types of compilers which will result in object code to run on his machine. His basic machine would be supplied without such compilers; therefore, in the area of unbundling, he would not pay for software but would rather pay for service whenever he is using the compiler center. Of major and specific interest to the small user is the fact that he is the man that needs the greatest compiling and language assistance in order to take proper advantage of his installation, since he cannot afford the elaborate programming staffs that a larger organization can. By using this technique, he has the advantage of the more sophisticated compilers and only pays for them as necessary.

This idea has been incorporated into the most recent 700 Line Plan and was described in an earlier memo by the undersigned to David Booth. Since this compilation center project would fall most naturally in the realm of a service offering, John Neuenschwander will work with George Feeney of IND to further pursue this opportunity.

Rolf Kates

/vk

GENERAL ELECTRIC

Information Systems
Marketing and Product Planning Operation

DIAL COMM • 8*223-1070 DATE • May 20, 1969

DEPT. • B&MPO

ADDRESS • Bridgeport, Building 28 EW

SUBJECT •

COPIES • R. W. Bemer
J. C. Fourot
R. Kates
J. H. Sweeney

Mr. Z. Quastler
Manager - Marketing
IND
7735 Old Georgetown Road
Bethesda, Maryland 20014

Attached is some material pertaining to a market opportunity suggested by Bob Bemer. He has suggested that we provide a Cobol compilation capability on the GE 600 for the purpose of translating original programs (source codes) into object codes ready for debugging or for running on IBM 360 computers.

I talked briefly with G. Feeney on this subject and he feels it is a good idea. He suggested I make you aware of this opportunity so that you could incorporate it into your Product Line Plans.

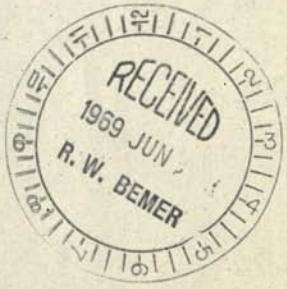
Regards,

J. W. Neuenschwander

J. W. Neuenschwander

JWN/dco

Note: This is typed copy of handwritten letter sent to Z. Quastler on May 20.



RB

GENERAL ELECTRIC

DIAL COMM 8•433 2569 DATE 1969 April 28 MAIL ZONE M2

Information Systems
Advanced Development and Resources Planning Division

DEPT.• Engineering and Manufacturing Integration Operation

ADDRESS•

COPIES• R. Kates

SUBJECT• Federal Government Compilation Centers

Messrs.: C. C. Black
R. M. Bloch
L. B. Cowles
C. W. Dix
G. J. Feeney
J. W. Haanstra
H. W. Paige
P. W. Sage
J. H. Sweeney
T. A. Vanderslice
G. F. Woodward

On January 7 I sent a letter to Mr. Smith, proposing to compile COBOL programs on the GE-600 into 360 object programs, as a salable service to 360 users. Most of you received a copy. Despite favorable comments, this proposal appears to be in limbo.

This is to advise you that Congressman Brooks' office has asked Dr. Grosch, of NBS, to prepare input for a bill to authorize and establish such centralized compilation centers for the Federal Government. Unless GE can provide this capability by that time, these centers will assuredly be equipped by IBM.

R Bemer

R. W. Bemer

po

BAYNARD, 69 MAR 10 – "PUSH IDEA OF REGIONAL CENTERS"

GENERAL ELECTRIC

cc: R.M. Bloch
G.J. Feeney
J.W. Haanstra

New York, January 14, 1969

Mr. Robert W. Bemer
Engineering & Manufacturing
Integration Operation
ADRP Division
Phoenix

Dear Bob:

Thanks for your letter of January 7 and your thoughts regarding COBOL compilation as a service application opportunity. I'm glad to see such examples of ways in which we can develop our mutually reinforcing strategy of offering the customer an optimum combination of service and equipment.

I have discussed it briefly with both John Haanstra and George Feeney. George will be in touch with you further on the matter. Since he is now located in Bethesda, you may want to suggest some potential Federal Government customers that he should talk with.

Sincerely yours,

J. Stanford Smith

meh

GENERAL ELECTRIC

January 14, 1969

cc: RM Bloch
PW Sage
GJ Feeney
WR Eaton
JS Smith

Mr. R. W. Bemer
Engineering & Manufacturing Integration Operation
Mail Zone 085
Phoenix, Arizona

Dear Bob:

I am very interested in your analysis of the service opportunity in COBOL compilation. If the technical possibilities are as you describe, then I certainly would concur that the opportunity should be pursued vigorously. Actually, a significant amount of present time-sharing service is engaged in FORTRAN compilation, which programs are later run on other equipment. The time-shared mode of using computer power is well proven in the program development and debugging area.

I hope our associates in the Information Services Division and at Phoenix will consider your suggestion very seriously, because I think it represents a very important connecting link with the data processing community. The only statement that surprised me in your letter was your evaluation that 10% of the IBM "market" is represented by FORTRAN compilation, and 2% is the usage of COBOL compilation. I would have expected it to be reversed, based on the much larger number of installations which today are doing the programming in COBOL.

In any case, if the technology of your "Step 2" is not too difficult, I agree that this is an important business opportunity.

Regards,

J. T. Coe

cdm

GENERAL ELECTRIC

DIAL COMM 8•433 4967 DATE 1969 January 7 MAIL ZONE C85

Information Systems
Advanced Development and Resources Planning Division

DEPT. • Engineering and Manufacturing Integration Operation

ADDRESS •

COPIES •

R. M. Bloch J. W. Haanstra
J. T. Coe P. W. Sage
L. B. Cowles J. H. Sweeney
C. W. Dix C. E. Thompson
A. L. Ellison T. A. Vanderslice
G. J. Feeney G. F. Woodward
+ WILDE

SUBJECT • Service Sales Opportunity

Mr. J. S. Smith, Vice President
and Group Executive
Information Systems Group

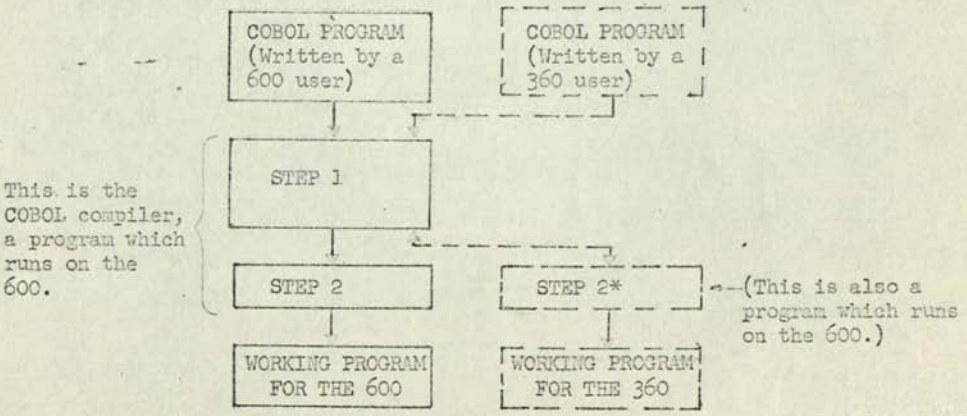
The Information Systems Group plans to wrest a market segment from IBM by replacing 360 hardware with GE hardware, presumably superior for the same work. A lead time of several years is required to produce this hardware and software.

There is an opportunity to get a similar result in a shorter time by performing some present 360 tasks upon our existing equipment, at better cost-performance. Furthermore, the investment capital required is less by two orders of magnitude!

Now that COBOL is a U.S.A. and Federal Government Standard, the opportunity exists to serve 360 users by compiling COBOL source programs into working 360 programs. Our planning people should have good figures, but I judge that the COBOL compilation process involves at least 2% of the total 360 workload.

Therefore this is an easy "second sale" opportunity for 2% of the computer market, just as legitimate as outright first sale of equipment. Transfer of workload to outside services smoothes and reduces capacity problems which otherwise give that revenue to IBM for more shifts or additional equipment.

Fortunately GE 600 COBOL has a modern table-driven processor which lends itself to this work. The diagram shows that there are two parts to a COBOL processor, the first (and larger) of which is independent of the computer upon which the working program is to be run.



Thus what is required is:

1. The construction of the Step 2* software for the GE 600, and

2. A sales campaign for the service provided.

Leroy Ellison and I claim that it would be very attractive for the 360 user (particularly up through Model 40) to utilize the GE 600 service to produce an object program which he could run on his own computer. The advantages of lesser cost come from:

1. The more efficient processing power of the GE 600.

2. The optional retention of the 360 COBOL program in 600 mass storage for inevitable correction and reprocessing. On the small 360 itself, each compilation (it is not uncommon to change the program 40 times) takes as long as the first. If the intermediate results are saved in the 600, the corrected program can be produced at a small fraction of the cost.

3. The GE 600 COBOL has superior diagnostic capabilities (for programmer mistakes).

4. The 360 is freed from a disruptive class of work (IBM once studied whether to make a special computer for this compilation process).

My general proposal for regional centers to do COBOL compilations for the Federal Government was received with enthusiasm by Mr. Ernest Baynard, Congressman Brook's EDP aide, who appreciated the economies of scale. I feel certain that the Federal Government would be a fertile territory for selling this service.

In proposing this, Mr. Ellison and I are convinced of the feasibility. We are also enthusiastic about the opportunity provided.

R Bemer

R. W. Bemer

po

PS.-Would you believe, for example, that FORTRAN compilation represents about 10% of the market, and can be done in the same way?

R.W.B.

EMIO

1968 December 12

Mr. Howard Bromberg
Information Management Incorporated
447 Battery Street
San Francisco, California 94111

C O N F I D E N T I A L

Attached is a copy of the telex just sent to Hekimi.

The report that I read to you yesterday is a GE report by L. Durand. There would be no purpose in copying the entire document. The pertinent extract is as follows:

"COBOL: Mr. Hekimi circulated a long report (doct. GA/68/29) but his comments during the meeting turn down all the contained conclusions. After a lengthy discussion a clear and strong position is established in favour of:

1. Modify the ISO Standard to make it a true subset of the USASI Standard ("with" statement). In view of this, our representatives should do their best to convince the various ISO members bodies to accept voting upon this change as an "auxiliary ballot" when voting for the main part of the Standard (this tactic is subject to confirmation after discussion with SC5 Secretariat).

2. Keep ECMA members fully aware of future possible changes of USASI COBOL by maintaining TC6 active in the maintenance of the language in cooperation with the US Committees. Regarding this question we (L. Durand) naively put a question which is left unanswered:

   The ISO document, which represents the ECMA views on COBOL, is said to be more difficult to maintain than a USASI document, then why don't we have an ECMA Standard which would not suffer from the same procedural delays?"

I suggest we do not waste our time worrying about the last point raised.

The general strategy is transcribed here.

// COBOL STRATEGY

1. General

   1.1 Several countries shall write to USASI deploring differences between ISO and USASI revisions. They shall suggest that USASI modifies ISO version in order to bring it into identity with the USASI version.

   1.2 USASI shall submit the modified text as Draft ISO Recommendation to the ISO members' vote, adding a subsidiary question: "Do you also agree with the proposed modifications?"

   1.3 The whole thing is to be handled in the procedural way and not at meetings (neither of SC5 nor of X3.4).

2. Special

   2.1 Italy: to be handled by Fubini, Raymond (UNIVAC), and Prennushi (Olivetti)

   2.2 UK: to be handled by Pinkerton (ICL) and Pow (NCR)

   2.3 CH: to be handled by Hekimi

   2.4 Eventually: France to be handled by Durand (Bourgain) and Mme. Chasles \\

I believe this shall hold with the exception of the alternate addressing as outlined in the telex. Do you agree that we should have an early draft of the transmittal for vote, so that it may be reviewed very carefully?

You will recall that I agreed with your division-of-labor proposal on COBOL maintenance.

R. W. Bemer

po

SPEED UP!

COBOL STRATEGY

1. General

TO (OR VIA) H. BLOMBERG AS USA REP TO [redacted] SC5

1.1 Several countries shall write to USASI exploring differences between ISO and USASI versions. They shall suggest that USASI modifies ISO version in order to bring it into identity with the USASI version.

1.2 USASI shall submit the modified text as draft ISO recommendation to the ISO members vote, ADDING a subsidiary question: "Do you also agree with the proposed modification".

1.3. The whole thing is to be handle on the procedural way and not at meetings (neither of SC5 nor of X.3.4)

2. Special

2.1 Italy : to be handled by Fubini
Raymond (Olivetti) and
Cremonini (Olivetti)

2.2 UK to be handled by Pinkerton (ICL)
and Pow (NCR)

2.3 CH to be handled by Hehmi

2.4 Eventually: France to be handled by
Dumont (Bourgain) and
Mme Charles

EMIO

1968 December 5

Mrs. Ethel Marden
Center for Computer Sciences
and Technology
National Bureau of Standards
Gaithersburg, Md.

Dear Ethel:

I am keeping your data descriptive language problem in mind. It has seemed to me that this is the time to go to the theoreticians, somebody who is knowledgeable in set theory and logic. Perhaps that is why you are in contact with Al Perlis.

I attach a book review from the 1968 November issue of The Computer Journal, which you will have at your office. It is marked for possible clues.

Wilkes' article on page 260 of the same issue seems particularly interesting in this context, for his outer syntax is data format free. This is the type of man we want to interest in this problem. So is Peter Landin, the reviewer, who worked for me at UNIVAC.

Sincerely,

R. W. Bemer

po

cc: J. A. Gosden, MITRE Corp.

4967 1968 Nov. 27 C85

Engineering and Manufacturing
Integration Operation

T. A. Vanderslice
L. B. Cowles

Furnishing Software Instrumentation
to Customers

Ref: My Memo to V. S. Cooper
1968 August 20

Mr. C. E. Thompson, Manager
Marketing LSD

In response to your request for elaboration, here are a few thoughts in addition to the advantages you already know:

1. The ad campaign I had envisaged might have shown an automobile dashboard, with a pointer to the odometer (A) and the fuel gauge (B), with the notation that A/B = miles per gallon. Then "Is your computer wasting gas? Instrumentation will tell you. Only GE provides it. To adjust to your driving conditions". Then perhaps a few case histories.

2. We have a case history of our own, which I am using in a talk. You will recall that instrumentation found that the 600 FORTRAN processor spent 7% of its time in a 4-instruction routine, which was easily cut to 2, saving 3.5%. Looks perhaps trivial, but suppose that:

   - o IBM did it
   - o 25% of the 360s are used for scientific work and 40% of this is in FORTRAN compilation
   - o IBM has a parc of 20,000 at an average cost of $300,000

   Then .0035 x .25 x .4 x 20,000 x 300,000 = $21,000,000

   And that is what those two redundant instructions would cost the computing public. I argue that many a Congressman has been reelected for saving the taxpayer less. And this occurs all of the time!

3. A firm called Boole and Babbage is selling such a service to IBM 360 customers. IBM salesmen hate them and badmouth them, for it often shows how they don't really need that extra 360. This strategy would not backfire on us, for we have already trimmed the fat on the 600. (See attachment)

4. I am attaching a paper by Ellison and myself, which I had noted to you previously. There are many good sales points to be found for an ad campaign.

5. Congressman Brooks is chairman of the Government Efficiency Committee. He got GE to produce longer life bulbs. He would just love to have this instrumentation argument. He is influential in the WWMCCS buy. Get it?

6. Is it useful to consider the 3-D symbol as other than a cube, for this purpose?

R. W. Bemer

po

GENERAL ELECTRIC

Information Systems
Information Systems Equipment Division

DIAL COMM 8•433 4967 DATE 1968 Nov. 22 MAIL ZONE C85

DEPT.• Engineering and Manufacturing Integration Operation

ADDRESS•

COPIES•
L. B. Cowles
J. W. Haanstra
E. M. Koeritz
J. H. Sweeney
E. R. White
G. F. Woodward

SUBJECT• STANDARD DISC PACKS

Mr. R. E. Roberts, General Manager
Medium Systems Department

We are rapidly approaching a worldwide standard for disc packs, as we have already for magnetic and paper tape.

The recording format, part of the standard, will be based upon (if not identical to) the IBM format as used on the 1316 and 2316 packs.

Wisely, all drives used by GE are plug compatible with IBM equipment and can be used thereon in IBM format. Unwisely, in my opinion*, the format used presently on the 400 and 600 is such that:

a. Disc packs produced on GE equipment cannot be read by the rest of the information processing community.

b. Disc packs produced by IBM and others cannot be read upon GE equipment.

I contend that, based upon the background information of Attachment A, GE is falling into a trap if it does not provide for graceful acceptance and usage of standard disc packs. It is therefore proposed that (while agreeing that the DSU167 drive is a suitable interim product, and does not create a system problem):

The announcement of the DSS167 Subsystem should not be allowed until:

- It has been demonstrated that the design of the DSC167 is such that the capability to read at least a restricted subset of the IBM 1316 and 2316 formats can be added without infringing upon or changing the present design, without impacting software, and without modification to physical items such as power supply, size, etc.

- A Group Plan for moving to standard disc pack formats has been formulated, agreed and scheduled. This includes the DSC167 additions and the schedule for their announcement.

R W Bemer

R. W. Bemer

po

*Attachment B, a history of these decisions, is available if required.

ATTACHMENT A - BACKGROUND
(Standard Disc Packs)

Federal Government Trend

The implementation policy for Federal Government procurement of information processing equipment is expected to be signed by the Secretary of Commerce by 1968 November 29. It spells out mandatory compliance with standards starting 1969 July. So far only ASCII, magnetic tape and paper tape are specified. However, COBOL is soon to follow. Past performance predicts a 6-disc pack standard on the IBM 1316 basis (via X3.2.7) by 1969 September. The Federal standard would follow no later than 1969 November.

General Trend

More users are expected to incorporate substantial portions of the GSA schedules (which will contain specifications for disc packs, and especially formats for interchangeability) in their own purchase contracts.

Disc usage has surpassed tape usage this last year. Although we were interchangeable with the rest of the world via tape, we are not via disc!

Skyrocketing software costs and existing heavy investments will force more voluntary adherence to standards.

General Method of Being Standard

a. Adopt an industry, national or international standard

b. Adopt the IBM de facto standard

c. Patent and license to IBM, so you will be the same as IBM

Impact Upon GE of Being Non-Standard

a. Equipment is less salable

b. Users cannot meet supply or subcontract requirements of Federal Government (e.g., Social Security might require submissions via disc rather than tape)

c. Dual or redundant equipment may have to be supplied, possibly at our cost

Present GE Situation

Due to past decisions reached on short-range bases and muddy arguments,[1] disc pack drives attached to the controllers for the 400 and 600 product lines are:

a. Unable to read packs generated on non-GE equipment or the GE-100 line[2]

b. Unable to write packs to be read by non-GE equipment or the GE-100 line[2]

c. Unable to achieve the capacity of IBM-type packs

Existing in a Two-Format World

Assume:

a. An Advanced Product Line which conforms to IBM-type format on disc. This gives the problem of emulating the 400 on the APL, which has been a part of that Product Plan since the 1st quarter of 1968.

b. The desirability of having (possibly) the 400 and (certainly) the 600 handle IBM-type format for longer viability in the market. This gives the conversion problem for data, and possibly for software.

---

(1) In one case, by a vote.

(2) Even though the packs themselves are mountable on and readable by the DSU160. As presently planned, the DSS161 is excepted from this statement, although a difficult code conversion is required.

Possibility #1 - Convert from disc pack to magnetic tape or cards via previous equipment. Read via new equipment and write disc packs in the required standard format.

Difficulty - Once-and-for-all conversion is impractical, and in most cases the contradictory equipment will have to coexist. Few installations can live with the single conversion.

Possibility #2 - For the APL, equip the PCP so that it reads several formats, such as 400, IBM, 100, etc.

Difficulty - The PCP cost goes up unnecessarily to the new user who doesn't have the problem.

Possibility #3 - Have different PCPs, one for each format.

Difficulty - The user who does have the problem now has an even worse excess of equipment to rent or buy.

- How will the PCP know that it is reading the correct pack so the system will not blow? We cannot have green packs for IBM, blue for 400 and 600, etc. As of now there are no provisions for self-identification of format used. It can't be a physical identification, for the packs must be usable and reusable for different formats.

- There must also be a convention in operating systems to read a newly-loaded pack at least once for the sole purpose of determining the type of format, the labels and intended usage, and the alternate track information.

Possibility #4 - (Like #2, except for the DSS167 as applied to the 400 and 600.) We make a distinction between:

- The software system, as resident upon and working from the disc (it is not interchanged), and

- The data which is read from, processed, and written on the disc (it is subject to interchange).

Thus the controller DSC167 should be capable of processing either format! We should not have to do too much to the software, for that would be very expensive. The routines to handle data, however, are only a small part of that software and can be alternating.

GENERAL ELECTRIC

Information Systems

Information Systems Equipment Division

DIAL COMM 8•433 4967 DATE 1968 Nov. 22 MAIL ZONE C85

DEPT.• Engineering and Manufacturing Integration Operation

ADDRESS•

COPIES•

O. Beltrami
L. B. Cowles
J. W. Haanstra
E. M. Koeritz
R. E. Roberts
G. F. Woodward

SUBJECT• DISC PACK COMPATIBILITY

Mr. J. H. Sweeney, Manager
Marketing & Product Planning Operation

The Magnetic Disc Subsystem DSS161, for the GE-100, is shown by Specification 300740720 (1968 April 22 letter of transmission) to be format compatible with the DSS160 as used on the 400 and 600 lines. This is not the same as the DSS130 (which is also available on the 130).

However, neither plans nor software to exchange are referenced in the Product Line Plans for the

GE 600, 1968 July
GE 400, 1968 October 25, and
GE 100, 1968 February 26

This says that either:

a. The Product Line Plans are incomplete in this important respect, or

b. There is no requirement to interchange disc packs between the 100 and the 400/600.

In the latter case there is no reason to choose the 400/600 format in preference to the format that will undoubtedly become a worldwide standard (see my memo to Mr. Roberts, this date).

If there are plans to exchange, it should be done with the cognizance that:

a. This would put the 130 rather permanently beyond the capability to act as auxiliary to non-GE equipment which, I am sorry to say, is more predominant, and

b. Since both character code and size differ between the 100 and either the 400 or 600, a translation problem of no small magnitude exists, and frequent usage would be a factor in performance degradation.

I understand that the design of the controller for the DSS161 is at such a stage that this aspect could be reviewed.

R. W. Bemer

po

GENERAL ELECTRIC

Information Systems

Information Systems Equipment Division

DIAL COMM 8•433 4967 DATE 1968 Nov. 22 MAIL ZONE C85

DEPT.• EMIO

ADDRESS•

COPIES•
R. M. Bloch
J. W. Haanstra
R. E. Roberts
J. H. Sweeney
G. F. Woodward

SUBJECT• Case History on Group Policies

Mr. L. B. Cowles, Manager
Engineering and Manufacturing Integration Oper.

Two other memos issued today concern the problems of disc pack incompatibilities among our several product lines and the rest of the information processing community. I have studied the Group Policies to see how this should have been caught, as follows:

1. 03.0 Product Planning

Supporting information is supposed to include technical approach and feasibility, and implications with respect to the Group strategies and other Group products.

I have checked the Product Line Plans for the 100, 400, and 600. No mention of this problem is made. Did Group General Management make oversights in this respect? I think not, for other subjects of similar importance are not treated in detail.

2. 04.0 Product Life Cycle

Section IV says the Project Plan, which includes the Product Plan and the EPS (Outline, items a and b) is to be approved by Group General Management.

This was not followed for the DSS167:

- The Product Plan for the 400 line was approved by no higher than the Manager of Product Planning for MSD. The EPS was approved by no higher than the 400 line Project Manager.

- The Product Plan for the 600 line was approved by no higher than the Manager of Product Planning for LSD. The EPS was approved by no higher than the Managers of Engineering and Marketing.

3. 04.2 Product Announcement

It is at this step that the difficulty is caught. However, the prototype exists. Marketing is clamoring for earlier announcement, and these pressures make it more difficult to take the correct action.

R. W. Bemer

po

EMIO

1968 November 19

Mr. Alexander C. Grove
Director of Standards
BEMA/DPG
235 East 42 Street
New York, New York 10017

Dear Alex:

General Electric votes NO on the proposed revision to X3.12-1966, USA Standard Vocabulary for Information Processing, for these reasons:

1. The document does not indicate which of the definitions have been either added, removed, or modified.

2. Quite apart from our opinion of the technical quality of the definitions, we object to a revision of a copyrighted document which requires such extensive use (and therefore repurchase at considerable cost) without appreciable consideration of the new terms inevitably appearing in a period of more than two years.

3. We have serious doubts on the advisability of having a "standard" vocabulary in force, particularly when it conflicts with the approved vocabulary for the Information Systems Group of General Electric, which is the IFIP/ICC Vocabulary. As an international manufacturer we must use an internationally accepted vocabulary.

Sincerely yours,

R. W. Bemer

cc: D. Hekimi, ECMA

bcc: L. Durand
L. G. Lauri
L. B. Cowles, EMIO
H. H. Green, MSD
Lonergan + note RCA

4967 1968 Nov. 19 C85

Engineering and Manufacturing
Integration Operation

A BASIC Standard

L. B. Cowles
L. Stanton - ISD

Dr. James C. Castle
Manager, Engineering
Information Service Department

Attached is a file of documents pertaining to the standardization of BASIC. A short study will indicate the urgency of obtaining such a standard both within and without GE.

The number assigned to the standard is B02.12, with the first draft scheduled for FW52 (1968), having slipped from February 1 of this year. Since the language is now in the public domain and many variants have now been implemented by other manufacturers, the question arises "Who shall submit the draft standard to USASI X3.4?"

It had been our hope for GE to submit such a document. With the advent of CALL 360, it may be that IBM will attempt to fill this void by making their own proposal.

Due to the difficulty of getting a standard for the Information Systems Group, would it be agreeable to you for IBM to propose the standard?

R. W. Bemer

po

GENERAL ELECTRIC COMPANY

13430 NORTH BLACK CANYON HIGHWAY, PHOENIX, ARIZONA 85029 . . . TEL. AREA 602—941-2900

Information Systems

EMIO

1968 November 18

Mr. Alexander C. Grove
Director of Standards
BEMA/DPG
235 East 42 Street
New York, New York 10017

Dear Alex:

General Electric votes NO on the proposed forwarding of X3.2/759 (Magnetic Tape Labelling) to the sponsor and USASI, for these reasons:

1. It is inconsistent with ISO Draft Recommendation 1323 on the same matter, which was forwarded to the ISO Council for approval on 1968 June 10-12.

2. It is inconsistent with ECMA-13, 1967 November, on the same matter.

3. Putting the matter to this vote does not follow the recommendation of X3/SAC/114-3, which asked specifically for a review of the proposal after correction and retyping.

4. The proposed changes have obviously not been coordinated with either ECMA or ISO, as evidenced by the anguished letter of November 6 from Mr. A. J. Raphael to Mr. Phillips as Chairman of X3.

Although the proposed change may have merit, there seem to be technical arguments from ECMA TC1 against the change, and we believe these should be answered satisfactorily.

Further, I remind USASI X3 and the sponsor once again that General Electric is an international company and cannot afford to serve conflicting standards in the basic areas of information processing.

Item 6.1.2b of the minutes of the X3 meeting of 1968 July 25 indicates clearly that X3 was to have notified ECMA and ISO/TC97 of this problem. Mr. Phillips, as I recall, said that he would take care of the matter. It is quite apparent that the matter was not taken care of during this period of over three months, and General Electric stands on both procedural and technical grounds in its negative vote.

Sincerely yours,

R Bemer

R. W. Bemer

po

cc: E. H. Clamons, X3/SAC
D. Hekimi, ECMA

X3/156
1968 November 15

DPG Standards Committee
and
USA Standards Committee X3
Computers and Information Processing

30-DAY LETTER BALLOT ON PROPOSED USA STANDARD MAGNETIC TAPE LABELS FOR INFORMATION INTERCHANGE

Resolved that:

X3 approve the proposed USA Standard Magnetic Tape Labels for Information Interchange (X3.2/759)* for immediate transmittal to the Sponsor, with the request that it be forwarded to USASI for approval and issuance as a USA Standard.

AFFIRMATIVE [ ]

NEGATIVE [X]

ABSTAIN [ ]

Name R Bemer

Organization G.E.

Date 1968 Nov 17

If you find that you cannot vote in the affirmative and desire to be recorded as abstaining or in the negative, please so state and explain the reason for your position.

Please execute and return this ballot no later than 1968 December 16 to:

Director of Standards
BEMA/DPG
235 East 42nd Street
New York, N. Y. 10017

*X3.2/759 is X3.2/513 as revised by the substantive change described in document X3.2.5/130 and the inclusion of editorial changes suggested by comments received during the balloting period of Letter Ballot X3/92.

8B

4967 1968 Oct. 18 C-85

Engineering and Manufacturing
Integration Operation

RESOURCE UTILIZATION

L. B. Cowles
T. A. Vanderslice

TO: J. W. Haanstra

FROM: R. W. Bemer

You mentioned changing many assignments for the next year. An auxiliary scheme is to utilize more fully some of the potential in the Group.

Let me take some personal examples on the revenue side, which involves sales:

1. We sell the 600 against the 360 and 1108. I was in charge of software at UNIVAC during the whole production period for the 1107 software, which is mostly what is running on the 1108 today.

   Attached is my memo to Rader kicking off the 1108. I set up the EXEC 8 design team.

   Yet I have never been asked by the Division to aid 600 sales in any manner. I was personally responsible for closing 1107 sales to Boeing and Westinghouse.

2. We sell timesharing as one of the 3 dimensions. Above my desk is a framed copy of "Timesharing Paper #1", which I published in 1957 March while at IBM. One man suggested that Charlie DeCarlo should fire me because it was so opposite to IBM's public position at the time. This claim of #1 has never been challenged, although exposed in Scientific American.

   Yet the Division has never made any use of this in sales.

3. I said the Scientific 400 could fly, either CP-4 or CP-5. Now I hear that the program cannot get commitments from Sales.

3. Continued

I had offered to go personally with a sales squad, and had suggested using copies of Lecht's book on FORTRAN, for which I had written the introductory history. This offer is not taken up.

Has anyone ever thought of getting a list of all computer users that are:

a. Doing more than P percent of their work in FORTRAN, and are

b. Paying as much or more in monthly rental than such a 400,

and then offering (either by sales or by advertisement) to take N of their current problems in FORTRAN (not above a certain size, of course) and run them on the 400 to demonstrate a better price/performance ratio? With a sales call and privilege of demonstration as the reward if it is better?

po

4967 1968 Oct. 18 C-85

Engineering and Manufacturing
Integration Operation

YOUR INTERLEAVER

J. C. Carroll
L. B. Cowles
A. M. MorganVoyce
J. W. Weil

TO: J. P. Lipp, IDD

FROM: R. W. Bemer, EMIO

Let us assume your 30 x 30 matrix, consisting of 30 rows, each of which contains 24 bits of information and 6 checkbits* on a hamming code of distance 4, computed for those 24 information bits. Thus 900 bits, written serially on a disc track column-wise from that matrix, such that the adjacent bits of the information characters are separated by 29 other bits unrelated to them.

I note first that you are limited to the case (is it really so universal?) when only one burst error occurs in this 900 bit pattern. I do not know what bit density you envision, but a reasonable value for the present outer tracks seems to be about 960 bits per inch. In this case 900 bits takes up almost an inch, which seems a large enough interval to have perhaps more than one scratch to cause bursts.

Secondly, I come back to my original qualms about your method--specifically, a burst error is a burst error, regardless of the allocation of the bits to information content. As I see your method with the above matrix, the information cannot be transmitted serially or even in parallel 8-bit groups until all of the interleaved bits are read into a buffer (I know it can start shortly after the beginning of the last group of 30).

Now, if the bits have to be buffered up before the checkbits are computable, or before readout, isn't it true that checkbits may be computed upon the basis of any regular pattern in that array?

Then cannot the characters be recorded serially and let the checkbits computation represent the interleaving? Try a diagonal function, for example.

*Checkbits is the proper term. They are redundant only if they serve no purpose.

Advantage #1 - Don't wait to buffer up. Transfer immediately and resend only if correction is required (2 percent of the time rather than 100 percent?).

Advantage #2 - Perhaps the checkbit function is then designable such that fewer buffers are necessary.

Advantage #3 - Maybe other manufacturers could read our data straight, ignoring the detecting and correcting bits by software or hardware.

Finally, I still do not understand how you avoid using more than one buffer. Assuming that checking of the first set of 24 bits starts when the last checkbit enters, and even that the check and correction process is instantaneous (?), how would you read all 30 x 24 or x 30 bits out while the last 29 checkbits are being read into the buffer? It would seem that double buffering would be required at the very least, for we cannot stop rotation. As an input, the 600 system operates inefficiently as a whole unless approximately 36,000 bits are transferred without interruption.

po

EMIO

Burst Correcting Encoding

J. Carroll, EMIO
A. Morgan-Voyce, IDD

1968 October 1

TO: L. B. Cowles

FROM: R. W. Bemer

Per your suggestion I called Jim Lipp of IDD, 8*353-2394, and asked for his material on the interleaved encoding system.

For your information, he stated that there is nothing down on paper, supposedly for security reasons. He is probably too busy to write it up, although he says that he already has circuits designed and other work in process.

At first he could not understand my interest, and I suspect he doubted my competence to review. Did I know anything about fire codes, etc.? I told him that I wished to review it from a technical point so that I could give input to Mr. Sweeney when the proposed product came up for approval under Group Policy, and from a systems compatibility viewpoint to avoid further difficulties like the DSU-160 format. I was told that the 167 and 170 would be incompatible anyway, so did it really matter?

He said he would confer with Morgan-Voyce about this. Until convinced otherwise about the miraculous quality of this scheme, I remain convinced that IBM could have invented it.

GENERAL ELECTRIC

Information Systems

Marketing and Product Planning Operation

DIAL COMM• 8-223-1873 DATE• September 11, 1968

DEPT.• M&PPO

ADDRESS• Bridgeport, Bldg. 28-EW COPIES•

SUBJECT• BEMA/DPG Advisory Committee on Plans and Policy Meeting, September 5, 1968

R. W. Bemer ✓
J. C. Croyle
G. Eltgroth
J. W. Haanstra
W. L. Lurie

These are notes I recorded at the above meeting per the Agenda attached. Other attachments are selected in accordance with your individual interests. (Copies available to others on request.)

Agenda Item 4.1 and 4.2

National and International Standards Program Status reports on the work of USASC-X3 "National Standardization" and ISO/TC97 distributed (copies to Bemer). X3 is to have more comprehensive report by October including full recommendations for reorganization.

W. Dowd (IBM) spoke to the issue of International Standards with its proliferation of committees and overlaps which should be reviewed and simplified. R. Hindman (NCR) pressured for BEMA staff to take more leadership.

Resolved that X3 will make recommendations to BEMA/DPG staff for National and International Standards and all DPG members are invited to comment (send to staff) or to request more information of X3. Plan to approve report October 29 and present to BEMA Board October 30.

C. Phillips reported on his presentation of BEMA statement re USASCII file standards to Herb Grosch, NBS and Cunningham. NBS is meeting with all government agencies and Grosch has issued an informal philosophy statement (copies to Haanstra and Bemer). Grosch has declined to submit formally to BEMA any material on his intended actions.

BEMA will take no further formal action. It is expected, however, that Grosch and Cunningham will informally advise BEMA of intended directives and Phillips will maintain his informal contacts. (IBM appears to be counting on government confusion and counter pressures from government agencies and possibly change of direction after elections.)

4.3 BEMA Newsletter

Cost of newsletter is $52,000/year (9% to DPG). Generally, comments on survey are critical of newsletter. Staff to determine costs to publish only essential information (e.g. meeting schedules, etc) on less frequent basis.

5.1 Membership

No changes in membership.

From Booz, Allen, Hamilton list of U.S. digital and analog equipment manufacturers 11 of the 111 are BEMA members; 18 of 200 communication equipment manufacturers are members. Further recruiting requires change in dues schedule.

5.2 Finance

1967-68 estimated expenses and proposed 1968-69 budget submitted (copy to J. W. Haanstra). New budget has lumped regular dues and special assessments (which I think covers up unwarranted increases in staff expenses). 1967/68 estimate is $244K regular plus $152K special or $396K total. 1968/69 request is $359K, but FCC investigation appears to be cut almost $100K so that regular expenses are up about $50K. Spangle would not agree to discussion of individual items and suggested we write finance committee for details. Budget is to be voted on in October meeting.

Dues schedule changes are proposed to decrease dues for small companies and increase for large. (GE would change from old category 3 to new category 4 and dues and assessments would change from $31K in 1967/68 to $28K in 1968/69. Regular dues in 1967/68, however, were only $20K.)

New schedule does not appear to solve small company problem because increments are excessive for small changes in employment. Membership committee to report on what small companies can afford.

5.3 Education

BEMA/DPG will publish a career book tailored from the AFIPS one at a cost of $5,465 for 10,000 copies. (Details to J. W. Haanstra)

5.4 Trade Matters

Only two responses to August 20 letter requesting comment on proposed statement to government officials. Time extended to September 19. (Lurie to call or write letter.) Suggested letters will be directed to committee chairmen instead of working people.

5.5 Industry Statistics

Twelve responses to questionnaire received and all favorable. (IBM had not yet responded.) Agreement noted on work to be done to better define classes of equipment.

5.6 Data Processing/Telecommunications

Report issued (copies to Lurie, Eltgroth, and Haanstra).

a. Stamford Research given contract to analyze FCC responses, but SRI is questioning industry and hasn't even read BEMA response.

b. Tariff filing of AT&T mailed to committee members August 30 (McShane, Scheckman, and Saumati of GE) and needs immediate response. (Lurie agrees to handle.)

c. Committee recommended that "Privacy and Security" matter be assigned to Jim Holmes, staff member and new committee be created. Lonegran (RCA) and Dowd (IBM) will recommend composition and chairman of committee for decision next meeting. Committee requires a top level executive as chairman and must include engineering and public relations talent. DP industry should be more vocal and get correct information publicized.

5.7 Patents

Concensus of Patent Office response to BEMA letter is that they admit a goof. BEMA to respond suggesting we give advice on new forthcoming guidelines and request Patent Office to publicize their letter or let BEMA do it.

6. Service Management

Report issued (copy to Croyle). Phillips with help of R. Rose (Honeywell) to draft letter to GSA per recommendation of report. Spangle (Honeywell) visited Commissioner of Federal Supply Service Abisfeller who is personally pushing for government take-over of service.(Since government maintains its own elevators, it can do it for computers.) Also, Congressman Brooks pressing issue. Apparently, they want to take over in high density areas and let manufacturers do it elsewhere. Abisfeller feels government can save 35%. Clancy suggested to him that he put his questions in writing to each manufacturer. One positive approach is to determine how GSA might change its contract procedures to save money.

BEMA will ask GSA to write a request for suggestions to BEMA or individual suppliers. (Concensus was that we would be better off working with GSA rather than Boston Computer Group which has already decided to advise government to take over servicing.) DP industry must make positive suggestions rather than be negative on report.

It was agreed that BEMA cannot discuss cost of service which is an individual supplier situation.

7.1 Customer Provided Programming

John Voorhees, BEMA Counsel, immediately requested specific subject matter to be discussed to assure avoidance of anti-trust discussion. Phillips was charged to document government actions and published material on this subject for legal review as to applicability for discussion at October meeting.

7.2 Customer Provided Peripherals

Lewis R. Caveney (Bryant Computer Products) appeared before joint congressional committee claiming unfair discrimination and suggested DP manufacturers should be required to provide standard interfaces so that government can buy separate peripherals.

BEMA has an I/O interface committee which cannot agree on anything. (Caveney is an observer on committee and Mr. Brooks of Bryant Computer is on committee.) (Committee list to Bemer.) The fact that BEMA has a committee (with 5 government representatives) should be some help if DP industry is called to task. BEMA objective is to avoid being "anti-standards".

7.3 Metric System

Public law 90.472 authorizes study of advantages and disadvantages of conversion to metric system. (3 year study at $500K) BEMA/DPG standards committee charged to follow study and invite members from other BEMA Groups.

Next meeting Tuesday AM, October 29 - Continental Plaza Hotel, Chicago.

Jim (mb)

J. H. Sweeney

/mb

1972 August 21

J. W. WEIL

R. W. Bemer • 993-2569

Advanced Systems & Technology B106

DANGEROUS SITUATION WITH 6000 SOFTWARE

Certain policies of PCO management are in direct conflict with a growing 6000 Parc. Inasmuch as that growing Parc is desirable, it follows that the policies are not, and must be circumvented somehow.

Specifically, the number and quality of 6000 programmers has been reduced rather than expanded - by edict. The symptoms are now surfacing rather painfully. Some examples:

- The customer benchmarking facility is finding it very difficult to obtain systems programming help when apparent bugs arise. People are all willing, but they do not have the detailed knowledge of the program sections. This might be overcome with good documentation - but see the next example.

- Susan Brewer is an excellent documentor and programmer. Doug Wattier wanted to hire her to unravel the GCOS III documentation, and weed out some bad design and coding in the process. Feldman personally turned it down three times.

- GCOS is being maintained by a team with, in many cases, a depth of one. One crucial section is being maintained by a nice lady that we both know - Sarah Fleming - but she has no one to pass on her knowledge to and no time to document. She is within two years of retirement. Suppose that she had a heart attack!

Conclusion

600 software is written mostly in a weak assembly language. It is not documented for efficient maintenance in that language which would be a job requiring many more trained programmers than we have. The situation is being forced to worsen because it is not understood.

Unless Feldman's policies can be overridden [which one must admit would be difficult], we shall have to get around them. I confess that I see only one way: for ASTO, under the guise of its charter, to replace the operating sections of software by writing them in ALFA. At the same time, make the program self-documenting in the Dartmouth style.

A-34-A

COMPANY CONFIDENTIAL

GENERAL ELECTRIC

DIAL COMM• DATE• October 9, 1968

DEPT.• INFORMATION DEVICES DEPARTMENT

ADDRESS• Oklahoma City, Oklahoma

SUBJECT• ERROR TOLERATION FOR MEMORY SYSTEMS

COPIES• A. M. MorganVoyce
K. W. Morrissey
W. T. Watson
R. L. Watters

TO: Mr. R. W. Bemer
General Electric
LSD
Phoenix, Arizona

Problem: It goes without saying that any memory system to be used at a central processor must be inherently reliable. That is, information stored must be capable of being retrieved with an extremely low probability of error. Figures like 10^{-12} or 10^{-15} for bit error probability (unrecoverable errors) are sought but are not now accomplished.

Basic Choice: In principle, and demonstrated in practice, there are two means of achieving low error performance in any information transmission or storage system. These are:

1. Error Prevention
2. Error Toleration

Any real system, of course, uses some combination of the two means. However, it is my belief that, at present, our memory systems are heavily weighted toward "Error Prevention".

Present Practice: The "error prevention" route involves building equipment with smaller and smaller tolerances to achieve interchangeability and to avoid errors. Very conservative disc bit densities are also necessary to avoid errors. Measured results show we could now increase our bit packing density - but with the danger of decreasing our safety margin against errors.

Cost of Present Practice: The emphasis on close tolerances is highly expensive in manufacture, inspection, and rejection rates. Potential capacity of the discs, for example, are not exploited because of the attempt at error avoidance. Equipment backups which are needed certainly multiplies system costs.

Alternative: It is our belief that if our basic philosophy were modified so that the system could tolerate errors, many savings would automatically result.

COMPANY CONFIDENTIAL

Approach: The Error Toleration approach is that which is now just coming into use in modern communication systems to decrease system errors, or to increase thruput capacity, or both.

Principle: In brief, a fundamental step is to introduce redundancy in a certain efficient manner so that errors can be corrected (rather than only detected). At first glance this appears to reduce the capacity of the system, but this penalty can be more than recovered by running the system harder - hence creating errors - and then correcting them. This principle is exactly in accord with Claude Shannon's Information Theory and also found to be true in practice.

Example 1) Bell Telephone Laboratories have run an exercise to determine the "ultimate" digital error rate over a voice grade line. Present lines permit 2,000 bits per second without much trouble but at 2400 require leased line facilities. Under such condition bit error rates are (optimistically) about 3×10^{-5}. At most anything above this rate, errors become much more frequent. However, Bell recently proceeded to implement high speed data sets which run at close to 11,000 bps. Needless to say, error rates were high and of the order of 10^{-3} to 10^{-4}. However, when a 7/8 efficiency code was used a useful rate of 9600 bps was achieved. I am not certain of the resulting error rate but my calculations would indicate a range from 2×10^{-6} to 2.6×10^{-10}.

Again, Bell employed the principle: "Transmit like mad and then clean up the errors with coding", to get higher thruput and fewer errors. (See Bell System Tech Journal, Feb., 1966, article by R. W. Lucky.)

Example 2) In 1965 GE allowed the USAF to test our burst error detection and correction equipment (BEDAC) over a 7,000 mile hf radio circuit. All previous attempts at Error Prevention Techniques, such as huge increases in power and exotic modulation systems, yielded a marginal (10^{-5} or worse bit error rate) about 3% of the time and never a useful (10^{-7}) output. With BEDAC, 70% of the time we not only achieved 10^{-7} performance, but were completely error free while using low power. This channel error rate is among the worst in the world and is, of course, many orders of magnitude worse than any memory rates we have to contend with. (See attached memo "Test and Evaluation of BEDAC System Performance".)

Aside: NASA and USAF are now in the process of adding error toleration techniques to their world-wide digital communication networks.

Noisy Media: The examples indicated are examples of transmitting digital data through a noisy medium with high reliability using coding techniques. The disc memory and associated circuits may obviously be considered a noisy medium and differing from the communication problem in only minor detail. Errors are almost always generated during writing or reading and not during the dormant storage state. Hence all fundamental principles

are shared except that the disc problem is not nearly as severe. A most important point is that the coding improvement increases as the medium error rate decreases.

Error Toleration Potential: If a memory system were constructed to operate satisfactorily in the presence of errors, the following would result:

1. Extremely close tolerances could be relaxed. This would not only reduce manufacturing costs but would drastically reduce rejection rates and assure better interchangeability of discs and other parts. Permanent errors resulting from disc blemishes or transient errors from head bounce or other causes would be automatically corrected during the readout process.

2. Some present practices (which cost storage but are actually steps toward some error toleration) could be relaxed because of the stronger error protection afforded in the error correction system. Some of these present practices include:

   (a) Reduced disc bit densities
   (b) Disc coding which assures flux transitions (thereby reducing errors) at a sacrifice of about 50% storage capacity
   (c) Parity and other error detection devices

Feasibility: The feasibility of using coding is not so much a technical question but one based on economic and time factors. In both examples cited, coding equipment was moderately expensive but they performed on much more difficult communications media. However, in the disc case it is likely that coding costs would be but a small fraction of memory costs. As an example, one implementation that appears technically feasible, which would correct an error burst as long as 30 bits, would involve only about 1,000 bit storage elements and associated logical circuits. With LSI or even MSI circuits coming into use, costs would appear modest. The significant variation over present practice, however, would be a different handling of input and output data about the core and disc memories.

Uptime and Maintainability: A highly significant capability of EDAC is to permit operation during faults. Rather than causing catastrophic errors and a penalty of down-time, we may continue to operate while indicators alert the operator that errors are being corrected. Hence, repair or maintenance provisions can be instituted while the system continues to operate. Further, the detection during correction feature may be used as a diagnostic aid in making repairs. Example: Were EDAC used, say, in the form of single error correction, double error correction across a parallel

interface (word serial), not only could the fault be localized to the interface, but the particular faulty cable/connector could be identified (a very common fault). This results because the decoder would not only correct recurrent errors in the K^{th} bit of the word, but a neon indicator would show that the K^{th} bit which corresponds to pin K, and cable K is the offending member.

IDD Program: To this point, the EDAC description has been qualitative and written from a motivational standpoint. I will now indicate what IDD has done as part of its advanced memory effort in this area:

About a year ago a study was performed to determine certain EDAC parameters applicable to the DSU 160-170 family. Comments re application to APL should be obtained from A. M. MorganVoyce. I will enumerate some of the conclusions:

1. Compatibility: By the very nature of EDAC, additional redundancy is imbedded in the text and is therefore directly incompatible with other manufacturers' formats. Hence, EDAC will be limited to closed GE systems. This, however, may still permit plug compatibility on the data processor side of the decoder with an option for using the available redundancy. Further, interchangeability of discs within like GE systems can be provided.

2. Application: The principle philosophy in using EDAC was to be in terms of increased capacity -- increasing the writing density by a factor of from three to four while assuring an output bit error rate of less than 10^{-12}. It should be remembered that compatible changes in head design, media, and head height would also have to be made. In this regard, however, it has been found that such resolution factors become more critical and the margin afforded by EDAC may be needed. However, coding will not be used as a crutch for poor design.

3. Error Distribution: Empirical data indicates errors occur in bursts due to media imperfections as well as randomly scattered events. Hence, a fundamental decision was made to guard against both burst and scattered errors. Observed worst case bursts at present densities and extrapolating for a three-to-four density increase, using a modified double frequency magnetic code, produced a worst case burst length in the mid-20 bit range.

4. Code Choice: From an efficiency and simplicity standpoint, a distance 4 augmented Hamming code was chosen in conjunction with a square interleaver. This was chosen over convolution coding because of error propagation uncertainties and over Fire codes which perform poorly on scattered errors.

Code: 26 information bits, 6 redundant bits, 32 bit word. 0.8125 eff. reducible to:

24 information bits, 6 redundant bits, 30 bit word. 0.8 eff.

Interleaver: 32x32 reducible to 30x30.

Property: Before interleaver, each code word corrects one and detects any two errors. With interleaving bursts to 32 (or 30) bits are corrected and bursts to 64 (or 60) are detected.

Output error probability is better than about $620P^3$ where P is the mean disc bit error rate and the max burst length is not exceeded. Hence, an error rate of 1.2×10^{-5} would yield a corrected error rate of 10^{-12}.

Development Program: In the course of development, a softwave simulation of the coding/interleaving was used with a DSU 160. Gross defects intentionally created on the disc surfaces were fully corrected. At present, circuit boards of dual in line integrated logic constitute the coder/interleaver. Other implementations and diagnostic techniques are under study at the Research & Development Center and at ASTO.

If you should want any further general information on the subject, I'd be most happy to discuss it with you. I can also provide detailed data on the present EDAC such as block diagrams, flow charts, and so forth. I hope this partially fills your immediate needs.

To add one final word, the information contained in this letter should be considered highly proprietory. While the technical content is mainly state of the art, the business aspects are obviously quite sensitive.

James P. Lipp
Consulting Engineer
Advanced Memory Technology
Building 2, Room 25G
Extension 394
Dial Comm: 8*353-2394

/jet

This document was produced in connection with a research project sponsored by SDC's independent research program.

TM- 3392/000/00
AUTHOR R Bosak
Robert Bosak
TECHNICAL V. Almendinger
RELEASE E. H. Jacobs
for J. I. Schwartz
DATE 2/23/67 PAGE 1 OF 15 PAGES

TECH MEMO *a working paper*
System Development Corporation/2500 Colorado Ave./Santa Monica, California 90406

A PROPOSED FILE PROCESSING LANGUAGE

ABSTRACT

This paper presents a preliminary description of a proposed file processing language. Basic definitions of terms used to describe the language are given, as well as definitions of various types of files. Procedures for declaring the various file types are given and operations on files are specified. The paper also lists statistical functions, operations on sets and operations on simple values. Finally, a proposed notation for indicating levels of functions and intratable references is discussed. (The paper is presented at this time mainly to solicit comments from others working in this field on the feasibility of implementing this language.)

1. INTRODUCTION

This paper presents a preliminary outline of a programming language that deals with aggregates and complex structures of data as its basic elements. The language is based on an extension of the concepts developed by the Language Structure Group of CODASYL, which were reported in "An Information Algebra," Communications of the ACM, April 1962.

The language is presented here in its current incomplete (and probably inconsistent) form to:

1) solicit comment and criticism
2) report on the status of the project
3) supply some (perhaps) novel ideas to others working in parallel or complementary areas.

Work has begun on a prototype implementation of the language. The programming will be done in the version of LISP 1.5 operating on the AN/FSQ-32V. The prototype is intended to test the feasibility of implementing the language and to gain insight into the techniques that would be required to implement a useful and efficient system for the language.

The language will, of course, change because of new insights gained from trying to implement it, and because of new ideas gleaned from the comments of readers of this paper.

2. BASIC DEFINITIONS

The fundamental unit of data in the language is a file. A file consists of three elements: a file identification, a property list, and an entry set.

The file identification consists of a file name and a file description. A file name is a label either explicitly given or derived by using a subscript, a qualifying expression, or a functional form. In any case, a file name at any particular moment of time refers to exactly one file. The file description contains structural and other types of information about the file.

The property list of a file is an ordered set, each of whose elements consists of a property name and a property description. More than one file can have properties with the same name, but within a particular file the property names must be unique. A property may have the same name as the file to which it belongs. The property description specifies the type, range, and other information concerning the values that the property has associated with it. A file must have at least one property.

The entry set of a file is a set each of whose elements (entries) is a list of values. Each entry contains exactly one value for each property in the property list of the file. The list of values in an entry is ordered in the same way as the corresponding properties in the property list.

It should be noted in the above that "value" (as well as several other terms) is not defined. In this language all values are themselves files, and the terms "file" and "value" are used interchangeably.

A particular value in an entry of a file is also referred to as being a file one level down from the parent file. Following the usual reversed nomenclature, the topmost level is level 0, the next one down is level 1, etc.

Note that because of the circularity of our definition of "value," all files have an infinite number of levels.

All the values of a particular property must have the same property list. Since this list is common to all the values of the property, it is referred to as the property list of the specified property, although--strictly speaking--a property does not have a property list.

3. SPECIAL FILES

It is convenient to consider and to have names for special kinds of files. While certain functions (to be defined later) apply to all files, many are specific to the types of files defined below.

A table is an ordered file. At the time a table is formed, a fictitious property named ENTNO is created. Its values are integers that directly correspond to each entry's relative position in the table. ENTNO of the first entry is 1.

A list is a table with one property (besides ENTNO) whose name is the same as the name of the table. In addition to being a table, a list is also an ordered set.

A set is a file with one property whose name is the same as the name of the file.

A simple value is a set with one entry, which has a number or a symbol as its sole value. A simple value is also a list.

A constant is a simple value whose name is the same as its value.

A program is a list whose elements are symbols that, when taken as a whole, are meaningful expressions in the language.

A declared file is one whose name has been explicitly associated with a property list by a declaration statement.

A null file is one whose entry set is empty.

4. DECLARATIONS

4.1 FILE DECLARATIONS

Files are declared by specifying:

1) an integer that indicates the level of the file
2) the type of file, i.e., FILE (for a general file), TABLE, SET, LIST, and VALUE (for simple values)
3) the name of the file
4) additional information, depending on the type of file.

For FILE and TABLE types, the additional information consists of a description of the properties of the file. This consists of a sequence of declarations similar to the above. The integer in the property declaration is one greater

than the one in its parent file; the type indicates the type of values of the property; and the name is the name of the property. The additional information again depends on the type of values of the property.

Note that there is no basic difference (except for level) between a file declaration and a property declaration.

For SET and LIST types, the name of the property need not be repeated in the property declaration since it is the same as the name of the set or list.

For VALUE types, the additional information specifies in more detail the type and form of the simple value (i.e., REAL, INTEGER, SYMBOLIC, etc.).

Constants are effectively declared when they are used in a program, and do not need to be formally declared.

Example 1:

```
0 FILE PERSFIL
   1 VALUE NAME SYMBOLIC
   1 VALUE DEPT SYMBOLIC
   1 VALUE MANNO INTEGER
   1 FILE EDUC
     2 VALUE SCHOOL  SYMBOLIC
     2 VALUE YRLEFT  INTEGER
     2 SET   DEGREES
       3 VALUE  SYMBOL
   1 FILE SALHIST
     2 VALUE DATE    SYMBOLIC
     2 VALUE SALARY  REAL
0 END
```

Example 1 shows a file declaration for PERSFIL, a file with five properties: NAME, DEPT, MANNO, EDUC, and SALHIST. The values of NAME and DEPT are symbolic, simple values, and the values of MANNO are integer, simple values.

The values of the properties EDUC and SALHIST are declared as general files; each file of EDUC has three properties and each file of SALHIST has two.

Example 2:

```
0 SET SAMPLETREE
  1 SET
    2 SET
      3 SET
        4 VALUE SYMBOLIC
0 END
```

Example 2 shows a file declaration for a 5-level tree structure named SAMPLETREE whose end points are simple symbolic values.

To simplify notation, when the next lower level from a SET is also a set, it need not be declared (and similarly for LIST). For example, the tree structure in example 2 could also be declared as

```
0 SET SAMPLETREE
  4 VALUE SYMBOLIC
0 END
```

4.2 TRANSFORMATION DECLARATION

A transformation can be predefined and named for future use. The form is

TRANSFORM name (dummy variables)

followed by a list of assignment statements, followed by END. The names that appear in the assignment statements, but are not listed as dummy variables or constants, are interpreted in the context of the transformation call. (See the description of the TRANS function in Section 5.5.)

4.3 FUNCTION DECLARATIONS

Function declarations have the form

FUNCTION name (dummy variables)

followed by a program, followed by END. All names appearing in the program are local variables, dummy variables, or constants.

5. OPERATIONS ON FILES IN GENERAL

Two types of functions--file functions and property functions--will be considered in the language. The functional forms that apply to the two are identical. The distinction is one of level; a file function is one level higher than the associated property function. Roughly stated:

1) A file function requires one or more file names (all at the same level) as parameters, and creates a new file at the same level as the specified files. The name of the new file is the string of symbols that describe the function. Most file functions also require a property function to be specified for each file named as a parameter.

   Many of the file functions also can optionally modify the files named as their parameters. We will use the notation $X, where X is the name of a file, to indicate that the file

is to be saved, i.e., is not to be modified by the function.

2) A property function effectively names and, when applied to a specifie file, creates a new property for the file. A property function can be considered as a functional form that only has meaning when associated with a specified file. When it is so associated, it is applied to each entry of the file to produce a value for the new property for that entry.

The notion of a level of a function exactly corresponds to that of level of a file.

When a file that is referred to in a file function is a set, then its associated property function must be a function of its single property whose name is the same as the name of the set. To avoid confusion and the necessity of repeating the set name (which could be a quite complex function) in the property function, the property name $E may be used when the property is referenced. This in effect makes $E a synonym for the property name of all sets. $E may also be used in any case where a property of a file has the same name as the file.

If a file function creates a set, the name $X is used as a synonym for the property of the set when it needs to be referenced within the function. $X must be used in this case to avoid circularity. This arises because the name of the derived set is the string of symbols that describe the function. If we wished to refer to the single property of this set within the function, we would have to use this same set of symbols within itself and so on, ad infinitum. $X is only a notational device to avoid this circularity.

Since file functions create files and property functions create properties, they usually may be used wherever a file or property name is called for. One exception is that they may not appear on the receiving end of an assignment statement (see below).

To simplify the exposition, the following notation is used:

X,Y,Z will stand for file names or file functions.
F,G,H are property names or functions.
≡ means "is the same as".
T means a transform.

5.1 ASSIGNMENT

In the operation

X=Y

X must be a name. The effect is to create a new file named X with Y's property list and entry set. If X is a declared file, then Y's property list must be

identical to that declared for X. If Y is any of the special types of files, then X will be of the same type, and any arrangement of the entries will be preserved. Y is not modified by the operation.

Strictly speaking, the assignment operator is not a function, but is instead what we will call an imperative. It should not be confused with the operator EQ, which is a function.

5.2 SUBSETS OF FILES

The operation

SUBSET(X,F)

creates a file with the same property list as X and an entry list containing those entries for which F is true. Relative order of the entries is preserved if X is a table. After this operation is performed, X has only entries for which F is false unless X has been prefixed by a dollar sign.

For example (refer to Example 1, Section 4.1):

SUBSET($PERSFIL, MANNO GR 100)

is a file whose name is the above expression, whose property list is identical to PERSFIL, but whose entry set consists of only those entries of PERSFIL that have a MANNO greater than 100. PERSFIL is not changed.

5.3 GROUPING AND ORDERING

GROUP(X,F)

is a set, each value of which is a file whose entries consist of all of those entries of X that have the same value of F. X is not modified by this operation.

For example,

GROUP(PERSFIL, DEPT)

is the name of a set, with one entry for each unique value of DEPT. The single value in each entry is a file with the same property list as PERSFIL, and whose entry set contains all entries of PERSFIL that have a like value of DEPT.

ARRANGE(X,F)

has the same effect as GROUP except that the resultant set is ordered on ascending values of F. There must be a unique value of F for each entry of X. X is unchanged.

Note that

ARRANGE(X,F) ≡ ORDER(GROUP(X,F),CONSTANT($E,F))

The function CONSTANT is defined in section 6. $E is used here as the name of the single property of the set GROUP(X,F).

5.4 CROSS REFERENCING

CROSS2(X,Y,F)

is a file whose property list consists of all the properties of X plus all the properties of Y. Where X and Y both have the same property, prefixes are attached by a dollar sign to the property name to form the new property names, e.g., if A is a property name common to X and Y, then two properties, namely X$A and Y$A will be in the derived file. The derived entry set has one entry for every pair of entries, one from X and one from Y, for which F (a function of the properties of both X and Y) has the value "true". In specifying F, the above mentioned prefix notation must be used to distinguish between names common to X and Y. Because of this, X and Y may not be the same file name.

After this operation, X has in it only the entries of the original X for which there was no member of Y that made F true; similarly for Y.

CROSSn(X_1,X_2,...X_n,F)

is the same as CROSS2 except that the n files, X_i, are involved. The entry set has one entry for each n-tuple for which F is true. F is a function of properties of all the X_i.

After this operation, X_1 contains only those entries for which no combination of entries from X_2,X_3,...,X_n could be found to make F true, and similarly for all other X_i.

Note that

CROSS1(X,F) ≡ SUBSET(X,F)

For example, consider a file declared as

```
0 FILE LOCFIL
  1 VALUE DEPT SYMBOLIC
  1 VALUE LOCATION SYMBOLIC
0 END
```

Then

```
CROSS2(PERSFIL,LOCFIL,PERSFIL$DEPT EQ LOCFIL$DEPT)
```

is a file with the combined property lists of PERSFIL and LOCFIL (note that DEPT had to be prefixed by the file names). In this particular example, the CROSS function is used to "match-merge" the two files. After the operation, PERSFIL contains those entries for which no matching department number could be found in LOCFIL and vice versa.

5.5 TRANSFORMATION

```
TRANS(X,T)
```

is a one-for-one mapping of X (preserving order if X is a table). T is a set of assignment statements, one for each property in the derived file. Each assignment statement has the form

```
A=F
```

where A is a property of the derived file, and F is a function of the properties of X. Instead of a set of assignment statements, the name of a pre-defined transformation may be used. X is unchanged.

Note that if the transform consists of only a single assignment statement that has $X as the name of the property of the derived file, then that file will be a set.

Names of properties of the derived file may be used on the right hand side of the assignment statements as long as no circularity of definition results. If the name of a property of the derived file is used on the right, and its name is the same as a property of X, then it must be prefixed by a dollar sign.

If X is a table, the up-and-down-arrow notation (see Section 9.0) may be used for functions of properties of X; only one or the other, not both, may be used with properties of the derived file.

The transformation function can be combined with other functions by placing the set of assignment statements after the last parameter of the function. Thus

```
CROSS2(X,Y,F,T) is the same as
TRANS(CROSS2(X,Y,F),T)
```

For example,

```
CROSS2(PERSFIL,LOCFIL,PERSFIL.DEPT EQ LOCFIL.DEPT,NAME=NAME,LOCATION=LOCATION)
```

creates a file with two properties, NAME and LOCATION. It has one entry for every matching pair of entries from PERSFIL and LOCFIL.

5.6 COMBINING FILES

```
COMBINEI(X,Y)
COMBINEJ(X,Y)
```

Each of these functions is a file whose entry set is a combination of the entry sets of X and Y. The difference is that the property list of COMBINEI consists of the intersection of the properties of X and Y (only those properties that have the same name and description in both X and Y), while the property list of COMBINEJ consists of the union of the properties of X and Y (the set of unique properties of X and Y taken together). In the latter case, null values are inserted where needed. If X and Y are tables, all entries from Y are placed after all entries from X and order is preserved within each. X and Y are made null files.

```
MERGEI(X,F,Y,G)
MERGEJ(X,F,Y,G)
```

Each of these functions is a table containing as many entries as X and Y combined. The intent of the terminal letter (I or J) is the same as in the combining functions above. The ordering of the resultant table is on F (a function of the properties of X) and on G (a function of the properties of Y). X and Y are made null files.

If an entry in X has the same value of F as some entry in Y does of G, then the entry from X precedes the one from Y. F must have a unique value for each entry of X, and G must have a unique value for each entry of Y.

5.7 INTERPOSITIONING

```
INTERPOSE(X,F,Y,G,T)
```

is a file with the same number of entries as Y. F must have a unique value for each entry of X. The properties of this file are those specified in the transform, T. X and Y are unchanged.

This transform is expressed in terms of the properties of both X and Y, but the up-and-down-arrow notation (see Section 9.) can be used only in reference to properties of X.

The intent of this function is to interleave entries from Y into the entries of X on the basis of G and F, respectively, and then perform the transform, T, on each entry of Y; the transform is evaluated using the values in that particular entry of Y and preceding and succeeding entries of X. A specific use for this function is interpolation.

5.8 CONDITIONAL FUNCTION

IF X THEN(Y ELSE Z)

is a file whose property list and entry set are the same as Y's if X has the value "true". Otherwise, it has Z's property list and entry set. Any arrangement of entries in the entry set is preserved. Y and Z are unchanged.

5.9 CONTROL OPERATIONS

GO S

transfers control to the statement labelled S. A statement label consists of a name immediately followed by a period, immediately followed by a blank.

IF X GO S1 ELSE S2

is the conditional form for transfer of control. S1 and S2 are statement labels.

The above control operations may be used only at the top level of a program or at the top level of a function declaration. The ELSE clause is optional.

6. FUNCTIONS ACROSS ENTRIES

In general, these functions produce a simple value as a result and leave the original file unchanged. The following statistical functions apply to a function F of the properties of a file X; they are self-explanatory.

```
SUM(X,F)
AVERAGE(X,F)
MAXIMUM(X,F)
MINIMUM(X,F)
MEDIAN(X,F)
PRODUCE(X,F)
```

Another class of these functions has to do with the organizational characteristics and the physical representation of the file itself.

NENT(X)

is the number of entries in file X.

NWDS(X)

is the number of words of computer storage required for the entry set of file X.

Other functions in this class are clearly needed and will be defined as the need arises in the implementation of the language.

A third class can best be labeled "miscellaneous".

CONSTANT(X,F)

has as its value F, if F has the same value for every entry in X. Otherwise, the value is null.

NULL(X)

has the value "true" if X is a null file.

7. OPERATION ON SETS

The following operations deal with sets, but not with files in general.

JOIN(X,Y)

is a set containing all of the unique elements of the sets X and Y combined. The property description of X must be the same as for Y. X and Y are made null.

INTERSECT(X,Y)

is a set containing all of the unique elements common to both of the sets X and Y. The property description of X must be the same as for Y. X and Y are made null.

LINK(X,Y)

is a set containing all of the elements of set X and all of the elements of set Y. If X and Y are lists, Y is tacked onto the end of X and the elements maintain the same order they have in X and Y. The property description of X must be the same as for Y. X and Y are made null.

EXECUTE(X)

executes the program X. The effect is exactly as if the program were substituted for the EXECUTE operation. X can be dynamically modified, but not by itself, i.e., X can be compiled at run time; any modification it made to the list X would not be effective until the next time that X is executed.

This operation is an imperative and not a function.

8. OPERATIONS ON SIMPLE VALUES

There is the usual set of arithmetic, relational, and logical operators between simple values.

8.1 ARITHMETIC

+	Addition
-	Subtraction
*	Multiplication
/	Division
**	Exponentiation

8.2 RELATIONAL

EQ	Equal to
LQ	Less than or equal to
GQ	Greater than or equal to
LS	Less than
GR	Greater than
NQ	Not equal to

8.3 LOGICAL

AND	"and"
OR	Inclusive "or"
XOR	Exclusive "or"
NOT	"not"

9. UP-AND-DOWN-ARROW NOTATION

The pair of symbols ↑ and ↓ are used in two distinctly different senses*: (1) to indicate a difference of level of functions, and (2) to refer to preceding and succeeding entries in a table.

9.1 LEVELS OF FUNCTIONS

In discussing functions of files, it was briefly mentioned that there is a direct correspondence between the level of a function, and the level of the files that it deals with and produces. In those file functions for which

*It may be that this can create ambiguous conditions. If this turns out to be the case, some other way will be used to represent one of the two meanings.

a property function is required, the point was made that this property function is really only another file function, one level removed. It was initially felt that the two concepts of "file" and "value of a property of a file" would provide a consistent and complete (if not necessarily convenient) way of expressing any desired relationships among the levels of a file. It became clear, though, that some additional language elements are necessary, if only so that we can conceptualize operations on files several levels removed from the top.

The following notation is intended to answer some of the problems encountered in trying to apply file functions in certain situations. The explication is by way of example rather than by explicit definition.

PERSFIL←SUBSET(SALHIST, DATE GR 12/6/65)↑

is a file with the same number of entries as PERSFIL, but the values of the property SALHIST will have entries from the original value for only those entries whose date was after Dec. 6, 1925.

9.2 INTRA-TABLE REFERENCE

In specifying a function of the properties of a table as applied to each entry in terms of entries ahead or behind the entry in sequence, the notation used is as follows:

1) $F\leftarrow n(v_{-(n-1)}\ldots, v_{-1}, v_0)$
indicates a function of properties, F, applied to the nth entry back from the one being referred to. (That is, F has an ENTNO that is n less than it.) The v_i in the parentheses are "start-up" values for F. These are needed because F must be defined for each entry in the table and the first n entries do not have any entry which is n entries back from them.

2) $F\leftarrow n(v_{m+1}, v_{m+2}, \ldots, v_{m+n})$
where m is the number of entries, indicates a function of properties, F, applied to the nth entry ahead of the entry being referred to. Again, there is the need to specify values for F for non-existent entries.

Note that in either case n may be a function, but it must evaluate to a simple integer value. If n is a function, the number of values in the parentheses must allow for the maximum value of n.

10. I/O FUNCTIONS AND FORMATS

These have not been specified as yet.

11. CONCLUSION

Many other kinds of operations have been considered for inclusion in the file processing language described above, but they have been set aside for the time being. These include operations on arrays, functions to create and use concordances, and a capability for recognition of a pattern in a list. While these are important areas and consideration will be given to their later inclusion, they are not particularly germane at the present time.

APPENDIX A

This appendix is directed at those readers who have some familiarity with the Information Algebra and who may wish to compare a problem expressed in the File Processing Language with one described in the algebra. Given below is a program for the solution of the sample payroll problems presented in "An Information Algebra," Communications of the ACM, April 1962. The problem is described on pp. 202-203 of that paper.

```
DEFINE
0  FILE OLDPAYFILE
   1 VALUE MANID INTEGER
   1 VALUE NAME  SYMBOLIC
   1 VALUE RATE  NUMBER
   1 VALUE TOTALSALARY NUMBER
   1 VALUE PAYPERIOD   INTEGER
   1 VALUE SALARY      NUMBER


0  FILE DAILYWORKFILE
   1 VALUE MANID INTEGER
   1 VALUE HOURS INTEGER
   1 VALUE DAY   INTEGER


0  FILE NEWEMPLOYEEFILE
   1 VALUE MANID INTEGER
   1 VALUE NAME  SYMBOLIC
   1 VALUE RATE  NUMBER
   1 VALUE PAYPERIOD INTEGER
END


TRANSFORM   HOURCALC()|
    MANID = CONSTANT($E,MANID)
     TEMP = SUM(IF HOURS LS 8 THEN HOURS ELSE 8)
    HOURS = SUM(IF HOURS LS 8 THEN HOURS ELSE 1.5*HOURS-4) + SUM(IF TEMP
             LS 40 THEN 0 ELSE 0.5*HOURS-20
END

TEMP1 = GROUP(DAILYWORKFILE, MANID)
TEMP1 = TRANS(TEMP1, HOURCALC())
```

APPENDIX A (cont'd)

```
TEMP2 = CROSS2(TEMP1, OLDPAYFILE, TEMP1$MANID EQ OLDPAYFILE$MANID,
         MANID = TEMP1$MANID,
         NAME = NAME,
         RATE = RATE,
         TOTALSALARY = TOTALSALARY+RATE*HOURS,
         PAYPERIOD = PAYPERIOD+1,
         SALARY = RATE*HOURS)


TEMP3 = CROSS2(TEMP1, NEWEMPLOYEEFILE, TEMP1$MANID EQ NEWEMPLOYEE$MANID,
        MANID = TEMP1$MANID,
        NAME = NAME,
        RATE = RATE,
        TOTALSALARY = RATE*HOURS,
        PAYPERIOD = PAYPERIOD+1,
        SALARY = TOTALSALARY)


NEWPAYFILE = COMBINEI(OLDPAYFILE, COMBINEI(TEMP2,TEMP3))
```

-4967 1968 September 25 C-85

L. B. Cowles
J. W. Haanstra
A. R. Wilde

600 PERFORMANCE IMPROVEMENT

Mr. C. A. Conover, Manager
Systems Design & Application Engineering

Having studied several aspects of Performance Improvement methods, at Mr. Haanstra's request, I come back to the technique that has always been the keystone to the entire work:

> Use Ellison's method. Operate a day or so in normal environment except with the timer runout set so low (3 or 4 msec?) that the various programs hardly have time to get started before the whistle blows.
>
> Keep track of the absolute instruction address where the interrupt occurs. After operating this way for some time (a day's worth may take two days), order and count these addresses, relating them to the program elements operating. This gives a kind of Monte Carlo simulation of the frequency of execution of those program elements.
>
> Now mine the highgrade ore, using this data, human analysis, and mapper.

It is my opinion that studies such as "600 Line Processor Usage Statistics," #120, are only of secondary assistance in this problem. I don't want to know that 28% of all instructions executed are load and store; I want to know what were they used for and was it useful and necessary? I don't want to know what is going on as much as I want to know why it isn't going on better!

A given process can be:

1) Unnecessary
2) Done more times than necessary (i.e., rerun)
3) Too slow due to hardware
4) Too slow due to software
5) Too slow due to hardware-software imbalance

6) Undesirable but imposed by conflicting or nonexistent standards
7) Inutile because of logic conflicts
8) Satisfactory

All of these can occur because of:

1) System software (in this case we fix it)
2) Customer usage (in this case we advise and control default options)

Here is a minor and incomplete checklist of possible areas of modification, depending upon what the data tells us:

1) GECOS III - Has it been proved not further tunable the way GECOS II was?

2) Adaptation of system philosophy:

   - Reduce usage of unreliable electromechanical devices such as card equipment

   - Reduce unnecessary printing. Force a higher penalty for dumps or eliminate altogether. Provide programs for doing the same type of analysis the user does from printouts.

3) Speed up data I/O

   - Better blocking and deblocking (Does my 4% thruput improvement to Get and Put have a corresponding trick in GECOS III?)

   - Overlap seek of linear files (CALL 360 example, for 4 or more physical units)

   - Is double buffering forced at every appropriate time?

   - Organize for faster peripheral access and data transfer

   - Better use of Output as Input (function of a file structure system, and don't forget we have plenty which don't really talk to each other - COBOL, IDS, FORTRAN, etc.) Minimize reloading and relabelling of interchangeable media (Is a high speed copy faster than the operator change and unreliability?)

4) Speed up Language Processors

- o Improve high usage elements, go direct to object code, force change by patching object code without recompiling the whole program, build in forced segmentation.
- o With this segmentation, consider multiple compilations (i.e., FORTRAN processor compiles many sections of the same FORTRAN program concurrently. Doesn't necessarily mean many copies of processor, nor say that reentrant coding is vital).
- o Analyze BASIC and 400 TS FORTRAN processors to compare which features really slow down, and can they be in on an exception basis? Do same for infrequently used features of languages. Then can programs be split into two groups, one group being put aside until its brand of processor is in store?
- o Analyze Lou Gatt's trick with 1107 FORTRAN, writing forward on the FH880 drum in such an interlace pattern that the files read logically backward.

5) Speed up Sort Programs

- o Check 400 disc sort for ideas. Any usable principles?
- o Take advantage of biased ordering? (some new papers available in this area)

6) Some additional Hardware?

- o More channels? (Ellison's canon says the optimum number of programs running concurrently is the number of channels divided by 2)
- o 645 type memory interlace?
- o Double index register size to full words? Or double the complement of arithmetic registers? (This would require recoding the highgrade routines, but might be worth it, then the rest could come by gradual usage in replacement programs during maintenance)
- o Store commands for sorting?

7) Eliminate IOC by direct connect to 355 (opinion - possible gain; if so, changes to GECOS III are worth it).

8) Hardware Speed Up?

   o Circuits, arithmetic, store access (opinion - very little to be gained here for present time).

R. W. Bemer

RWB:sm

GENERAL ELECTRIC

Information Systems
Information Systems Equipment Division

DIAL COMM 8•433_____ DATE August 22, 1968 MAIL ZONE _____

DEPT.•

ADDRESS•

COPIES• A. Conover

SUBJECT• 600 Performance Improvement

Mr. R. Bemer
Bridgeport

Bob, as I look at the situation, it appears to me as though performance improvement stopped a couple to three years ago on the 600 when we got into hardware difficulties.

As I poke at the situation, it becomes increasingly apparent that there are a whole host of fairly minor items, each of which can contribute significantly to our current performance. I am sure you have a lot of these stirred up yourself.

I have been in discussion with Al Conover on this subject, and I am sure he would be appreciative of the list that you would generate as input to his program.

John W. Haanstra

JWH/sab

ACRONYMS LIST

BUSINESS DATA PROCESSING COMPILERS

COBOL	Univac II, Univac 1103A
COBOL 60	USS 80
COBOL 61	Univac III, Univac 490, Univac 1107, Univac 1105
COBOLETTE	(Mod X)
AIMACO	Univac 1105
E-1	USS 80
FLOWMATIC	Univac I, Univac II
PROGENY	USS 80

NUMERICAL MACHINE TOOL CONTROL COMPILERS

APT III	Univac 1107
ROHR NUMERICAL TOOL CONTROL	USS - I 80

SYMBOL MANIPULATION & LIST PROCESSING COMPILERS

IPL-V	Univac 1105

MATHEMATICAL & ALGEBRAIC COMPILERS OR INTERPRETERS

ALGEBRAIC COMPILER	Univac 1107
APT	Univac 1103A
ALGOL 60	Univac III, Univac 1107, Univac 1105
BAMABELL	USS-I 80
BELL	USS-I 80, USS-I 90
CALCULUS	Univac I
FAP	Univac 1103A
FORTRAN I	USS-80
FORTRAN II	USS-I 80, USS-I 90, USS-II 80, USS-II 90, Univac 1107
FORTRAN IV	Univac 1107, Univac III
GAT	Univac 1105
INTERCOM	USS-I 90
IT	Univac 1105, Univac 1103A
MYSTIC	Univac 1107, Univac 1103A
NELIAC	Univac M460 Countess, M490
UNICODE	Univac 1103A

ASSEMBLY LANGUAGES

ACUTE	Univac III
ALMOST	Univac III
AS-1	Univac 1206
AUTOCODE	Univac 1103
AO, A1, A2, A3	Univac I, II
B-FORMOST	Univac III for 1107
CAP	Univac 1103A
CS-1	Univac M460
CUT-AS	Univac 1218
FLIP	Univac 1103, 1103A
FORMOST	Univac 1107
GP	Univac I, II
GPX	Univac II
K5	Univac LARC
LAS	Univac LARC
MISHAP	Univac 1103, 1103A
~~PAL~~ SPAR	(Mod X)
R III	Univac M460
RAWOOP-SNAP	Univac 1103, 1103A
RELCODE	Univac I, II
S4	Univac SS80/90
SAIL	Univac LARC
SAL	Univac LARC
SALT	Univac III
SLAP	Univac 1103
SLEUTH	Univac 1107
SPURT	Univac 490
TRANSUSE	Univac 1103A
TRIM	Univac 1218
USE	Univac 1103
UTMOST	Univac III
X1, X2	Univac I, II
X1L	Univac I
X6	Univac SS80/90
Z	Univac SS80
1107	Univac 1103A

OBSOLETE LANGUAGES

ACT	Univac I
BIOR	Univac I, II
BOEING COMPILER	Univac 1103A
CHIP	Univac 1103
COMPILER I	Univac 1103A
FLIP-SPUR	Univac 1103
MATH-MATIC	Univac I, II
MJS	Univac I, II
OMNIFAX	Univac I, II
SHORTCODE	BINAC, Univac I, II
SNAP	Univac 1103A

OPERATING AND LOADING SYSTEMS

BOSS III	Univac III
BOSS 7	Univac 1107
CHIEF	Univac III
CLAMP	Univac 1107
DUTY	Univac III
EXEC	Univac 1107
MASCOT	USS
PHEM	Univac 1107
REX	Univac 490
TOPS	Univac 490

LIBRARIES

DPL	Univac III
LION	Univac 1107
SUPPORT III	Univac III
SUPPORT 7	Univac 1107

IBM International Business Machines Corporation

Armonk, New York 10504

Office of Director of Commercial and Patent Relations

June 17, 1968

Mr. R. W. Bemer
General Electric Company
13430 North Black Canyon Highway
Phoenix, Arizona 85029

Dear Bob:

We certainly appreciate the effort you have put forth in helping to arrange for participation between GE and IBM on the public documentation exchange program.

Mr. Richard Imershein, of our Commercial Analysis group, will contact your Mr. R. D. Hill later this week by telephone to work out the necessary details.

If we can do anything to assist in this program, please let us know.

Sincerely,

Wally

W. C. Doud
Director-Commercial
and Patent Relations

WCD:ksb

1968 June 7

Wallace C. Doud, Vice President
Commercial Development
IBM Corporation
Armonk, New York

Dear Wally:

I understand from Mr. Kelliher of our legal staff that the details for GE joining the exchange program on public documentation will be worked out and agreements signed by Monday, June 10th.

The man in charge of this program for the GE Information Systems Group will be R. D. Hill, here in Phoenix. I understand that it is necessary to get your basic approval for the interchange and this is sort of an early notification and request to you.

Due to the importance to the System/360, our priority request for documents would certainly be for the set listed in the bibliography, Form A22-6822-6.

I would appreciate whatever expediency you might give this. It does seem funny to me to have worked in GE over three years without this facility when I provided the starting emphasis at IBM a decade ago.

R. W. Bemer

/s

cc: M. J. Kelliher
R. D. Hill

2/2/68

Mike Kelliher —

Will you please ease Ben
Berner's frustrations by sitting down
with him for an hour to complete
what has to be done for ISE
to participate in the documentation
exchange which he has previously
reviewed with Joe. This
is ok with me.

Bill Lurie

NOEL K. ZAKIN
MANAGER
COMPUTER TECHNICAL SERVICES

AMERICAN INSTITUTE
OF
CERTIFIED PUBLIC ACCOUNTANTS

666 FIFTH AVENUE
NEW YORK, N. Y. 10019
LT 1-8440

8*273-6139

Bob Bemer -
I discussed with
Algotte for dir. action.
He said he would talk with
you and then with Lucie
if nothing happens
again. Logan

GENERAL ELECTRIC

Computer Equipment Department
Phoenix, Arizona

SUBJECT

DIAL COMM 8*433 3658
MAIL DROP B-121

COPIES: W. L. Lurie

RECEIVED
JUL 6 1967
R. W. BEMER

1967 May 31

RECEIVED
JUN 5 1967
L. B. COWLES

Mr. Logan B. Cowles
Manager, Engineering
Systems Integration
Information Systems Division
Electronics Park
Building 7, Room 115
Syracuse, New York 13201

Mr. Clamons of UNIVAC informed me today of an IBM publication which would be of extreme interest to us. It is published by IBM in the UK and entitled "IBM System 360 Data Processing Standards"; the number is F-10-0001-0.

UNIVAC received this document through their normal exchange procedure with other computer manufacturers. Attached are copies of two memos bearing upon this exchange. The status is quo.

It has occurred to me that if the Information Systems Equipment Division could enter independently into such an agreement (Mr. Handros works for Mr. Wengert), then it might be equally correct for the System Development and Components Division to enter into such an agreement; it also contains two manufacturing departments.

If objection were to be raised to this, then it follows that even if the Information Systems Equipment Division did enter into such an agreement, they could not include documents originating from BGE and OGE in their exchange. Under these circumstances other international manufacturers, particularly IBM, might refuse to accept us as a member of the exchange. Basically, then, this is an Information Systems Division problem; if my information is correct, Mr. Lurie has no objection to our entering this arrangement. I consider it an extremely important matter and request action. ISMO in particular would be delighted.

HANDROS (X2396) NOW HAS MIKE KELLEHER. SAID (JUL 11) WOULD CONTACT ME.

R. W. Bemer

/cac
attachment

RICHARD IMMERSHEIN
IBM D.P. DIVISION
112 E. POST ROAD
WHITE PLAINS, N.Y.
914- 949-1900

RICHARD F. WELCH
UNIVAC
P.O. BOX 8100
PHILADELPHIA, PA. 19101
215- MI 6-9000 EXT. 3266

STUDENTS, EDUCATORS, CUSTOMERS IN NORMAL COURSE OF BUSINESS-
PROPRIETARY, > ANNOUNCEMENT. (f(GSA), CAN XCHNG
PRICES > ANNOUNCE. AUTOMATIC.
IBM WANTS TO, LOOK BETTER IN GOVT. EYES.

MIKE KELLHER HANDROS-2396
9872 M.K. IS READY. CAN U GIVE
HIM THE GO-AHEAD.

-47

3658
C-76

March 22, 1967

J. Handros
D. Rosner

Dr. R. L. Shuey
Information Sciences Laboratory
R & D Center
Schenectady, New York 12301

Regarding your request for technical information on the UNIVAC 1108, I have not been able to find the meager data that I had. However, Mr. J. Handros of CED has advised me that he will be able to fit GE into the general documentation interchange with other computer manufacturers within a period of two or three weeks from now. This generally beneficial arrangement will make the information you require available to Mr. Don Rosner of ISMD, who can then provide it to you.

R. W. Bemer

RWB:cm

GENERAL ELECTRIC

Information Systems Division
Phoenix, Arizona

DIAL COMM 8*433
MAIL DROP

SUBJECT

Documentation Exchange

COPIES: J. Handros
E. M. Koeritz
D. E. Rosner

October 26, 1966

Mr. R. W. Bemer
Consulting Engineer
Computer Equipment Department

Dear Bob:

I have discussed your letter of October 17 with Mr. Handros and have been assured that he will undertake the work as soon as possible.

Unfortunately, the current exchange arrangements have decided legal overtones which must be reviewed and, if necessary, revised in order for General Electric Company to participate.

I believe Mr. Lurie's concurrence was intended as an approval for Mr. Handros to work out the details necessary for our participation rather than approval of the program as it now exists.

Very truly yours,

L. E. Wengert

LEW:mp

9/29/66

B54AA O B20CD 199
PHOENIX ARIZONA 9-28-66
BUGELEC
PARIS FRANCE

VERNIERES. RE MANUAL DISTRIBUTION. EXPECT THIS WEEK G.E. WILL JOIN INTER-CHANGE AGREEMENT WITH OTHER COMPUTER MANUF-ACTURERS. MORE THAN THIS, MANUALS ARE COPYRIGHTED BUT NO RESTRICTIVE DISTRIBUTION CLAUSES ARE PRINTED ON THE PAGE CONTAINING COPYRIGHT. THEREFORE ALL CLEAR

RW BEMER
GECOMP

TELEGRAM

0084 1501 B200D C B54AA 0295
PARIS FRANCE 09/27/66
MR BEMER
COMPUTER PHOENIX ARIZ

TY 66/18369/270/RP
IS THERE ANY RESTRICTION WITHIN GE FOR DISTRIBUTION OF DOCUMENTATION LIKE COBOL MANUAL WITHIN A TECHNICAL COMITTEE E.C. ECMA COBOL COMITTEE IF NO RESTRICTION WE WILL APPLY THE SAME RULE IF ANY WE WILL LOOK FORWARD TO SEE HOW MUCH OUR POSITION IS LEGALLY DIFFERENT - REGARDS

J.C. VERNIERES BUGFLEC

-71

3050

C-76

D. E. Rosner

DOCUMENTATION EXCHANGE

1966 October 17

TO: L. E. Wengert

FROM: R. W. Bemer

While at IBM in 1957, I started an exchange of documentation among computer manufacturers which is operative to this day. General Electric is not yet a party to this agreement. Mr. Rosner and myself have been trying for some time to get Mr. Handros to enter General Electric ISD into this arrangement. So far we have had no success, as he states it will take about 6 hours of his time.

The purpose of this memo is to inquire if you could persuade Mr. Handros to do this at some early date. The expenditures within ISMO to obtain and xerox multiple copies of competitive documentation are exorbitant, when one considers all it requires to get these free is to send out perhaps 10 sets of our own documentation. Further, our competitive information is very incomplete and we operate under serious handicap at this time. True, the other manufacturers have difficulty knowing about us, which is 10% of their competition, but we have difficulty knowing about 100% of ours. Joining this agreement has already been approved by Mr. Lurie.

4967 June 7, 1968 C-85

Engineering and Manufacturing Integration Operation

Phoenix, Arizona

Reaching Architecture Decisions

1968 June 7

TO: J. W. Haanstra

FROM: R. W. Bemer

I indicated to you that:

1) The architecture decisions are very late.

2) Current proposals deviate unreasonably from Charlie and the 360, possibly because of lack of exposure to the Charlie documents.

3) Few of the necessary decisions and aspects thereof are being attacked.

4) So far most elements of proposals are unsubstantiated by references, statistics or written pros and cons.

5) A substantial number of the participating personnel do not comprehend the magnitude and basic importance of the problem.

6) The documentation mechanics are poor.

7) The operational and management structure is not conducive to reaching these vital decisions, on schedule or not. (This is true of the entire project, as far as I can see.)

Ellison is planning the meeting on phase-in for the other lines, tentatively on the week of June 17.

The proper corrective action in this case is to modify the line control and pattern of assignment. If we must continue on a task force basis, I suggest the following week (or two if necessary) for another meeting to capture the

essence of the architecture. By this I mean more than agreement; we must record the bases for agreement, with tradeoffs. Meanwhile, Ellison is achieving some progress in the architecture, and the statistics and references can be gathered, duplicated and distributed to those who will participate. Kelliher is scheduled to have the documentation interchange with other manufacturers (after 2 years of begging!) set up by today. We can then rush an order for 360 documents.

Such a meeting could be run in Phoenix; there are ways of insulation. The general mode should be to discuss in the presence of really high level documentalists, tape record the essence of and inputs to those agreements. While these are being typed for recap, the next technical items are prepared in groups for general discussion. The tenor should be to hear all arguments, and the leader decides - only a little tentative during the session, and very firmly at the end.

I have attached a proposed representation. I think it a shame that only one person from Charlie has been assigned for all this time, resulting in almost a complete start from scratch. I recommend conscripting Jim Wilde and any B-GE Charlie personnel that Lepicard recommends.

Of course, this is only one way to try to achieve results. An alternative which I believe more effective is to put these people together organizationally.

/s

1) DVIA (Deer Valley Institute of Architects)

Bemer	EMIO	Alternates -	Gaines	MSD
Cantrell	MSD		LeClerc	B-GE
Ellison	ADPO			
Knoke	MSD		Barronet	
Lepicard	ADPO		Bellec	
Merner	ASTO		Bienvenu	Charlie
Vance	ADPO		de Poncins	
Wilde	LSD		Milleret	
Zethraeus	ADPO		Verdier	
			Couleur?	

2) Documentalists, High Grade

Grems	MSD
Klick	MSD
Harrington	MSD

3) Live 360 Programmer with Detailed Knowledge

	- Gillis	Phoenix
GE	- McCoy	MSD, Valley Forge
	- ?	Evendale (GECENT Postprocessors)

Non-GE - Davis (TRW, El Segundo, Formerly GE)
- Nutt (CSC)
- Lecht (ACT)
- ?? (CUC)

UNIVERSITY OF PITTSBURGH
800 CATHEDRAL OF LEARNING
PITTSBURGH, PENNSYLVANIA 15213
DEPARTMENT OF COMPUTER SCIENCE

Area Code 412
621-3500 Ext. 7185

February 26, 1968

Mr. Robert Bemer
Weston Road
Weston, Connecticut

Dear Mr. Bemer:

We are considering the appointment of Dr. Walter Burkhardt as a member of the faculty in the Department of Computer Science and he has given you as a reference.

You no doubt are aware that we recently established Computer Science as a new graduate department at the University of Pittsburgh. I would be pleased if you could provide an evaluation of Dr. Burkhardt's potential as a teacher and his ability to conduct independent research. Our intent is to establish a first quality program in Computer Science and we recognize the need for the judgement of individuals like yourself to assist us in making faculty appointments for the development of the graduate program.

Thank you for your assistance.

Sincerely yours,

Orrin E. Taulbee

Orrin E. Taulbee, Ph. D.
Chairman

OET:sak

1968 March 14

Dr. Orrin E. Taulbee
Chairman
University of Pittsburgh
800 Cathedral of Learning
Pittsburgh, Pennsylvania 15213

Dear Dr. Taulbee:

This is in reply to your request for aid in evaluation of Dr. Walter Burkhardt.

Walter worked for me at UNIVAC, as you know, and I am pleased to consider him a personal friend. You ask for information in two areas:

1) Ability to conduct independent research.

   Here I think there is much evidence in the affirmative. He was in our Programming Research unit with such associates as Landin, Burge and Parham. I thought his work excellent and original. He made some substantial contributions in ALGOL-like languages.

2) Potential as a teacher.

   Here I have no evidence other than the style of his papers, as teaching was not a requirement for that position. However, my intuition from this tells me yes, he probably would make a fine teacher.

Sincerely,

R. W. Bemer

RWB/ek

MEMORANDUM FROM DR. W. H. BURKHARDT

Jan. 27, 1968

6562 Walnut Avenue
Merchantville, N.J. 08109

Mr. Robert W. Bemer
Manager Progr. Dept., GE Co.
13430 N. Black Canyon
Phoenix, Arizona 85029

Dear Robert:

I had hoped to see you at the FJCC in LA last fall, but could not find you. But I have seen Julian Goody, Jim Pontham and Fred Lawler there. They have told me that you are now back again in Connecticut.

The reason I am writing to you is that I have the opportunity for a teaching assignment at the University of Pittsburgh and they have asked me for references. So I have given your name and would very much appreciate if you could put in a few good words on my behalf, if they will contact you.

By the way, I have very much enjoyed your paper in the Communications on the ambiguity problem with

the characters. I have found a few more. So the number one "1", "1", "1", "I" and letter i as "I" and with number seven as "7", "7", "7" or "7"; then the last of it again with letter F! So, there is really no end in sight! the only help would be a standardized set.

With best regards, Your

Walter H. Burkhardt

A-14-A

GENERAL ELECTRIC

DIAL COMM• 8*353-2431 DATE• February 12, 1968 COPIES•

DEPT.• INFORMATION DEVICES DEPARTMENT

ADDRESS• Oklahoma City, Oklahoma

SUBJECT• DUPONT CHROMIUM DIOXIDE TAPE

G. C. Arndt
L. B. Cowles
H. H. Green
J. W. Haanstra
J. W. Sweeney
T. A. Vanderslice

R. W. Bemer
Computer Equipment Department
Mail Drop C85
Phoenix, Arizona

The Tape Memories Business Section does not plan to use the DuPont Chromium Dioxide Tape on their planned 1600 BPI Phase Modulation Program.

IBM is now delivering 1600 BPI equipment to the field in quantity using standard iron oxide tape media compatible to that used at 800 BPI. Their most recent announcement according to the February 5 issue of Electronic News clearly indicates the use of "standard computer tape" for their new 200 ips drive.

We expect the majority of our customers will insist on the same IBM compatibility requirement for 1600 BPI as experienced at 800. The present DuPont Chromium Dioxide Tape is not compatible with iron oxide on the 1600/800 BPI environment. Therefore, we have no choice but to implement our equipment using standard iron oxide tape.

Our 1600 BPI equipment is being designed with an "open door" for higher densities. We expect to use a different media for densities higher than 1600 BPI. This media could be chromium dioxide or a suitable equivalent.

As far as the $b_8 = 0$ problem is concerned we believe that IBM could open the question again at any time without regards to the media. We do not recommend considering the media and $b_8 = 0$ problems together.

Please keep us informed of any new information that you may uncover and contact our tape people any time you need information from them. We are very concerned about problems of this nature and would appreciate very much if you would send your comments from time to time.

W. T. Bayer

W. T. Bayer, Manager
Tape Memories Business Section

WTB:dm

INFORMATION SYSTEMS DIVISION
GENERAL ELECTRIC
1285 Boston Avenue
Bridgeport, Conn. 06602
27DE

DIAL COMM
8*-223- 1873

GE CONFIDENTIAL

January 25, 1968

Dr. T. A. Vanderslice
General Manager - Information
Devices Department
Oklahoma City.

cc: G. C. Arndt
R. W. Bemer
L. B. Cowles
H. H. Green
J. W. Haanstra

Dear Tom:

Please note the attached letter from R. W. Bemer to me asking for our plans on using DuPont Chromium Dioxide Tape.

Since we look to you as Manager responsible for world-wide integration of magnetic devices, it is appropriate that you and your organization should make this decision for the Information Systems Group. We will be pleased to help if we can but I believe this is a question which your organization should handle. Would you please respond directly to Bemer and key me for a copy?

Regards,

J. H. Sweeney
Manager-Marketing & Product
Planning Operation

cd

-3740
C-85

DuPont Tape

GE CONFIDENTIAL

G. C. Arndt
L. B. Cowles
H H. Green
J. W. Haanstra

January 23, 1968

TO: J. H. Sweeney

FROM: R. W. Bemer

Does G.E. plan to use the DuPont Chromium Dioxide Tape?

If so, at what densities and recording methods?

This is a vital input to external standardization representation. Honeywell has gone to 1200 NRZI with this tape on their existing drives, and may well provide 1600 NRZI for their customers.

It is reported that the DuPont tape is more reliable at 1600 NRZI than present tape at 1600 PM. The present USA Standard Magnetic Tape and the Draft Standard for Unrecorded Tape and Test Methods only go to 800 cpi NRZI; some elements of these are inconsistent with 1600 NRZI.

Thus the standard could be reworked, or it could be retitled for Ferr___oxide only. The danger of the latter is that the b_8=0 environment is embedded in the standard for the tape itself. IBM could then open the entire question again and really sabotage the USA Standard Code for Information Interchange!

Hope with me that IBM doesn't think of this strategy. It could cost G.E. in the millions.

It seems to me that we must take a very careful position in this matter, even to the possibility of limiting the standard to go no higher than 800 cpi. We might attack the DuPont tape for being hygroscopic, requiring a different environmental spec on humidity. The NSA magnetic tape certification is already under attack for reasons which include serious humidity requirements, and it is not yet known whether DuPont has submitted its tape for certification. Computron may help in this.

R. W. Bemer

RWB/ek

OLIVETTI
GENERAL ELECTRIC

Olivetti - General Electric S.p.A. - Sede Sociale in Caluso (Torino) - Capitale L. 10.962.000.000 int. vers.

Direzione Generale: Milano, Via G.B. Pirelli, 32 - Tel. 654.641

Direzione Progetti e Studi - Pregnana Milanese - Tel. 93.94.11

Pregnana, February 2nd, 1968
OM/eb

Mr. J. Carrol
General Electric Company
E. M. I. O.
1285, Boston Avenue
Bridgeport - Conn. 06602
U.S.A.

Dear Jim:

During my visit at Bridgeport on Jan. 5th, I had a too short discus - sion with you on the problems of the high density recording on tapes.

As I think this point to be of reasonable importance for ISO, espe - cially in this moment in which the overall planning is beeing review- ed, I have summarized below the facts at my knowledge and my recommen dations.

1. Summary.
The new "Crolyn" tape, made by Du Pont, permits to increase up to 1,200 bpi the packing density of the NRZI recorded tapes, whose pre - sent max. density is limited to 800 bpi, provided that the tape hand- ler features excellent tape guiding characteristics (dinamic skew).

2. Foreword.
For a better understanding of the interest of the new recording densi ty, used on the 200 series Honeywell computers, it seems valuable to recall briefly the chronological development of the magnetic tape stan dards.
The standards currently under consideration at ISO ASA or ECMA are not taken into consideration; attention will be focused on the "stan- dards de facto" which have acquired weight in the market place.

2.1. Standards "de facto".
In the area of magnetic tapes only the IBM standards have reached a diffusion wide enough to be of importance.
The standards introduced by IBM are the following:

7 tracks - 200 bpi NRZI
- 556 bpi NRZI
- 800 bpi NRZI

9 tracks - 800 bpi NRZI
- 1600 bpi Phase encoding

2.2. 7 tracks NRZI.
Starting with 200 bpi with the first 729 tape handlers and the 7 tracks format (punched card oriented), IBM reached the 800 bpi through an intermediate step at 556 bpi.
In so doing, IBM followed the policy of retaining the same information support and data format and of increasing the packing density according to the increase in performances of tapes and handlers.

It was possible, with this policy, to assure the access to the previous densities by merely changing some timings in the tape handlers. Existing tape files could be used at the packing density they were certified for, and below.

IBM judged that 800 bpi were a maximum over which a good reliability could not be guaranteed with 0.5 mil iron oxide coated tapes and within the limits of tape guidance accuracy provided by their pinch roller - vacuum bins tape handlers.

In fact, over 800 bpi, the reliability of the data recording is impaired by the limited resolution of the 0.5 mils iron oxide coated tapes and by the dinamic skew of the handler.

2.3. 9 tracks NRZI.
In 1963, with the 360 series, IBM introduced a new format in which, in order to conserve the same information support as in the previous 7 tracks system, the efficiency of the record was increased by going to 9 tracks (two numerical characters can be packed into one "frame")

2.4. 9 tracks, phase encoding.
In accordance to the policy of retaining the same information support in order to be able to use it also in the previous standards, IBM introduced, in 1965, a new standard in which the recording density was doubled by using the "phase encoding" instead of "not return to zero" technique.
This made it possible to overcome electronically the problems caused by the dinamic skew and to reduce, to some extent, the difficulties due to the limited resolution of the 0.5 mil tape.
The adoption of tapes with an oxide coating thinner than 0.5 mils was proposed many times in Industry Standards Organizations meetings, but the problem of maintaining the NRZI compatibility came out as the determining one.

A tape is considered as NRZI compatible when it gives a reading output at 200 bpi which is within ±5% of the signal given at the same density by the IBM master output tape.

At the present time 0.1 mil iron oxide coated are available, which would permit a higher resolution and, consequently, a significant increase in reliability, but they cannot be considered as NRZI compatible.

3. The "Crolyn" tape.
The "Crolyn" tape is coated with a 0,2 mil cromium oxide layer.
As the cromium oxide has a hysteresis cycle with higher B_R if compared to the iron oxide, a Crolyn tape with a 0,2 mil coating gives at 200 bpi the same output as conventional tape with 0,5 mil coating.

The Crolyn tape is, therefore, NRZ compatible and, being thinner, gives a higher resolution at high densities.

Honeywell, having a single capstan tape handler with a good tape guiding accuracy available, arrived at the conclusion of exploiting at 1200 bpi NRZI the superior resolution characteristics of this new tape

./.

OLIVETTI
GENERAL ELECTRIC

4. Recommendations.
Rumors have been heard about the reliability problems IBM is experienc ing with the present 1600 bpi - PE equipment.

It seems, also, that IBM is examining the possibility of adopting tapes with thinner coating to increase resolution renouncing, in so doing , the compatibility with the previous standards.

On the other hand, the new cromium oxide tapes appear to be very pro - mising under the reliability and cost stand points.

It has already been demonstrated that the Crolyn tape can work at 1200 bpi - NRZ on a Single Capstan tape handler (whose guiding accuracy is certainly not better than the accuracy of the GE Single Capstan) and this is not, probably, the upper limit.

It is not necessary to recall that the NRZI technique has a definite cost advantage over the P.E. technique and that the performances of the existing magnetic tape equipment of GE could be improved by merely adopting a new tape and modifying some timings.

It seems, therefore, to me that this possibility should be carefully e valuated from both the Product Planning and technical points of view.

Yours, very friendly,

O. Maggi

copy: L.B. Cowles

R Bemer

GENERAL ELECTRIC

Information Systems Division
Phoenix, Arizona

DIAL COMM 8*433 3658
MAIL DROP C84

SUBJECT

COPIES:

600 Timesharing System
for ISD

1967 September 18

TO: J. W. Haanstra

FROM: R. W. Bemer

There are some danger signals to be observed for three elements of the large scale timesharing system planned for Information Services. They are:

1. System Configuration - Hardware and Software

The R & D and ISDD proposals differ. Neither recommendation can be called more than intuitive in view of the absence of models. Both are high risk configurations simply because of the multiple suppliers of foreign hardware; there will be interface problems, mismatch, etc. These can all be sorted out--after a considerable delay which must precede the tuning of the system. There may be a secret to do this, but it is not known to the industry. The RTIOC does not conform to the Common Peripheral Interface and requires additions. Thus modified, CED now estimates the RTIOC would have about twice the reliability of the IOC-C. However, this is not the figure for the total system, nor does it consider the probable T^2L IOC in 1969. By present measurements the IOC-C has ample capacity for this system.

S10D
DATA PRODUCTS
PCBS
HONEYWELL 4[illegible]
GENL INSTR DRUM

2. System Efficiency and Effectiveness - Hardware and Software

It is required to model a target hardware configuration and then iterate to find a better one. The software configuration and interaction must then be modeled. It is presently expected that the software will be done by programmers outside of CED. Production software is more intricate than experimental, particularly as developed by universities (Dartmouth) (MULTICS). Available programmers are not known to have adequate systems programming experience, they have not fabricated at this level of complexity (SOS), and there are very few plans for instrumentation, which is crucial.

3. Production Method for the System

In view of the above, it seems unnecessary to take the intermediate step with the 605, which may not be a stable and mature system, just to get on actual hardware. While it is true that failure to meet schedules on such a system would bring an awareness of urgency to the workers, this is hardly a proper management device. An alternate plan is to model with a simulation of a 4 BAR CPU on a 635 under GECOS III (or II if the risk is considered too high). The modeling team should have the actual production programmers write the elements for their software units, to the interactive level only. Excessive detail is pointless. This will verify specs and allow modification when improvement or changes are found necessary. At the same time, Quality Assurance programs should be designed and written, and software tools should be constructed to aid in production (i.e., for comparison of test results, filters for screening code for insertion in the system, linkage modifiers, algorithm replacers, graphing programs to ensure non-interference with software units which should not be affected, etc.)

Further, the software system should not be written anew! GECOS III is an excellent design which has been produced to a point of considerable effectiveness in a short time scale, and has passed instrumentation tests very well. It has queued I/O, queued dispatching, and perhaps things that new developers might not think of. It has been tested for flexibility to change system anatomy. Cantrell says 90% of producing a timesharing system is developing a multiprogramming operating system of high capacity and reliability; the rest is adding the timesharing features per se. Thus we would be way ahead of the game to start with GECOS III and modify it sufficiently, particularly since the modifications start from a stable condition, and the cause is pinpointable if it goes bad. Not so with an all-new system. Knock out batch processing capabilities only if retention disables or lowers the value of the system. Even if not needed for the proposed class of operation, availability (although perhaps suppressed) will remain for future business decisions.

GENERAL ELECTRIC

Conclusion

There are two major paths to take:

1. Proceed with present plan. This is so risky that present CED plans (to continue their planned modification of GECOS III to a timesharing system) must be continued as insurance. In this case CED will probably have their system finished before ISD! When both are complete, they will be incompatible.

2. Bring ISD design goals and manpower to bear on the CED conversion as a joint effort. This might shorten the delivery schedule. It would certainly give a single, better product without compatibility problems.

R Bemer

po

67 Dec 6

cc: LB Coules - E,SI - Spt.
DB Schneider - SC -
Waynesboro, Va.

MM: M.M. Green - CED
L. Durand - B-GE
L.G. Lauri - O-GE
W.W. Foster - PRBS
R. Barton - ASTO

Subj: First Meeting
Industry Standards Subcouncil

The recent problem with CCR-A in the U.S. indicates that it is time to convene our first meeting under the sponsorship of the Engineering Council. As the bulk of membership is in the U.S., I suggest that the meeting be held in this country. Again, as one of our most important tasks is to get rebriefed by the U.S. Technical Representatives (the majority of which are in Phoenix), I suggest that locale.

An agenda is attached. January 16-17 has been chosen for two reasons; the first is that USASI X3.1, CCR, meets in Phoenix Jan. 15-18, and the second is the X3 meeting in New York City (BEMA) on Jan. 18.

Messrs. Durand and Lauri are invited to attend the X3 meeting as GE observers. I have mentioned this to BEMA and they are most welcome. However, it would seem even more beneficial if there were a possibility to be accredited by AFNOR and UNIPREA. I feel this is a splendid opportunity for the members of X3, most of whom have had no technical participation in international standardization, to get some first hand briefing, and indeed I will suggest to BEMA that an agenda item be set up for this.

One of our most important tasks at the subcouncil meeting will be to recreate a comprehensive list of tasks and decisions for 1968. This is to be related closely to the budget to provide our best estimate of the apportioning of benefit to the several components. Ranking various committees in importance of having representation will also be required, together with determination of whether our specific representation comes from the proper product area.

Attached is a copy of some material I have sent to both BEMA and ECMA. Youwwill note sample pages for future and past meetings. In the extreme right hand column (not furnished to the exterior), there is coded the expected attendance from

I Division
C CED
M MED and other
B B-GE
O O-GE

Would you please use this future meetings list as a basis to prepare for the subcouncil meeting, filling in

a. All planned representation
b. Major topics of work
c. Crucial points requiring GE positions

Reservations for Durand, Foster and Lauri will be made at the Executive House, Scottsdale, for nights of Jan. 15, 16. Advise of additional nights if required.

Note to H.H. Green: Please copy for technical representatives and arrange for their participation.

Notes to L. Durand and L.G. Lauri:
1) If you wish to attend part of the X3.1 meeting because of the OCR situation, please advise me and I will then make the necessary arrangements.
2) Suggested flight on Jan. 17:
AA 164 PHX 1515-2120 JFK

R. W. Bemer

RWB:dda
att.

GENERAL ELECTRIC COMPANY

1285 BOSTON AVE., BRIDGEPORT, CONNECTICUT 06602 . . . TELEPHONE 334-1012

INFORMATION SYSTEMS DIVISION

67 Nov 27

Messrs: C.A. Phillips, Director, BEMA DPG
D. Hekimi, Secretary General, ECMA

Gentlemen:

I believe it is the opinion of participants in the standardization work under your aegis, both manufacturers and users alike, that a really comprehensive master plan for interaction between BEMA and ECMA does not exist and that cooperative efforts fall short of achieving a close coordination. Such cooperative efforts as there have been in the past have been on an ad hoc basis for each emergency, whenever it became apparent (primarily via those who participate in both efforts) that conflicting standards were nearing adoption through the normal mechanisms. This is not to demean the cooperative efforts. For the most part they have worked well - but in crucial issues they have not kept matters from falling through the cracks.

Certainly some duplications of effort are most worthwhile, particularly to avoid provincial viewpoints and ignorance of other requirements. However, much other duplication is waste because of this lack of a master plan. I am convinced that both groups spend a great deal of money in ignorance because they are not adequately informed on what the other group is doing. The formation of the X3 Systems Advisory Committee is a belated step in the proper direction.

Action may be taken on two levels:

1) You gentlemen, as the top leadership, could extend your present cooperation to developing a master plan for possible adoption by both of your parent bodies. This could lead to more formal arrangements at the working levels.

2) At a simpler and more immediate level, a joint calendar and record of meetings could be published. I have had a formal request in for three months for BEMA to include ECMA meetings on the unsatisfactory calendar now published in its News Bulletin. More than this, a simple calendar does not explain adequately what the issues and emergencies are likely to be.

GENERAL ELECTRIC

Messrs: C.A. Phillips
D. Hekimi

67 Nov 27
Page 2

Attached is a format and method of such a calendar. This is comparable to that used by several DPG members internally. While these could be continued by them, it would be unnecessary if BEMA and ECMA were to publish a satisfactory schedule in line with their general responsibilities.

The format is self-explanatory. Each meeting is recorded on a punched card (layout attached.) Two lists are suggested - one for future meetings and one for past meetings as a matter of record (this will also give sponsoring groups some idea of the amount of money expended, in which they have an interest). The meetings are encoded from the master list of committees, structured in such a way that activities in the same and related areas are well grouped. The past list is suitable for periodic publication (photo-reduced) in professional journals as an indication of the activity. The future list is suitable for publication in the BEMA news Bulletin in the same way, and in suitable media for Europe.

I would suggest that BEMA take the responsibility for deriving the comprehensive list of past meetings from its files, or by soliciting the aid of the subcommittee chairmen, who might just as well furnish the information on cards. ECMA might do the same, and trade, both publishing the merged list.

This latter device is not very profound, but it could help for better standards coordination management, which is needed urgently.

R. W. Bemer

cc: X3/SAC

1967 July 27

Mr. R. B. Forest, Editor
DATAMATION Magazine
1830 W. Olympic Blvd.
Los Angeles, California 90006

Dear Bob:

Here is a copy of the ALGOL log in the form to appear in this fall's Annual Review in Automatic Programming. You may recall that McGraw-Hill said it was too lengthy to print in Lecht's book.

So I believe that I should get some formal permission from DATAMATION to use the excerpts included here. If I have the guts to print what you said about me [98], surely you have the guts to print it.

Lovingly,

R. W. Bemer

po

C-76

CONSTRUCTION LANGUAGES

J. W. Weil
P. A. Quantz
D. C. Klick
W. L. Sullivan
E. R. Vance

1967 May 2

TO: E. R. White

FROM: R. W. Bemer

Reference: My 67 April 25 memo to you, on Fortran Processor Design and Implementation.

In the referenced memo I deprecated POPS for Fortran processor construction and yet stated that Fortran "can afford special macroinstructions which produce efficient machine code".

Mr. Klick suggested to me that I was really suggesting POPS again, of the generative (rather than interpretative) type. I think not. I am suggesting an investigation of a feature which exists for the UNIVAC 1108, 1107, 1050, III and 418. This is known as the "PROC". Attached is a dissertation as it appears in the 1107 manuals.

It was a deliberate decision on my part to have this feature exist for all of the previously mentioned machines. An example of the resulting gains may be taken from the case of an assembly language processor written for the 418, to run on the UNIVAC III. Because of the power of this feature, the total processor construction and checkout was accomplished within 48 hours. You might contrast this with times for comparable projects here in Phoenix.

I have additional information available in the:

a) 418 ART Assembly manual - Pages II-22 through II-45

b) 1107 SLEUTH II Manual - Pages III-9 through III-27.

c) UNIVAC 1050 Manual - UP-2590 - Pages 106-124.

All of this material has been available here in my library since 1966 May 1, but in the environment of extreme pressure to finish existing software projects none of our programming people have taken advantage of it, despite my propaganda.

If additional consultation is required, you can contact David Ferguson, President of Programatics, in Los Angeles, who purveys a later version of this technique (which he developed while at CSC) as METAPLAN (which stands for Meta Programming Language).

If such techniques are not known generally to our programmers, this can be ascribed to a faulty GE policy which has kept us from receiving documentation on a free interchange process from other computer manufacturers. I am attaching (Mr. White only) copies of correspondence concerning Mr. Handros on this point.

/cac

File

-64

GENERAL ELECTRIC

Computer Equipment Department
Phoenix, Arizona

DIAL COMM 8*433 3658
MAIL DROP C-76

SUBJECT
• Cantrell/Ellison Instrumentation Work
Ref: Your 1967 April 18, Memo

COPIES: H. Cantrell
J. Couleur
L. Ellison
D. Klick
P. Quantz
E. Vance
J. Weil

April 25, 1967

Mr. E. White, Manager
CED Engineering
D.V.P.P.
Mail Drop B-89

You have said that you "do not concur with any plan for publicity at this time." I offered none, but rather proposed September 1st, as a target date.

We must consider the balance between:

1) Disclosure allowing competitors to use ours or similar methods, and
2) Restoration of confidence to our present customers, improvement to their operating efficiency, and better industry image for sales.

With respect to the first point, I submit that:

1) Information flow is remarkably sluggish in this industry dedicated to it.

Lou Gatt of CSC, who stands as the designer of the best FORTRAN processor presently existing, spent 7 months at IBM in Poughkeepsie as a consultant, and they never solicited his advice on processor design for the 360!

2) Credit and advantage should be taken while it is yet possible. For example, here is the abstract of a paper given last week at the SJCC, heard by some 300 people:

"SNUPER COMPUTER - A COMPUTER INSTRUMENTATION AUTOMATON

The purpose of this paper is to discuss approaches to implementation of a system whose goal is observation, display and possible interaction with on-going activity of other information processing systems.

The first part of the paper establishes the kinds of events which we consider as interesting observable behavior of an information processing system including its central equipment, system programs, user programs and user behavior as reflected through terminals. The nature of the instrumentation to measure that behavior is then discussed.

The second part of the paper presents system approaches to the instrumentation problem and describes the characteristics of a system under development at UCLA."

It is important to note that this work involves a computer of the IBM 360 line, and the results will certainly be available to them.

With respect to introduction of the method, I agree with the people involved that our A.E.s have a great need for this tool to perform one of their main functions-helping the customer to get the most value from his installation. You have concurred with this approach.

You mentioned discussing the alternatives with ISMO. It would be of the greatest importance if you could do so sufficiently in advance of the May 10-12 Users Meeting here in Phoenix. ISMO might find it greatly to our advantage to discuss this work with selected 600 users, in an advance release.

Robert Bemer
Consulting Engineer

RB:cm

GENERAL ELECTRIC

Computer Equipment Department
Phoenix, Arizona

DIAL COMM 8*433-2597
MAIL DROP B-89

SUBJECT
• Cantrell/Ellison Instrumentation Work

COPIES:
H. Cantrell
J. Couleur
L. Ellison
D. Klick
P. Quantz
E. Vance
J. Weil

April 18, 1967

Mr. R. W. Bemer
Consulting Engineer
C-76

I have reviewed your proposed policy on the instrumentation work done by Messrs. Cantrell and Ellison. I concur with your proposal for extensive application by GE components. I do not concur with any plan for publicity at this time. I understand the inherent simplicity makes it difficult to consider Company confidential procedures, however, I do not wish to encourage competitor utilization.

It is clear that we must ultimately consider application for customer slave programs. The form of documentation and the procedure we will follow to introduce these techniques is not so clear. I have, for example, a question in my own mind as to the sales potential, e.g. your memo pointed out that 10-20% efficiency improvement could be expected.

I will discuss your proposal along with other alternatives with ISMO. In the meantime, I wish to take no specific action to release these techniques to other than GE personnel.

E. R. White
Manager - Engineering

/bp

SOFTWARE INSTRUMENTATION PACKAGE

P. Quantz
E. Vance
D. Klick
H. Cantrell
L. Ellison
J. Weil
J. Couleur

1967 April 5

TO: E. R. White

FROM: R. W. Bemer

I propose the following policy on the instrumentation work done by Cantrell and Ellison:

1) September 1 is a target date for:

   a) Release for customer usage
   b) Publication of technical articles
   c) Publicity

2) The tools are to be applied extensively until this date in order to:

   a) Gain sufficient experience
   b) Maximize efficiency of present software for competitive advantage
   c) Accumulate documentation, listings, techniques, etc., for input to technical articles.
   d) Package for customer usage.
   e) Coordinate with B-GE and O-GE to provide them with the tools/methods and simultaneous announcement.

This policy recognizes the natural evolution in the field of application from software master programs to software slave programs to customer slave programs. I envision an expenditure of perhaps 1% in customer installations to return from 10-20% additional efficiency by giving the user the instrumentation necessary to optimize the performance of his particular job mix. This would be a definite first in the industry and should be capitalized.

It is advisable to agree to a policy now, so that there may be no misunderstanding or losses. I have discussed this with Mr. Quantz and he concurs in the essentials.

Rough Draft
M. Grems
1967 April 3

USER-ORIENTED INFORMATION PROCESSING SYSTEM

TABLE OF CONTENTS

USER-ORIENTED INFORMATION PROCESSING SYSTEM

1. PROPOSAL

A critical examination of customer application programs that are written in any one programming language, such as FORTRAN, reveals common patterns of coding details that are continually repeated in different but similar-application areas. These common patterns seem better suited to the manipulations of a computer than to the intellectual pursuits of a human being. Obviously, the coding details can be handled by a computer routine.

It is proposed that GE develop a software package that automatically handles for the customer both the coding details that are normally repeated by the customer, and the corresponding documentation that is normally hand-written by the customer. This proposed package would offer guidance to the customer at the problem-solving level; i. e., one level above FORTRAN coding. The proposed user-oriented system would take advantage of the technical skills and experience of the user within his special area of interest. At the same time, it would produce better tools for the customer to use than the present systems produce, and would require fewer computer manuals for the customer to study.

The proposed system can be adapted to either the GECOS type or BASIC type of operating system, and can be offered to the customer as an extension of either system.

The proposed User-Oriented Information Processing System (USOR) consists primarily of a Customer Handbook, a USOR Processor, and USOR Files that operate in conjunction with an existing operating system.

2.0 CUSTOMER HANDBOOK

A Customer Handbook is a document that can be distributed to the outside customer to replace the technical reference manuals that are currently being distributed. This document is composed of three separate sections: Customer Instructions, Application Programs, and Indexes.

The first section, Customer Instructions, contains instructions to inform the customer on how to use the system. The information is arranged as a sequence of steps that include specific details regarding: the locating of data in the Handbook; preparing original input data; typing at the console; responding to computer requests; and anticipating printed output results.

The second section, Application Programs, contains a collection of entries to describe the application programs that are available to him via the system. These entries include not only the conventional application programs, but also parts of programs (subprograms), and general methods for solving certain types of problems on a computer.

The third section, Indexes, contains a number of different indexes to the entries. The various entries are arranged by categories in the same application areas, by categories of different computer methods, by keywords for macros, by keywords for problem conditions, and by combinations of categories.

2.1 CUSTOMER INSTRUCTIONS

This section informs the customer how to use the system. In essence, the customer searches the Handbook to find a comparable program for his problem, matches its optional conditions with the requirements for a solution; uses a codename as input at a console; and then continues to respond to all computer requests either for data symbols and values, or for yes/no answers.

1) The customer searches the indexes in the Handbook to find a program or method that corresponds to the needs of the problem at hand.

   e.g. SIMULTANEOUS LINEAR EQUATIONS

2) The customer reads the descriptive material for that entry, and verifies that it is suitable and adequate. He records the macro that identifies this entry for use later.

   e.g. SIMEQ.

3) The customer studies the optional condition in the option-chart, and selects an option# whose conditions match his own requirements and conditions. He attaches this option# to his macro, and thereby forms a two-part codename to be used later as console input.

   e.g. SIMEQ.2

4) The customer records (either mentally or on paper) the required Handbook symbols that are shown for this selected option#. At a later time, the computer will expect him to supply a corresponding symbol and value of his own choice for each of these Handbook symbols.

   e.g. A1, A2, B1, B2, A, B, X1, X2

5) The customer prepares input data for each required Handbook symbol. This consists of a Customer symbol equated to an assigned value. The Customer symbol is a name or label that is any combination of not more than six characters, excluding the equal sign character. (The equal sign character is used to separate the Customer symbol from his assigned value.)

6) The assigned value is the data value that the customer wants assigned to that symbol for his computation. Later, at the console, the data values are typed in a way that is convenient and natural for the customer; except when special instructions are given in the Handbook. Most frequently, data values are written as 1.576 or .25 or 1237.45. However for engineering notation, the values are written as 1.2375 x 10+3 or 1.278 x 10-6; or written in conventional programming notation 1.278E-06.

7) Special situations for assigning Customer symbols and values are handled as follows:

When the Handbook symbol is satisfactory as a Customer symbol, the customer repeats the symbol.

When the value of a symbol is to be computed, the customer assigns a question mark as its value.

When a value for the same Customer symbol was assigned or computed in a previous macro, the customer assigns an asterisk as its value.

When the value to be assigned is non-mumeric, the customer can assign any character string (including but not beginning with the equal sign) as its value.

```
e.g.  TM =  .25
      A  =  *
      X  =  ?
      Z  =  TRUE
```

8) The customer also prepares at this time any narrative information (such as a header or comment) that he wants printed along with the equations as part of the problem documentation. This is referred to as a header. A header can be a character string of indefinite length, is not translated, and is not part of the computation. The header is merely copied into and out of the computer store exactly as the customer types it. A header is accepted into the system only when it accompanies the codename for the entry.

e.g. SIMEQ.2 SOLVE FOR VALUES OF X AND Y

9) At various times, the customer must be ready to respond with a YES or NO answer to questions that the computer presents about his data. Most often, these questions occur when the computer is waiting for additional data.

10) The customer is now ready to put his problem on the computer. The customer signs on at the console in the customary way and asks for USOR. Unless indicated otherwise, USOR assumes that a one-shot one-macro problem is to be checked out, and assumes that default conditions (for printing and saving the program) are implied. Other conditions are indicated by control words that are similar to the control data for GECOS.

11) The customer types the two-part codename for the macro and option#, and follows the codename with his header information.

e.g. SIMEQ.2 SOLVE FOR VALUES OF X AND Y

12) At this point, the customer waits for the computer to find and print the Handbook equations (or formulas or statements) that correspond to his selected macro and option conditions. The Handbook symbols are used in these equations.

e.g. $A1\ X1 + A2\ X2 = A$
$B1\ X1 + B2\ X2 = B$

13) The computer then prints each Handbook symbol followed by an equal sign. The computer expects the customer to respond with his Customer symbol, an equal sign, and his assigned value.

e.g.

computer printout	customer response
A1 =	TM = .25
A2 =	R = .50
B1 =	S = 1.0
B2 =	M = *
A =	A = *
B =	B = *
X1 =	X = ?
X2 =	Y = ?

14) The customer waits for the computer to execute his program. Then the computer prints all the information it has both collected and computed for this problem. This information includes the tally#, code-name, customer header, equations with customer symbols, and customer symbols with two sets of values (i.e., assigned values and computed results).

e.g. #1 SIMEQ.2 SOLVE FOR VALUES OF X AND Y

TM X + R Y = A
S X + M Y = B

	assigned value	final value
TM	0.25	0.250
R	0.50	0.500
S	1.0	1.000
M	*	0.500
A	*	3.525
B	*	2.475
X	?	8.000
Y	?	11.050

15) The customer signs off, continues with another problem or macro, revises the current problem, etc.

2.2 APPLICATION PROGRAMS

This section of the Customer Handbook is a collection of entries for application programs. Each entry describes a separate program, part of a program (subprogram), or a computer method of solving general types of problems on a computer. Each entry consists of descriptive and technical information, followed by an option-chart for special and exceptional conditions. Machine coding to correspond with each entry is stored in the USOR Library File.

The information for each different entry in this section is arranged in the following format:

Macro Title

Application area, category, or method

Narrative descriptive information

General equations, formulas, and statements

Special definitions and symbols

References to published articles and textbooks

OPTION-CHART

option#	special and exceptional conditions	required Handbook symbols

2.3 INDEXES

(To be filled in later)

3.0 USOR PROCESSOR

The USOR Processor is a computer program that monitors, controls, and executes the system functions; generates individual Problem Program Blocks; and retrieves data from the previously constructed USOR Files. The USOR Processor is described here in terms of an Operating Procedure and Program Blocks.

The Operating Procedure is explained as a series of steps that illustrate the actual performance of the system functions, including retrieval, conversion, symbol replacement, program generation, file maintenance, problem documentation, etc.

The Program Blocks are four blocks of data that are assembled during the execution of the USOR Processor. These four blocks constitute the computer program for a specific problem, and include the Sequence Block, Output Block, Symbol Block, and Coding Block.

3.1 USOR OPERATING PROCEDURE

The USOR Operating Procedure assumes that the software operating system has previously translated any control words (such as list, deck, compress, etc.) to flags, and then called in the USOR Processor with the USOR Files. The USOR Files are permanent files and include the Directory File, Interface File, Abstract File and Library File. The Program Blocks belong to a particular problem and include the Sequence Block, Output Block, Symbol Block, and Coding Block.

1) Store the job#, the flags, and a tally of zero in the Directory File. For a repeat problem, find the job# in the Directory and continue from there.

2) Initiate a Squence Block and store its pointer in the Directory.

3) Initiate three more data blocks: Output Block, Symbol Block, and Coding Block. Store their pointers in the Sequence Block as ptr4, ptr5, ptr6.

4) Establish a working directory for the next-available locations in the three variable length blocks: Output, Symbol, and Coding Blocks.

5) Add +1 to the tally in the Directory and store in the Sequence block.

6) Accept the input character string for the codename and header (m. o, header), isolate the codename, and find its match in the Interface File.

7) Retrieve the corresponding interface data, and store it as the first part of the first row in the Sequence Block. (This data consists of the m. o and 3 pointers; ptr1, ptr2, ptr3.) The first row in the Sequence Block now contains: tally#, m. o., ptr1, ptr2, ptr3, ptr4, ptr5, ptr6.

8) Copy into the Output Block, the tally # from the Directory followed by the image for the input character string; i.e., +1, m.o, header.

9) Get ptr1 from the Sequence Block, and apply it to the Abstract File in order to retrieve the abstract record for this codename.

10) Print an abstract for the customer to examine, including the Handbook macro, Handbook title, and Handbook equations for this option.

11) Print separately, each Handbook symbol followed by an equal sign, and wait for the customer to respond with his Customer symbol and assigned value.

12) Build a new row in the 6-column Symbol Block for each Handbook symbol. In the first 3 columns, copy the coding symbol, conversion data, and Handbook symbol from the Abstract File. In the next 2 columns, store the Customer symbol and assigned value. In the 6th column, store a blank space to be replaced later by a final result.

13) Convert each assigned value to the internal form as indicated in column 2 of the Symbol Block, before that value is stored in column 5.

14) When a series of macros must be assembled and computed as one problem, a Patchcode block may be needed to tie a previous Coding block with the next adjacent Coding block. The Patchcode block is then inserted at the beginning of the next adjacent Coding block. For instance, asterisk values are filled in by a patchcode. Previous Symbol blocks are searched until a matching Customer symbol is found; machine instructions are generated that will move the corresponding final value for that match to the current Symbol block; and the generated machine instructions are stored in the Patchcode block.

15) Execute the symbol-replacement routine that is included in the abstract record. This routine modifies the Handbook equations by replacing the Handbook symbols with the Customer symbols.

16) Store the revised equations in the Output block. The Output block now contains: #1, m.o, header, and customer equations.

17) Get ptr2 from the Sequence Block, and apply it to the Library File. Retrieve the record of FORTRAN coding that is stored for this codename.

18) Execute the machine instructions found in the record. These instructions use data from the Symbol block to modify a set of skeleton FORTRAN statements (such as I/O and format statements) that are part of the coding.

19) Move the modified coding to the Coding block.

20) Get ptr3 from the Sequence Block and apply it to the Languages File.

21) Use the language processor to execute the Coding block. Move the final values to the 6th column of the Symbol block.

22) Print the Output block, and print columns 4-6 from the Symbol block.

3.2 PROBLEM PROGRAM BLOCKS

For each different problem, the USOR Processor constructs four separate program blocks, referred to as the Sequence block, Output block, Symbol block, and Coding block.

The Sequence block consists of a row of data for each macro in the problem. Each different row contains a tally# and codename for identification, and 6 pointers for locating the corresponding data to be retrieved. Pointers 1, 2, and 3 point to the Abstract File, Library File, and Languages File. Pointers 4, 5, and 6 point to the Output block, Symbol block, and Coding block.

The Output block contains one variable length record for each macro in a problem. The information in each record is the problem description for that macro. It is collected gradually by the processor and then modified to fit the requirements of the customer. This information consists of the tally#, a codename for macro. option, a header if available, and the equations with Customer symbols substituted for Handbook symbols.

The Symbol block contains a variable length record for each macro in the problem. Each record contains a description of the required symbols that need to be identified for that macro. A record consists of the following data for each required symbol: Coding symbol, conversion data, handbook symbol, customer symbol, assigned value, and final value.

The Coding block contains one variable length record for each macro in a problem. This block of coding consists of the actual instructions that are executed in conjunction with the Symbol blocks for the solution of the problem. Specifically, each record contains the modified FORTRAN coding for that macro, along with any patchcode instructions that are needed to modify the data.

4.0 USOR FILES

The USOR Files are data files that are called in and used by the USOR Processor. These files are permanent files that belong to the processor, and have been previously prepared, assembled, and stored. The USOR files include the Directory File, Interface File, Abstract File, and Library File.

4.1 DIRECTORY FILE

The Directory File contains data that identifies the customer problems that are stored as computer programs. For each different customer problem, the processor stores one row of data in the Directory. This row of data consists of the job#, flag codes, highest tally#, and a pointer to the Sequence block. The job# is self-explanatory, and the other data items are explained briefly.

The flag code is a bit-pattern to inform the USOR Processor how to handle certain physical conditions for this problem, such as how the program blocks are stored, what documenting is required, what printing is suppressed, etc. The information for the flag code was originally introduced by the customer at the beginning of his problem, and remains in effect until he decides to change it.

The tally# is a consecutive number that is assigned to each new macro in a problem. It is used to uniquely identify that macro in a problem, and a tally# is never repeated in a problem. The tally# is stored for identification along with the codename in the Sequence block. At some later time when an insertion or deletion of a macro is necessary, the tally# taken from the problem description printout is the only means of locating a particular macro in the overall problem.

The pointer is the last item in the Directory. This data item points to the beginning of the Sequence block that corresponds to the job# for a customer problem.

4.2 INTERFACE FILE

The Interface File contains interface data to tie a handbook selection to the USOR files. The interface data lets the USOR Processor find information in its USOR files that corresponds directly with the descriptive material that the customer found previously in the Customer Handbook.

The Interface File consists of one row of data for each macro. option combination. In turn, each row in the file contains a codename and pointers to three other files for that codename. Each of the three files contains a specific kind of data for the codename, including abstract data, library data, and languages data.

4.3 ABSTRACT FILE

The Abstract File contains abstract data for the Application Programs that are described in the Customer Handbook. The abstract data is written in a general form, so that the same abstract can be used for several different macro. options.

Each different set of abstract data consists of a macro name, the corresponding Handbook title, general equations with Handbook symbols, a 3-column table of symbol data, and a set of machine instructions. The machine instructions are used to modify the general equations so that the equations printed for problem documentation reflect the data, requirements, and conditions introduced by the customer.

4.4 LIBRARY FILE

The Library File is an internally stored collection of individually identified records of source coding, such as FORTRAN coding. This coding can be written in any of the common programming languages for which the operating software system has a language processor. Instead of a large library of stored programs, the Library File is a relatively small collection of records that perform the general functions that are found in a large collection. In essence, the Library File is the power of the system that permits the "automatic programming" of a solution to a customer problem.

Each Application Program described in the Customer Handbook refers to a record in the Library File (or to more than one record by indirection) via the interface data in the Interface File. However, it is important to point out that there is not a one-to-one correspondence between Handbook Application Programs and Library File records. One record in the Library File may satisfy the requirements of multiple different application programs and computer methods.

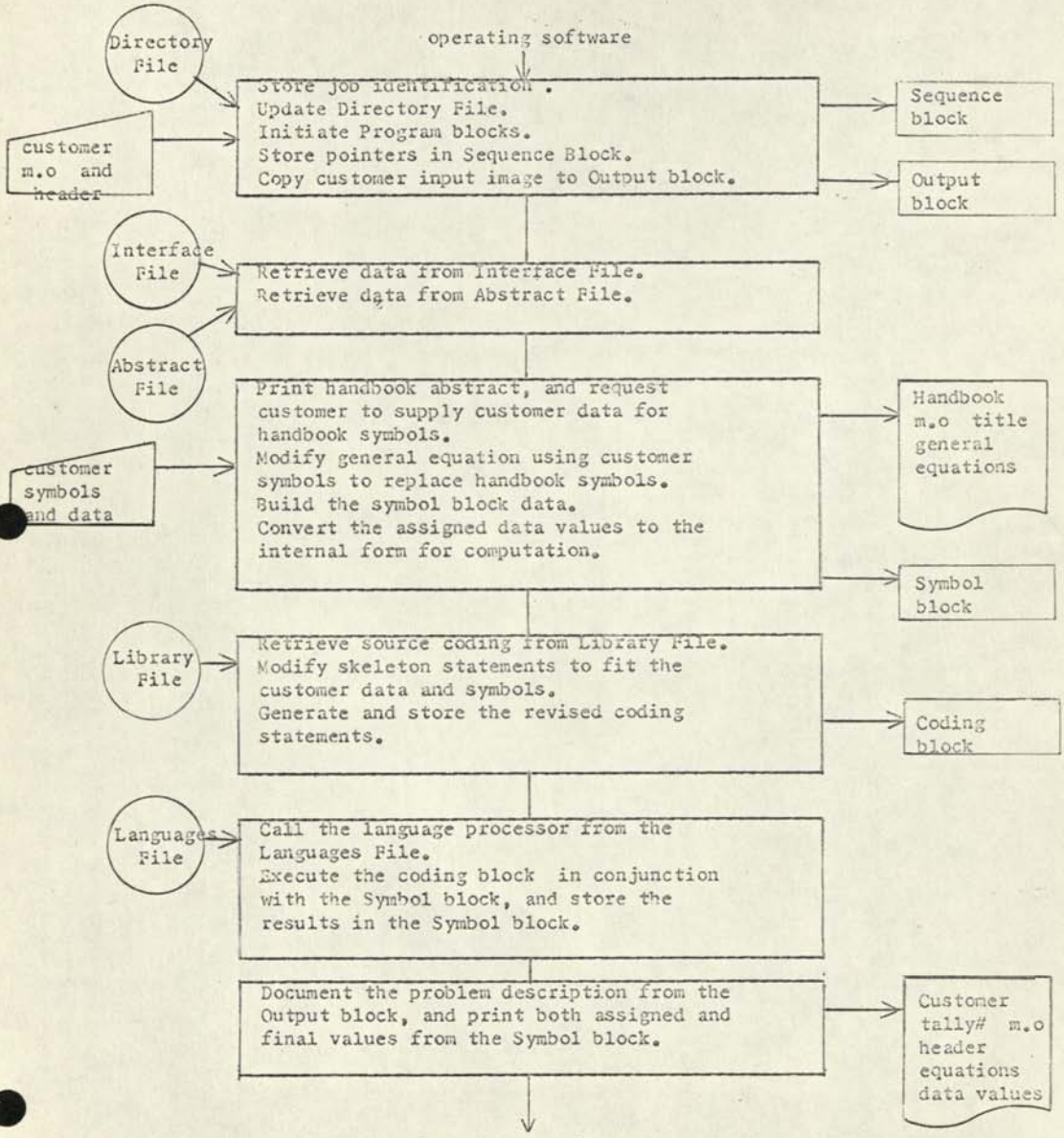
Each record in the Library File consists of an identification, skeletal coding statements, actual coding statements, and machine instructions to modify the skeletal coding. For the most part, the skeletal coding statements are I/O, format, and dimension statements into which the customer symbols and data values can be inserted.

The Library File is comparable to the conventional programming library of utility routines and subroutines, but is applied at the next higher level in the problem-solving process.

It is obvious that application programs can be added to the Customer Handbook, and the corresponding interface data to the Interface file, without adding any new records to the Library File.

5.0 DIAGRAMS & CHARTS

5.1 BLOCK DIAGRAM OF USOR PROCESSOR



5.2 SYSTEM FILES

Directory File			
job#	flags code	highest tally# for this job#	pointer to Sequence block for this job#

Interface File			
codename for macro.option	ptr1 to Abstract File	ptr2 to Library File	ptr3 to Languages File

Abstract File
Handbook macro, Handbook title, Handbook equations. Machine instructions to modify the general equations to fit the option conditions and the customer data. coding symbol conversion data handbook symbol

Library File
ID# Skeleton FORTRAN statements. Actual FORTRAN statements. Machine instructions to modify the skeleton FORTRAN statements.

5.3 PROGRAM BLOCKS

Sequence block			
tally#	codename	pointers to 3 files	pointers to 3 blocks

Output block			
tally#	codename	header	customer equations

Symbol block					
coding symbol	conversion data	handbook symbol	customer symbol	assigned value	final value

Coding Block		
ID#	patchcode instructions	FORTRAN coding statements

GENERAL ELECTRIC
DEFENSE PROGRAMS OPERATION
TECHNICAL MILITARY PLANNING OPERATION
735 STATE STREET, SANTA BARBARA, CALIF.

DIAL COMM
8*432-6

SUBJECT

1967 March 9

TO: R. W. Bemer

cc: J. C. Fisher
H. R. J. Grosch
Dr. J. Weil

FROM: M. Grems

SUBJECT: User-Oriented Information Processing

Enclosed is additional material for the previously-proposed User-Oriented Information Processing System.

The USOR Operating Procedure briefly outlines the steps for a system as seen from the customer viewpoint. The block diagram shows the same flow of information.

The USOR Files and Program Blocks illustrate the contents of and organization of data that must be prepared ahead of time. These files and blocks illustrate data and techniques as seen from the systems programmer viewpoint and internally by the operating system itself.

Obviously, the collecting and especially the organizing of the data for the USOR Files is not a trivial undertaking. It is anticipated that different macros can select the same block of coding from the Library File. Of course, this assumes that the coding symbols for these macros must be identical, while the Handbook symbols for the macros may be different.

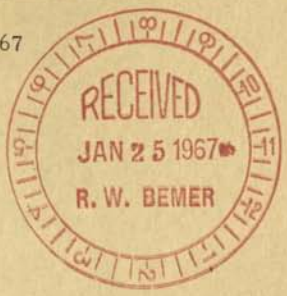
M. Grems

GENERAL ELECTRIC
DEFENSE PROGRAMS DIVISION
TECHNICAL MILITARY PLANNING OPERATION
735 STATE STREET, SANTA BARBARA, CALIF.

DIAL COMM
8*432-6

SUBJECT

January 23, 1967



Mr. R. W. Bemer
General Electric Company
13430 North Black Canyon Highway
Phoenix, Arizona 85023

Dear Bob:

Enclosed is a copy of a proposal that describes very briefly some of my ideas for a User-Oriented Information Processing package. I realize that certain portions of the proposal are hazy, but hope it is adequate for a first examination of the overall plan.

You will probably notice that the time-consuming part of the package will be the development of the Handbook and Library, both of which must be stored in the computer and should be prepared by a computer program. The Handbook will be used by people, while the Library will be used by other computer programs. Therefore, I believe that the Handbook should be large and expansive, with possibly the same subprogram described in different parts of the Handbook for different problem areas. Each duplicate copy could have different variables and different emphasis on its use. On the other hand, the blocks of source code in the Library should be compact and condensed with all detected duplications removed. (This will make it easier to build the same Library for several different computer hardware systems).

Of course, a large Handbook and small Library assumes a well-structured interface section for 'Handbook to Library' and 'Library to Handbook'. The interface section would have to accommodate many Handbook subprograms by means of a relatively few Library blocks, and could be handled by an internal synonym table or internal thesaurus (black box). It all fits into a general pattern, doesn't it?

Do you know if anyone else has attempted this approach? If so, what happened? It sounds exciting to me.

Sincerely,

Mandy

M. Grems

cc:
Dr. J.C. Fisher
Dr. H.R.J. Grosch

USER-ORIENTED INFORMATION PROCESSING

or

COMPUTER-AIDED PROBLEM SOLVING

INTRODUCTION

It appears that the computing industry has reached the stage in computer software development that it is ready for the next step toward user-oriented information processing. The next step can be accomplished at a level just above the common programming languages in current use, such as FORTRAN, ALGOL, COBOL, JOVIAL.

For GE, information processing at the level discussed in this proposal can fit into either the GECOS or BASIC systems, and can be introduced to the GE customer as a natural development or outgrowth of the present systems. For the rest of this proposal, any programming language for which GE has a processor can be substituted for the name FORTRAN.

After examining a number of FORTRAN programs, it is apparent that many coding details are repeated over and over again. These details are routine operations that are better suited to the manipulations of a computer program than to the intellectual pursuits of skilled and highly trained human beings. Obviously, these details can be handled by internal routines that pass the data back and forth among a variety of

routines for IR, LP, convertors, translators, decision tables, data charts, black boxes, etc.

e.g., The GE-625/635 FORTRAN IV Manual lists 284 error comments for "Processor Diagnostic Error Comments". An examination of the 284 comments illustrates an overwhelming abundance of errors due to coding details that are imposed by the FORTRAN language, and relatively few comments that pertain to the needs of the problem being solved.

The proposal described here suggests a USer-ORiented computer package. The proposal very briefly describes a Language, Handbook, Library, and Processor. It shows the overall picture for such a package, and how the big blocks fit together. It pinpoints the advantages to be gained from offering more guidance and better tools to the customer at the problem-solving level, and at the same time offering less information and fewer manuals at the system maintenance level.

This package can be considered an extension or outgrowth of any one of a number of previous systems. In particular, it combines and coordinates some of the salient features (from a user viewpoint) of:

GECOS, BASIC, "Programming by Questionnaire", BACAIC, ACM JUG-CAD, SIMSCRIPT, and the SHARE DIRECTORY.

PROPOSAL

It is proposed that GE design and develop a USer-ORiented computer package that can be implemented to fit into an existing Operating System. Hereafter in this proposal, the proposed package is referred to as USOR. USOR would be similar to the conventional assembly package, except that the computer user applies USOR at a higher level of problem solving. The computer user can be a customer, an analyst, an application programmer, or a systems programmer, and hereafter in this proposal will be referred to as the user.

USOR is composed of a USOR Language, a USOR Handbook, a USOR Library, and a USOR Processor. Each of these are described briefly.

USOR Language

Instead of encouraging a prospective user to learn to program according to the rules as defined for the current programming languages, why not encourage him to state his problem (to be solved on a computer) as a sequence of steps that closely resemble his own description of a solution to his problem? In the USOR language, he writes each step as two parts, in which the first part, MACRO, OPTION, tells what-to-do; and the second part, DATA ELEMENTS, gives the names of the data items.

e.g., MACRO, OPTION DATA ELEMENTS

In the first part, MACRO.OPTION, the MACRO is a mnemonic name (or I D#) for a subprogram that is described in the USOR Handbook. By means of indexes and abstracts in the Handbook, the user locates the description of a subprogram whose basic function fits the needs for his problem solution. Initially, this will be a manual search and lookup from a big book. Later on, however, automatic searching techniques can be incorporated via a console. For the Handbook, a subprogram is defined to mean one or more blocks of source code that accomplish a particular function, and can be identified by a mnemonic or ID#. The function may be a method, technique, routine, formula, table, arrary, or any combination of these. Incidentally, the block of source code is stored in the USOR Library and must be written in a programming language that the existing Operating System can process, and the Handbook can be prepared by a computer program.

The OPTION is a code letter to indicate a specific set of conditions for that subprogram. The user chooses the code letter to represent the set of conditions in the Handbook that corresponds to the conditions required for his problem solution. It is possible that occasionally a set of conditions not previously considered by the user might be presented in the Handbook for his consideration.

In the second part of the program step, DATA ELEMENTS, the user is provided with a means to identify a variable (from the Handbook subprogram) by a data element name of his own choice, and to assign to that variable a value, quantity, or number. The user does this by equating the variable to a Data ELEMENT name, and in turn, equating this to an assigned value.

e.g. Refer to the BASIC program on page 2 of the "BASIC Language Reference Manual". In this case, a user has a problem in which one program step needs the X and Y value for two simultaneous linear equations. By means of the indexes in the Handbook, the user finds a subprogram "SIMEQ" listed under simultaneous equations. The SIMEQ page in the Handbook contains the following description:

SIMEQ Simultaneous Linear Equations

Function - - - - -
- - - - - - - -
- - - - - - - -

Option Code	Conditions		Required Variables
.			
.			
H	AX+BY=C	2 equations	A=
		2 variables X, Y	B=
	DX+EY=F	constant coefficents A, B, D, E	D=
		variable coefficients C, F	E=

On the USOR program sheet, he writes

MACRO. OPTION	DATA ELEMENTS
SIMEQ. H	A=1
	B=2
	D=4
	E=2

(Note. When the USOR program is executed, the equations showns for Option H will be printed along with the assigned and computed values. This information should be sufficient to satisfy the documentation needs of the user).

USOR Handbook

Instead of distributing to the outside customer the voluminous reference manuals and documents that describe the technical operation of each computer and each piece of software, why not distribute to the customer a USOR Handbook that he can understand immediately? USOR Handbook is in two parts. Part I contains the Indexes for the subprograms that are stored in a USOR Library, and PART II contains an abstract or description of each subprogram.

In Part I, one of the indexes probably will be a KWIC index of the titles of all the subprograms. For this Handbook, the titles will be written in technical terms to express the purpose of the subprogram

rather than in glamorous Madison-Avenue terms. Additional keywords can be appended to the title whenever desired. Another index that is useful to the customer is a subject index (prepared on a computer, of course) of categories, synonyms, and data base names.

In Part II, the abstract for each subprogram is written in a uniform way so that specific information for a subprogram can be copied easily and accurately by the reader to his USOR program. The abstract consists of: ID#, Title, Function, and Options. The ID# is the mnemonic for the title, the Function states and explains the formulas, methods, techniques, tables, etc. for the subprogram; and the Option includes the information about all the sets of conditions that are available for that subprogram along with the code letter and required variables for each set.

The USOR Handbook can be developed as rapidly as the source code blocks can be prepared for the subprograms (and options) that must be stored in the USOR Library. The Handbook will include new subprograms and options whenever they can be made available. It will not include proprietory information nor reflect every trivial change in the source coding.

The Handbook can be issued periodically and might eventually replace the Technical Documents that are now being distributed. The present Technical Documents contain proprietory information that the manufacturer might want to retain for his own use. (Improvements to the Library source code do not need to be reported to the customer. This practice is really admitting that you didn't do a very good job in the first place).

USOR Library

Instead of trying to collect a large library of stored subprograms, why not try to accomplish the same functions with a small library? The USOR Library is an internally stored collection of individual blocks of source code and can be built up and improved gradually. Each block is identified by an ID#, Option code, and control data. The ID# and option code have already been discussed. The control data for each block provides the means for replacing the variables with Data Element names and values; for detecting the need for data transformation and connectors; as well as all hardware, software, and data requirements. The format for the control data needs to be carefully planned, as the control data is the key to all internal interfaces, data movement, and program generation.

e.g. In a FORTRAN source code, the control data would include the statement numbers for any READ, WRITE, PRINT, and PUNCH statements, and possibly for FORMAT statements. These statements in the source code could then be replaced or amended as required by the Option and Data Element in the user USOR program.

(Note. This area needs a lot more study and thought, especially regarding the internal standards, formats, criteria, and check-lists for organizing the blocks of source code. I have lots of ideas, but they do not all fit together yet. For instance, we need a Software Handbook for systems programmers. If we had one, I believe that new software could be developed more quickly. We need an "organizing routine" whose main function is to examine, review, and evaluate each block of source code that is offered for storage, so that duplications are discarded and not accepted.)

USOR Processor

1. Let GECOS read the USOR program cards as if they were FORTRAN source cards, and then give the character string to the USOR processor.
2. For each program step in a USOR program, the processor matches the combined MACRO and OPTION to select the appropriate

block name and control data from the stored library.

3. The processor then uses the name and control data to build a sequence table or possibly a decision table. The sequence table will include references to indicated source code, and to the needed connectors, joiners, converters, etc. The table will also include means for replacing the variables with DATA ELEMENTS.
4. When a variable is missing, or when the data representations that are required to interface from one subprogram step to the next subprogram step are not compatible, the processor prints a description of the missing data.
5. When the data are satisfactory, a machine program can be built in the form of an execution table.
6. The execution table can be handled the same as any other program in GECOS, that is: RUN, SAVED or CONTINUED.
7. Unless instructed otherwise, when a program is RUN, it also prints a brief report of each original USOR step.
8. A user can request that only certain abstracts or results are printed in his final RUN.

CONCLUSIONS

When COBOL was designed, it was anticipated that COBOL would offer this kind of customer service for business problems. However, COBOL became bogged down in all the detail operations that the committee members could think of, and then these details were imposed upon the language itself. Even at that period of time, some of those details had already been computerized, but the committee either was not aware of the advancements, or else could not get general acceptance of the concepts.

GE customers would probably be willing to pay for a service that minimizes their programming and coding details. They could then concentrate on improving their own methods for problem-solving, and de-emphasize their employee training for computer programming (for details imposed by the software). The continued appearance of new special problem-oriented programming languages indicates that many customers are groping for such a service.

It is also possible that this service would open up new markets for computer usage. Consider what happened to document reproduction when Xerox introduced dry copying techniques by means of the expensive Xerox 914 Copier.

REFERENCES

ACM-JUG Ad Hoc Committee
"Proposal for an ACM-JUG CAD, Computer Applications Digest"
Submitted to JUG Executive Committee and ACM Council, April 1965.

Chapman, Ansom E,
"DETAB-65 Preprocessor"
Share Program Library, SDA 3396, Jan. 1966

Childress, C.
"DART - Dynamic Automated Reporting Technique"
General Electric Co., Pacific Range Eng. / RGO, 64-SPC-3

General Electric
"BASIC Language Reference Manual" IPC-202026, May 1966
"GE-625/635 Comprehensive Operating Supervisor" CPB-1002A, Jan 1965
"GE-625/635 FORTRAN IV" CPB-1006C, Jan 1966

Grems, M.
"Problem Description Steps for ADP Applications"
Univac, Sperry Rand Corp., Sept 1963

Grems, M., Porter, R.E.
"A Truly Automatic Computing System - BACAIC"
Boeing Airplane Company, WJCC Proceedings, Feb 1956

Ginsberg, Allen S., Markowitz, Harry M., Oldfather, Paula M.
"Programming by Questionnaire". The RAND Corp. Prepared for U.S. Air Force Project Rand RM-4460-PR, April 1965. DDC 8D-613976

Hyman, R., Anderson, B.
"Solving Problems"
International Science and Technology, Sept 1965

Kaplow, R., Strong, S., Brackett, J.
"MAP - A System for On-Line Mathematical Analysis"
Massachusetts Institute of Technology, MAC-TR-24, Jan 1966

Karr, Herbert W., Kleine, Henry, Markowitz, Harry M.
"SIMSCRIPT 1.5"
California Analysis Center, Inc. C.A.C.I. 66-INT-2 Oct 1966

Markowitz, H.M., Hausner, B., Karr, H.W.
"SIMSCRIPT: A Simulation Programming Language"
The RAND Corp., Prepared for United States Air Force Project RAND RM-3310-PR, Nov 1962

SHARE Program Library and SHARE Directory

ID#: 67.1
DATE: 1967 March

PROPOSAL ABSTRACT FORM

SUBJECT AREA: Information Processing Systems

CATEGORIES: Online application programming, programming language, language processor

GE address & tel.:
Box QQ
GE TEMPO
Santa Barbara, Calif.
8-432-6431

RESEARCHER: Grems, M.

PROJECT TITLE: User-Oriented Information Processing - USOR

LONG RANGE SCOPE: An information processing system for solving computer applications online as input to the system that uses only customer data with a Handbook code for the method. A possible solution is automatically programmed and documented that corresponds to the conditions selected from the Handbook.

IMMEDIATE OBJECTIVE:
Assemble a Handbook of Application Abstracts for problems in one selected area.
Build a Library File of basic coding for applications in that same area.
Establish standard codes for characteristics and attributes of data items.
Build a prototype USOR Processor that accepts Handbook codes and basic coding for processing an application, and for automatically documenting a summary of the problem solution.

NEW CONCEPTS, DESIGN, IDEAS and TECHNIQUES:
The recognition of duplicate coding in apparently dissimilar applications and problem solutions, and thereby reducing the size of the conventional Library File of basic coding.

The substitution of Customer Symbols for Handbook Symbols, and thereby automatically documenting a problem description and solution from the customer input data.

The substitution of Coding Symbols for Handbook Symbols, and thereby automatically programming the selected option conditions into the basic coding as found in the stored Library.

STARTING POINT and PREREQUISITIES:
Computer Hardware with a working Operating System that includes a language processor such as FORTRAN.
A Program Library of customer applications for study.
Coding for the applications written in the above language.
Utility routines for data conversion.
A limited application area for detailed study.

WORK PLAN FOR STUDY, METHOD, & DEVELOPMENT:

ANTICIPATED First Week	ANTICIPATED Elapsed # Weeks	
	1	Describe the steps a customer takes to solve his problem.
	7	Outline the detailed steps of an operating procedure for a computer processor to handle the same problem solution.
	4	Define the format and contents of data files that are prepared for the processor previous to execution of a problem.
	4	Define the format and contents of data files to be built by the processor during execution.
	2	Examine each step of the operating procedure to detect conflicts, errors, and inconsistencies; and then develop the necessary diagnostics, checkpoints, and interfaces.
	2	Verify and validate the data files to remove or resolve conflicts in format or content.
	4	Collect and study existing material for data characteristics.
	4	Establish standards for suitable data characteristics.
	4	Study a number of similar applications from the Program Library to detect patterns of logic.
	4	Restructure the applications into components of logic blocks.
	4	Study components to determine what conditions might be suitable for options.
	8	Segement the corresponding FORTRAN coding into blocks of basic coding that fit the components.
	4	Rewrite the hardcopy descriptions of the applications to fit the suggested Abstract form.
	12	Prepare the Abstracts and FORTRAN basic coding as part of the internal files for the USOR processor.
	20	Implement a USOR Processor to handle customer data and USOR files, and to execute the operating procedure as outlined.
		Study additional applications to detect duplicate basic coding, so that Handbook Symbols and Coding Symbols can be made synonymous. Continue to prepare Abstracts and basic coding for the files.

PROPOSAL ABSTRACT FORM (continued)

REQUIRED RESOURCES — ESTIMATES ONLY

Manpower

grade level	# of persons	manweeks/person
9 - 12	3	52 weeks
13	1	52 weeks

Equipment

function	machine #	machine-hours/machine
process	GE-635 with Operating Sys.	500
Input/Output	console	on demand

Printing

document type	audience	pages/copy	# of copies
Trial Handbook	customer	100	100
USOR Ref. Manual	systems prog.	100	10
Final Research Report	management	30	5

EXPECTED OUTPUT & RESULTS:

A Customer Handbook of Computer Applications.

A USOR Processor to automatically program and document a problem solution that fits the problem condition to be selected by the customer for an applicaton.

A Library File of basic coding for applications that uses common coding symbols as data elements.

Standard codes for characteristics, dimensions, and attributes of data items.

PREVIOUS RESEARCH CONTRACTS

By the researcher:

SIMILAR PROJECTS:

GENERAL ELECTRIC
DEFENSE ELECTRONICS DIVISION
SPECIAL INFORMATION PRODUCTS DEPARTMENT

SUBJECT

DIAL COMM
8*256-1813

COPIES GA Hoyt
LC Maier
EF Roache

R. B. CURRY
RECEIVED
MAY 13

Building #5 Room C-6
Court Street Plant
Syracuse, New York
May 12, 1966

Dr. Louis T. Rader
Vice President and General Manager
Information Systems Division
2000 Holiday Drive
Post Office Box 909
Charlottesville, Va. 22901

Dear Lou:

I mentioned to you a couple of weeks ago that we had an appropriation in the mill to extend our SIPD computer facility. The appropriation is for slightly over one million dollars, and I told you that the Services Review Board were concerned about integration with ISD and with making sure that we didn't duplicate any ISD existing or planned capability.

In accordance with your suggestion we discussed the appropriation request with Len Maier, and then in response to his suggestion, discussed it further with John Weil, and established that our facility plans would not duplicate facilities planned for ISD.

Since this appropriation request will be coming up for review at the Executive Office late this month, and since some members of that Office might inquire of you about the request, I thought you might like to know the use to which the facility is planned to be put. There are two:

One - to permit SIPD to do the hardware and software development, design and verification for military programs such as AWACS, MOL, ADSAF, NMCS and ANEW

Two - to permit SIPD to do similar development, design and verification for real time applications we have and have planned. We have programs with Martin and Hughes, as well as work with FSO to support ISD schedules on the Huntsville, RADC and NSA programs. We expect to use the facility on the 615 program, when we find the key to getting it off the ground, and the Advanced Computer Development Program a little later in its schedule.

I'm sure you are well aware of the difficulty in getting business and performing to contracts without adequate facilities. This particular appropriation should bring our facility up to the level required for today's systems needs.

Very truly yours,

John F. Burlingame

John F. Burlingame
General Manager
Special Information Products Department

JFB/jm

SIPD - FOS

Modified 605

New GECOS + 1

RTIOC

PRO	CON
Meets initial closeout?	Not 100% Compatible
Based on MSD job	Likely very inefficient when upgraded
	+
	Same as other SIPD proposal

SIPD - PHX

Modified M605
New GECOS
IOC-B - 20% > S.C. than PHX

PRO	CON
Design uses fewer ckts	Upgrade requires more swaps
Upward compatible	No shop cost advantage next 2 yrs
Smaller configuration	$941K one-time charge 2 ~~yrs~~
	Still an interim
	Qty limited by peripheral availability
	Non-std ckt set (parts + training cost)

Need

Limited Offer
to key prospects

PHX

Detumad 625
GECOS identical to 625/635
IOC - B

PRO	CON
Available	Cost
No risk of new GECOS	Requires 625 configuration
100% ↑ compatible 625/635	Still Interim
Upgrade with minimum swapout	Periph-limited production
All DDGM's favor	
Nothing new for training & parts (FEO & Marketing)	

GENERAL ELECTRIC

Computer Department
Phoenix, Arizona

SUBJECT

DIAL COMM
8*433-

COPIES: J. T. Coe

April 4, 1966

Mr. C. E. Thompson
Manager-Product Planning
ISMO - DVPP

Dear Chuck:

Attached is the GE 615 Business Plan which you recently forwarded to me for consideration.

Based upon factors presented, I have indicated my concurrence with this recommendation.

Very truly yours,

L. E. Wengert

LEW:mp

Attachment

Copy

GENERAL ELECTRIC
INFORMATION SYSTEMS DIVISION
Data Processing and Communication

570 Lexington Ave.
New York, N. Y. 10022

DIAL COMM
8*222-3531

SUBJECT

COPIES: L. E. Wengert

April 15, 1966

Dr. L. T. Rader
Charlottesville

Dear Lou:

Enclosed is the GE 615 Business Plan and final price approval form.

You recall that the marketing organization had been authorized to selectively quote this equipment, and there is enclosed a summary of the present situation.

The equipment is needed to broaden the 600 line at the low end, and create customers who can grow up into the 600 systems of larger size at a later time.

I have instructed Vern Cooper and the marketing administration people that there must be absolutely no more than one shipment promise per month in 1967 for the 615 equipment. The reason for this is that we cannot count on more than one 600 system per week next year, and tentatively one in four of these has been allocated to export.

My comments on the proposal from Special Information Products Department are in a separate letter.

Regards,

J. T. Coe

cdm
Att.

GENERAL ELECTRIC

INFORMATION SYSTEMS DIVISION
Data Processing and Communication

570 Lexington Ave.
New York, N. Y. 10022

DIAL
COMM
8*222- 3531

SUBJECT

Special Information Products
Department Proposal

COPIES: L. E. Wengert
L. C. Maier
H. Van Aken

Dr. L. T. Rader
Charlottesville

April 15, 1966

Dear Lou:

I have reviewed carefully the Special Information Products Department proposal to build 20 615 central processors in 1967 and 30 in 1968. With my present understanding of the situation, I recommend that we do not accept the proposal as it stands.

Here is the situation as I see it:

1. We must have "turn-around" processors to match up with returned 2 microsecond memories and IOC-B input-output controllers. This turn-around processor can be supplied as a 625, a 615 processor produced as a de-tuned 625, or a new 615 processor from Special Information Products Department.

2. As John Burlingame has been informed and understands, our basic capacity limitation is not in the central processor area. The capacity limitation appears to be most critical in disc files, tape handlers, and new designs of card readers and card punches. We must emphasize the absolutely crucial nature of the disc shortage. The 600 line equipment is not useful in the marketplace without a disc, and even the DSU 200 now being furnished is apparently quite inadequate to match the capability of the processor, the expectations of the customers, and the competitive capability. We estimate that 90% of all 600 systems will require a disc of some kind, and that most of them will require at least two DSU 250 discs per processor.

In the other area of critical capacity limitation, we are familiar with the memory supply limitations and the drum supply limitations.

3. Lacking a firm schedule on bringing the design and production of the major peripherals up to the rate of system sales, I believe it would be unwise to commit for more processors.

GENERAL ELECTRIC

4. I like the concept of broadening our engineering and manufacturing base, however, and if we could see our way clear to balancing our systems output, I would be in favor of the proposal with one modification. I would recommend that Special Information Products Department absorb the engineering costs of the 615 and quote us the same firm price on all units.

5. I do not have the export picture clearly in mind, but I doubt if tape handlers and disc units are available from overseas sources to match up with these proposed processors.

Regards,

J. T. Coe

cdm

Following 4/15

8*273-6116 April 6, 1966 Dr. L. T. Rader

Information Systems Division
P. O. Box 909
Charlottesville, Va.

SIPD 615 Proposal

Messrs. J. T. Coe
L. C. Maier, Jr.
H. Van Aken
L. E. Wengert ~ on 5/18

Attached is a copy of a letter from John Burlingame, General Manager of Special Information Products Department. Dr. Rader would like to have your written comments as soon as possible.

Sincerely,

R. B. Curry

Attachment

4/4 RB Cawley

GENERAL ELECTRIC
DEFENSE ELECTRONICS DIVISION
SPECIAL INFORMATION PRODUCTS DEPARTMENT

DIAL COMM
8*256-1813

SUBJECT

COPIES JT Coe
GA Hoyt
LE Wengert

Building #5 Room C-6
Court Street Plant
Syracuse, New York
March 31, 1966

LOUIS T. RADER
RECEIVED
APR 4 1966

Dr. Louis T. Rader
Vice President and General Manager
Information Systems Division
2000 Holiday Drive
P. O. Box 909
Charlottesville, Va. 22901

Dear Lou:

We have reviewed the 615 proposal with just about everyone in ISD who might have an interest in the program and have come up with the following pros and cons:

Pros

1. There will be more than 2500 615 size computers sold in the 1966-1970 time period as estimated by Chuck Thompson's organization.

2. General Electric should be able to get 150 of these according to Thompson. (This sounds low to me, but should be a conservative estimate in any case.) Not low But lucky to do so—

3. The only way open to ISD today to penetrate this market is through a "detuned 625". While the ISD master plan contemplates a 615 machine, there is no such machine being designed today and very little likelihood that one could be available for shipment before 1968. The "detuned 625" is an uneconomical approach compared to the SIPD 615. If twenty SIPD 615 systems were sold in 1967 in lieu of the thirteen "detuned 625's", General Electric's sales would

improve about four million dollars and revenue between one and one and one-half million dollars, split about 60-40 ISD and SIPD.

4. An SIPD 615 can provide immediate market penetration with orders in 1966 and sales in 1967 with more than 50% more penetration than a "detuned 625" will give.

5. An SIPD 615 program will put no load on ISD 1966 budgets and should be self-liquidating in 1967.

6. The SIPD 615 program will give GE companion commercial and militarized hardware and permit increased penetration of the military market.

7. The SIPD 615 program gives ISD a 615 design with an inherently much lower manufacturing cost than a 625 and with the option of later manufacture at either ISD or SIPD locations.

Cons

1. The ISD planning people would prefer a 615 with 100 and 400 emulation capability. (We would, too, but the market is available today and so is the SIPD design, and we still don't know whether such a machine as the master plan contemplates is economically feasible and can generate sufficient margin.)

2. There is considerable question as to whether time and effort should be put in a less than ultimate design. (Since the design is effectively complete, and since the program is profitable and the market available, it is difficult to put much weight on this objection.)

3. Peripheral capability limits 615 market penetration. (This is true, but certainly by 1968 this should be history, and plans today must take into consideration the 1968 state of affairs, not today's limitations.)

Lou, it seems to me that an SIPD 615 program is a winner for General Electric, Information Systems Division and SIPD. After all the discussions I am more firmly convinced than ever

that ISD should consider seriously.

1. ordering 50 615's from SIPD with delivery of 20 in 1967 and 30 in 1968 with an option to cancel the 30 by January 1967 if desired,

2. accelerating their efforts to break the peripheral bottleneck (and we in SIPD and DED are interested in making whatever contributions we can here), and

3. doing a feasibility study to see if a 615 model B with 100 and 400 emulation can be an economical design and take whatever action is indicated by the study. SIPD would be happy to do this job or to help anyone else to do it.

this proposal before

I'll give you a call next week to see where you stand on this program. If I can contribute to a decision I'll be glad to meet with you and your deputies individually or collectively at your convenience. As I mentioned to you previously, we're ready to take off on the program.

Very truly yours,

John F. Burlingame

John F. Burlingame
General Manager
Special Information Products
Department

JFB/jm

CR 400 - 600
NO SOFTWARE SUPPORT
NO MAINTENANCE SUPPORT

TRAINING OF MAINTENANCE

will give Metz something to react

GENERAL ELECTRIC

Computer Department
Eastern Region
570 Lexington Ave.
New York, N. Y. 10022

DIAL COMM
8*222-2171

SUBJECT
• 615 Processor

COPIES:

April 19, 1966

Dr. Louis T. Rader
Vice President & General Manager
PO Box 909
Charlottesville, Virginia

Dear Lou:

Bob Curry in his letter of April 6 asked that I send you my comments with respect to John F. Burlingame's letter of March 31 concerning their proposal to manufacture a 615 processor of their design.

I am not familiar with the cost details of their proposal compared to a detuned 625, nor am I familiar whether their proposed 615 can run all of the software now proposed for our present line of 600 equipment.

However, it would seem to me that software would have to be 100 per cent compatible before we could even consider SIPD's proposal. As you know, if it is not compatible, the extra cost would far outweigh any cost savings which the SIPD proposal might show over the detuned 625.

In addition, it seems to me that the analysis of SIPD's proposal versus the detuned 625 should be looked at in great detail to determine whether costs of respective proposals are on a comparable basis net to GE Company.

Of course, the possibilities of having another operation help out in our overall manufacturing plans has merit.

Very truly yours,

Harm Aukes

HVA:km

DEPARTMENT CONFIDENTIAL

GENERAL ELECTRIC

COMPUTER DEPARTMENT

FINAL PRICE APPROVAL

Page 1 of Pages 4

FPA NO. 65-041 Date 11/16/65

References:

FPA NO. _____ Date _____

PPA NO. 5200 Date _____

PRODUCT NAME GE-615 System | LINE GE-600 | TYPE NO. CP8032

BRIEF DESCRIPTION: (A) Product, (B) Reason for Price, (C) Exceptions to Standard Pricing Policies, (D) Competition, (Source) (E) Effective Date and Availability, (F) Other

(A) Product Description -

The very strong requirement exists for a GE processor system which is price/performance competitive in the market segment existing between the GE-435 and the GE-625/635. The offering should be a low-end 600 system which is upward compatible with current systems. The availability timing (shipments in 1966) dictates an implementation depending heavily upon hardware developed for the GE-625/635. The GE-615 is essentially a modified 625, and will require no additional allocation of resources to software development. The performance of the 615 is targeted competitively with that of the IBM 360/50; the modifications to the processor are such that performance may readily be restored to 625 levels.

B. Reason for Price

Deferring availability of a small scale 600 system will divert a significant number of key users into the competitive fold. The system will be marketed selectively, the intent being to hold sales to key customers who:

a. Are placing multiple system orders, or
b. Are high prestige value users, or
c. Represent a potential for growth to 625/635/645.

System size will be constrained to one processor, one IOC and a maximum of two memory controllers and 128K of memory.

Estimated Cost Per Unit Based on * Units

Cost Estimate No. *

Date *

Shop Cost $ * % to Sale Price *

Manufacturing Cost $ * % to Sale Price *

Gross Margin $ * % *

Shop Cost Recovery in Months *

Sale Price to Rental Price in Months *

Recommended Prices:

Use Price Per Month *

Sale Price *

Maintenance Price Per Mo. *

Product Type (A) B C D E (Circle One)

Equivalence 50 Months

Domestic Shipment Forecast - LRFC Date _____

19 66	19 67	19 68	19 69	19 70	Total
3	25	46	40	30	144

APPROVALS:

Deputy Div. Gen. Mgr. for Information Systems Equipment
L. E. Wengert — Date 4/13/66

Manager - Pricing Analysis & Forecasting
T.H.Lee — Date 12/20/65

Manager - Product Service
J.C.Croyle — Date 1/5/66

Legal Counsel
J. Handros — Date 4/14/66

Mgr. Finance
C.A.Gillespie — Date 11/12/66

Division Vice President
L.T.Rader — Date

CERTIFICATION AND APPROVAL

This is to certify that the pricing decision has been independently arrived at and is in no way related to any implied or expressed agreement or understanding with competition.

Prepared By
F.J.Boyle — Date 11/16/65

Manager - Product Planning
C.E.Thompson — Date 12/8/65

Manager - Sales
A.A.Aschauer — Date 12/30/65

Manager - Marketing
V.S.Cooper — Date 12/22/65

Deputy Div. Gen Mgr. for Data Processing & Communication
J. T. Coe — Date 1/11/66

CK 76 (6-65)

* Refer to Attachment.

FINAL PRICE APPROVAL

No. 65-041 Date 11/16/65

PRODUCT LINE GE-600

PRODUCT OR SERVICE GE-615 System

REV.						

B, (Continued)

The pricing and performance of the GE-615 are set at values which place the GE-615 and the IBM 360/50 on approximately the same competitive level.

Approval is requested for the following GE-615 central system prices:

System	Use	Sale	Mo. Maintenance
615-40	$11,400	$ 547,200	$ 913
615-64	14,370	689,800	1,032
* 615-72	15,270	733,000	1,068
615-96	20,220	970,600	1,266
* 615-104	21,120	1,013,800	1,302
615-128	24,909	1,156,400	1,421

Note: Each System Price includes:
Processor
Memory
IOC-B with 3 high-speed and 5 low-speed channels
Console
MG Sets, with power sequencer.

* Not available as initially installed system; these are designations and prices only for systems up-graded from initially installed 615-40 systems.

FINAL PRICE APPROVAL

No. 65-041 Date 11/16/65

PRODUCT LINE GE-600

PRODUCT OR SERVICE GE-615 System

REV.

C. Exceptions to Standard Pricing Policies

Pricing per FPA No. 223, excepting that maintenance prices are the sums of established 625 maintenance prices for central system elements.

D. Competition

The market towards which the GE-615 is directed is that in which the RCA Spectra 70/55, the CDC 3300 and the IBM 360/50 are sold. Representative configurations are shown below.

RCA 70/55	Use	Sale	Mo. Maintenance
Processor, 32K Words	$ 9,950	$497,500	$ 398
Card Reader 1435 CPM	650	32,500	91
Card Punch 300 CPM	750	37,500	105
8 Magnetic Tape Units 60KC	3,600	169,200	504
Mag Tape Control 2 x 8	975	48,800	39
Printer, 1250 LPM	1,000	50,000	140
Disc Storage (2 Disc Packs)	1,150	52,600	102
Disc Storage Control	525	26,300	21
Selector Channel	800	40,000	32
Console	325	16,300	13
	$19,725	$970,700	$1,445

CDC 3300			
Processor, 32K Words (24 bit word)	$11,000	$ 497,000	$ 756
Card Reader 1200 CPM	400	22,500	65
Card Reader Control	100	4,800	35
8 Tapes 60KC	4,800	204,000	1,040
Magnetic Tape Control 2 x 6	1,100	52,000	145
Magnetic Tape Control 2 x 4	800	38,000	130
Card Punch 250 CPM	295	18,150	60
Card Punch Control	450	22,000	65
Printer, 1000 LPM	865	42,000	240
Printer Control	515	22,000	60
2 Data Channels	320	13,500	85
Disc Storage (2 Disc Packs)	1,150	52,600	102
Disc Pack Control	625	26,200	109
Typewriter	280	11,000	70
	$22,700	$1,025,750	$2,962

CS-67 (6/65)

PREPARED BY	DATE		APPROVED	DATE
F. Doyle	11/16/65	PRODUCT PLANNING COMPUTER DEPARTMENT		

IBM 360/50	Use	Sale	Mo. Maintenance
Processor 32K Words	$ 9,950	$ 499,700	$ 290
Card Read-Punch	660	35,000	90
Reader-Punch Control	970	46,500	41
Printer, 1100 LPM	900	41,200	138
Printer Attachment	75	3,000	1
8 Mag Tape Handlers and Control	4,560	222,600	579
Console	290	13,805	16
Disc Storage (2 Disc Packs)	1,150	52,600	102
Disc Storage Control	525	27,250	56
2 Selector Channels	1,400	63,200	52
	$20,480	$1,004,855	$1,365

GE-615			
Processor 40K	$11,400	$ 547,200	$ 913
Card Reader 900 CPM	650	26,000	130
Card Punch 300 CPM	825	33,000	165
Printer 1200 LPM	1,400	56,000	280
60KC MTH (8)	4,720	209,840	528
2 x 8 MTC	1,380	63,700	106
DS-15 Controller	700	32,310	54
DS-15 Disc Unit (2)	900	40,000	100
	$21,975	$1,008,050	$2,276

E. Effective Date and Availability

The pricing in this FPA becomes effective June 1, 1966, with availability of the initial system for delivery. Current 615 forecasts:

1966	1967	1968	1969	1970
3	25	46	40	30

UNIVAC 494 minim vvk

CK-67C (6/

Type #	Eqt. Type	Item	Use	Sale	Mo. Maint	Shop Cost	Mo. Rec. S.C.	SC % of Sale	Mfg. Cost	Gross Margin					Mo. Rec
										Sale		Use			
										Amount	%	Amount	%	#Mo	Sale Notes
615-40	A	40K 615 System	11,400	547,200	913	241,039	21	44.0	433,568	113,632	20.8	136,432	23.9	50	48
615-64	A	64K 615 System	14,370	689,800	1,032	306,662	21	44.5	528,568	161,232	23.4	189,932	26.4	50	48
615-72	A	72K 615 System	15,270	733,000	1,068	321,814	21	43.9	545,568	187,432	25.6	217,932	28.5	50	48
615-96	A	96K 615 System	20,220	970,600	1,266	403,457	20	41.6	670,568	300,032	30.9	340,432	33.7	50	48
615-104	A	104K 615 System	21,120	1,013,800	1,302	418,609	20	41.3	687,568	326,232	32.2	368,432	34.9	50	48
615-128	A	128K 615 System	24,090	1,156,400	1,421	484,232	20	41.9	782,568	373,832	32.3	421,932	35.0	50	48
Page 3 Configuration			21,975	1,008,050	2,276	395,506	18	39.2	630,241	377,809	37.5	468,509	42.6	50	48

Note: Each system price includes:

Processor
Memory
IOC with 3 high-speed channels and 5 low-speed channels
Console
MG Set with power sequencer

Peripheral subsystems and additional IOC channels are priced as shown in their respective FPA's.

The following data comprises an expansion of the shop cost information related to the Page 3 configuration for a 615-40. The analysis indicates the division of system shop cost between peripheral subsystems and the central 615-40 system. Costs are shown at current program estimate levels, and also as projected for 1968. One further entry is based upon the case of a system which includes a memory and an IOC-B returned from the field after one year of use in a 625/635 system. Shop costs for memory and IOC-B used are at the then existing book value, plus 10% of that value assigned as R&R cost. The book value figure reflects a twelve month amortization, six months in each of two fiscal years.

Shop Cost Data - 615-40 System Configuration

Shop Costs at 1966 and 1968 Levels:

	Shop Cost 1966	Months to Rec. S.C. 1966	Shop Cost 1968	Months to Rec. S.C. 1968
Central System	$241,039	21	$185,066	16
Peripheral Subsystems	154,467	15	131,967	12
Totals	$395,506	18	$317,033	14

Shop Cost of System when IOC-B and 40K Memory shop costs are represented as book value after one year plus R&R costs:

	Shop Cost	Months to Rec. S.C.
Central System	$192,294	17
Peripheral Subsystem	154,467	15
Total	$346,761	16

4/22/66

XEROX

RE: 615 (SIPD)
COMPATIBILITY

→ BOB & CARL:

BOENNIGHAUSEN SAYS:

SYSTEM COMPATIBILITY OF:

SIPD 615 (= M605) CPU

+ CED MEMORY CONTROLLER
+ CED IOC

SI!

↓

Program compatible

but

SIPD 615 CPU

+ CED MEMORY CONTROLLER
+ RGO RTIOC

NON!

fully program incompatible in I/O operations

WBC
CPR
PC

R. B. CURRY

R. B. Curry

April 14, 1966

Dr. L. T. Rader
Vice President and General Manager
Information Systems Division
Charlottesville, Va.

Dear Lou:

This refers to Bob Curry's letter of April 6 — attached — requesting comments on John Burlingame's letter on the subject of the 615 program.

Based upon my review of the product planning 615 proposal and a presentation by John Burlingame, it is my opinion that we should not accept the Special Information Products Department proposal for the following reasons:

1. This would inject a new, incompatible system into the product line which would add considerably to the confusion and expense of the operation.

2. It appears that eventually we would design a 615 line which would be different from the product proposed by SIPD, recreating the logistics problem at some later date.

Why can't SIPD design a 615 compatible with the present 600's and then we go ahead?

3. The principle advantage of the SIPD proposal is the availability of additional processors at a time we are capacity limited. This is very appealing, but the basic problem lies in having all the other items of equipment to go with the system; and SIPD is no help in that situation. The result could be that we would negatively affect other 600 line shipments in order to serve SIPD's 615 equipments.

? What are the problems getting the cost down?

4. I favor the "derating" of a 625 system to serve any specific need for a 615 system, as this retains within the organization the compatibility and production experience which could eventually be a factor in reducing cost.

Overall, because of the cost of both development and production by SIPD, the presumed net gain in income can only come from being able to sell more systems than we would otherwise be able to handle. It is certainly advantageous as to timing.

It seems to me that the negatives outweigh the advantages, and I recommend we proceed with the product planning program for a derated 625 system.

Very truly yours,

L. E. Wengert

LEW:mp

GENERAL ELECTRIC
DEFENSE ELECTRONICS DIVISION
SPECIAL INFORMATION PRODUCTS DEPARTMENT

SUBJECT

COPIES

DIAL COMM
8*256-1813

Building #5 Room C-6
Court Street Plant
Syracuse, New York
April 19, 1966

Dr. Louis T. Rader
Vice President & General Manager
Information Systems Division
2000 Holiday Drive
Post Office Box 909
Charlottesville, Virginia 22901

Dear Lou:

We have received the following information from Chuck Thompson's organization concerning 615 systems:

1. The following orders were not taken because a 615 was not available.

   a. Internal General Electric - 4 systems
   b. Colorado University
   c. First National Bank
   d. State of Michigan
   e. Mountain State (Bell Telephone)

   (A total of eight systems with no attempt to determine what might have been sold if a 615 was available and offered.)

2. 615's could be sold to the following customers if available. Requests to quote were at Phoenix as of 3/1/66.

   a. Allied Chemical (on order)
   b. G.M. Research (on order)
   c. Emerson Electric (on order)

d. Clear Oil
e. N. E. Tel & Tel
f. L.D.S. Church
g. Alcoa Aluminum
h. Western Electric
i. U. S. Radiological Lab
j. NASA-Huntsville
k. Oklahoma State
l. U. S. Steel (Birmingham)
m. U. S. Geological
n. Aeronetics
o. G.M. (AC Spark Plug)
p. U. S. Steel (Fairless Works)
q. Western Electric (Kearney, N. J.)
r. G.M. (Sangamo)

This would appear to answer the market question of specifically where would we start selling 615's if the machine were made available. The above list doesn't include other opportunities such as Western Union and a number of General Electric departments currently considering competitive equipments. It would appear that sufficient orders can be generated to support the ISD-SIPD program we have been considering.

Very truly yours,

John Burlingame

John F. Burlingame
General Manager
Special Information Products Department

JFB/jm

-52

C-76

600 SERIES TIME-SHARING

P. A. Quantz
A. L. Ellison
H. N. Cantrell

1967 March 29

TO: E. R. White

FROM: R. W. Bemer

Dr. Shuey provided you with a list of changes that he felt necessary to equip the 625/635 for time-sharing. In order to comment on these, Messrs. Cantrell, Ellison and myself have documented a plan for a time-sharing system. We feel this is a very useful plan which has value outside of simply judging these hardware features.

The rationale for the several features were considered on the basis of our present position and investment, particularly since a software system represents both a considerable expenditure in dollars and a substantial minimum elapsed time for fabrication. This resulted in the following box score for these features:

1.	Illegal Op Code Trapping	done, or in process
2.	Multiple Base Registers (4)	no
3.	Large Disc	yes (DSU 250 ideal)
4.	Fast Drum directly connected to store	no (if #3 satisfactory)
5.	TRIOC	no
6.	GEPAC 4020	Perhaps
7.	PDP Graphic Terminal	no

Special note should be taken of the GEPAC 4020, for we considered it a general adjunct as a collector or pre-processor and not basically for online experiments per se.

Assuming a GE corporate policy of prohibiting proliferation of machine types, we are provided with a basic premise that software available to one model in the line must be compatible upward, at least, if not downward. So far the 605 has followed this position by having the same instruction set as the 625 and being able to run all 625 programs. By extension our analysis of Dr. Shuey's suggestions must apply equally to the 605!

If such changes were to be made to the 605 and not the 625, GE would not only be in the position of expensive proliferation but also in a bad marketing position with existing and future 625/635 customers who could not take advantage of a time-sharing system which depended upon these added features. The authors of this report are in agreement that it is impractical, if not impossible, to sell large computing systems in 1969 or 1970 without terminal facilities. Additionally, there could be a high possibility of return of rental systems.

It should be noted that although we have excluded consideration of a second base address register by virtue of heavy software commitment to single registers, there is nevertheless no study available to us on comparative cost between these options. It is suggested that this would be very useful. Note also that the RTIOC, which we excluded by reasons of not existing for the 625, is really excluded for all time-sharing systems because it is not a suitable collector.

It is reported that the 605, being a later machine, implements the same instruction set by simpler logic. This is of course quite possible, witness the situation between IBM's 360 and RCA's SPECTRA. We should not fail to investigate this for possible benefit to the 625/635.

/cac

TO LANCE 67 MAR 21

Is Mr. Bennett.

O.K.

From: BAQ

BY SCOPE?

COMPAGNIE
BULL
GENERAL ELECTRIC

94 AVENUE GAMBETTA PARIS 20

CO/GEC/1015

July 27th, 1966

Mr. R.W. Bemer
GENERAL ELECTRIC COMPANY
P.O. Drawer 270
PHOENIX Ariz.85001

Dear Bob,

On my last trip to Phoenix, in June, you asked me for a list of all Time-Sharing projects presently going in various GE departments. Here is a likely incomplete review :

1 - Computer Dept. GECOS T.S. for 625/35. Dave LATIMORE

2 - Computer Dept. 400 T.S. Art Mc CABE

3 - Computer Dept. Project ASSIST for 625/35. Al DEAN

4 - Computer Dept. MULTICS, for 645. Ed VANCE

5 - IPB. Dartmouth system. Bill HAWKINS (Phoenix) - also people from the Dartmouth College and IPB in Hanover (N.H.), Falls Church (Va.), Santa Barbara (TEMPO).

6 - IPB (Phoenix), Oklahoma City, and Pillsbury Corp. Dartmouth system for 225.

7 - MEDINET (Boston) on a special 400.

8 - Heavy Military Electronics Dept. (Syracuse). Some kind of GECOS + Real Time system. Dick WARD

9 - (Maybe) Valley Forge Space Technology center (Philadelphie).

10 - French Weather Bureau (Paris), ou 635. Louis POUZIN.

As you said, it seems appropriate that some information interchange be organized among all these projects, rather than let everyone start from scratch. In my case, I would be very happy to use some ideas or programs already worked out by other people. Do you think you can help that ?

.../...

Dick Ward, from Syracuse, told me he sent you some specs of their system. Could you get me a copy ? Since we have not yet written our specs, we are particularly now interested in all sources of inspiration.

Thank you for your diligence.

Sincerely,

Louis Pouzin

Louis POUZIN

DICK WARD OF S.I.P.O CALLED JUN 21 ~1430 MST (WORKS FOR DIX) WILL SEND COMPATIBILITY STATEMENT AND CERTAIN SPECS. SINCE CHO & WEIL RECOMMENDED MY ADVICE, REQUEST A VISIT.

GENERAL ELECTRIC
DEFENSE ELECTRONICS DIVISION

DIAL 8*256-1262

SUBJECT

COPIES
R.A. Boennighausen (w/att.)
J.F. Burlingame (w/att.)
J.G. Hutton (w/att.)
E.F. Roache
R.E. Ward

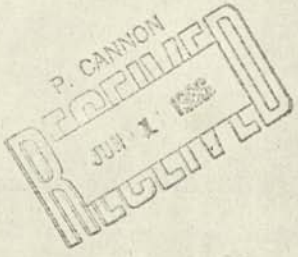
Room H6, Building 5
Court Street Plant
31 May 1966

Dr. P. Cannon
Manager, Operations Analysis
Division Planning Operation
2000 Holiday Drive
Charlottesville, Virginia 22901

Per our telecon of 27 May, enclosed is a copy of the Preliminary Functional Specification describing the Software package we are proposing for the 615 computer. Of specific interest is the 615 Executive. As discussed with you, the executive is defined with GEFRC as the primary I/O interface, thus cross program compatibility between the 615 and the 625/635 is assured as long as that interface is maintained.

The spec is written for the 615 in a RT/IOC, 32K memory configuration and is based on the 605 executive. Conversion of this executive for a 615 in an IOC configuration is a relatively minor project (of the order of six man-months) and is also being proposed as part of the 615 program.

We would be pleased to go into any details beyond the rather brief specification content if you desire.



C W Dix

C. W. Dix, Manager
Computer & Data Systems Eng'g
SPECIAL INFORMATION PRODUCTS DEPT.

CWD:np
Att. (1)

VANCE HEFNER

-51

GENERAL ELECTRIC
COMPANY

13430 NORTH BLACK CANYON HIGHWAY, PHOENIX, ARIZONA 85023 . . . TELEPHONE 941-2900

COMPUTER
DEPARTMENT

DEER VALLEY PARK PLANT

1967 March 28

Mr. J. A. Haddad
IBM Corporation
Armonk, New York

Dear Jerry:

Please add my congratulations to the many I know have come to you both from within and external to IBM. It must be a source of great personal satisfaction to be elected a corporate officer of a so distinguished a company.

You might like to know that it is a source of satisfaction and pleasure to me, as well. I am happy that this recognition has come to someone I have known for a long time and in friendship. I am pleased that another person from the technical and scientific area has been recognized by IBM in this way. Moreover, I am delighted that your title includes the word "programming", for now I might begin to hope that the 360 software could be brought under control before the very momentum and appetite of this behemoth vitiates the computing industry. It can be fixed, you know.

It would disappoint all who have worked with you, however, if this change were to deprive the USA Standards Institute of your services. You know well that real understanding at the X3 level is possessed by very few of its members, and for you to relinquish your membership to an IBMer with less vision (regardless of how technically competent) would lessen severely the value and effectiveness of X3 and its substructure. Besides, I like a reasonable adversary that one can agree with most of the time.

Bob

R. W. Bemer

/cac

C-76

l.cc: M. Grems

GREMS PROPOSAL

1967 February 24

TO: J. W. Weil

FROM: R. W. Bemer

CRITERIA FOR EVALUATING RESEARCH CONTRACTS

Although an apparent plausible method, I do not see where ASTO will be doing this large a business in terms of numbers of contracts. Somthing like the Federal Government might use this but they would not get down to the detail of the individual researcher. Unless there is something else which is not evident, I am not enthusiastic.

USER ORIENTED INFORMATION PACKAGE

This is something else again. Some respected studies of the future in information processing have taken the expansion to a vastly great number of users as the major area of investigation. The RAND Corporation, for example, had what they called COMBOMAT back in 1951. This was a very simple system which excluded as many machine characteristics as possible. This company still takes the same approach and each office in the buildings is wired for terminals for the use of physicists, mathematicians, economists, etc.

On the other hand, look at PL/1, which is so complex that there is general agreement that it will be impossible for the casual user. Obviously we must try simplified methods periodically.

I think Miss Grems proposal an excellent one. It has been tried before, surely, but rigor and standards are the key to possible effectiveness. It is obviously an extension of the program interchange followed by several user groups, but in a homogeneous environment.

One would ask whether this should not be university research rather than that sponsored by a computer manufacturer. The answer is no, because most of the techniques are known to us and it is a matter of development rather than invention. Furthermore, it requires the controllability which a manufacturer can impose on his software during production to make sure that an adequate

number of functions is available coincidental to system introduction. This is an interfaces problem of software, which only the manufacturer or a standards group could control.

I think this falls right in the middle of the ASTO bailiwick.

1967 February 23

TO: E. M. Koerits
E. R. White

FROM: R. W. Bemer

In view of the spectacular work which Ellison & Cantrell have been doing to boost 600 software performance, I pass on a rather ancient document found in the SHARE (IBM users) files, which I obtained from TIPO.

It came as a surprise to me in looking through these early documents to find many notes by Harry Cantrell, and the one attached here indicates how often wheels are re-invented in the computer business. This may be an opportunity for me to say that the work of Ellison and Cantrell is absolutely outstanding and worth many millions of dollars to the company. If IBM knew how to do this as these people do, most of the 360 problems would disappear. We should be grateful that they don't.

/cac

-20

C-76

E. R. Vance
P. A. QUANTZ

600 FORTRAN

2/10/67

TO: E. R. White
(Acting Manager, Programming)

FROM: R. W. Bemer

1) Mr. Vance's report of February 5 states "Changes to GECOS in Change Letter 9 and 10 are taking away much of the advantage the reimplementation had over the current FORTRAN, i.e. these changes are significantly improving the compile speed of the current FORTRAN. We are coordinating with ISMO and 600 Project Operation to determine the proper strategy after all timings are in."

2) Mr. Dobbie says that a very major difficulty in meeting TRW's acceptance requirements lies in the FORTRAN area.

3) In the Design Review of the new FORTRAN last November 15, the elapsed compilation time comparisons were given as follows:

a) Current FORTRAN (drum) - 210 statements/minute
b) Current FORTRAN (disk) - 113 statements/minute
c) New FORTRAN (64K) - 327 statements/minute
d) New FORTRAN (32K) - 210 statements/minute

Contrasting with this I must advise you that our competitor, UNIVAC, is able to compile between 6000-8000 statements a minute. IBM feels that 6000 a minute is a reasonable goal for the highest model 360.

With the cleanup of GECOS II we more nearly approach numbers of statements/minute based upon residence time. In this connection I attach a copy of my 66 September 19 memo. On the basis of GECOS improvement, we might expect (roughly) the 590 statements per minute for FORTRAN to go to 800 or so. A similar type of improvement (perhaps not proportional) might bring TASMIN up to 3500 statements per minute on the 625. If this same processor were applied on the 635 this would perhaps reach 4500. In this magnitude we begin to become almost competitive.

- -

Summary:

The 600 is sold primarily as a scientific machine. Most scientific usage is via FORTRAN. Compilation speed is a significant factor because the same problem is often compiled 30-50 different times before it is checked out and ready to do work. Therefore, a major factor in the buy-decision in the scientific area is the FORTRAN compilation rate.

You will notice (in the next to the last paragraph in the attached memo) that a one-man-year effort required to rework TASMIN into FORTRAN was offered as a basis for a business decision re insurance against inadequacy of the new FORTRAN. Since I received no reply I assume that the business decision was negative. However, even at this late date you may wish to reconsider this decision. If so, please remember that TASMIN is written in G-language according to the same desire for maintenance and construction ease as the new FORTRAN is written in Digitek-Pops. It might be useful to find out whether the inefficiencies of the new FORTRAN are due to the construction language or the compilation theory.

1967 January 25

TO: E. R. White

FROM: R. W. Bemer

I suggest that you might have a discussion with Mr. Quantz on the effectiveness and goals of Mr. Dobbie's operation. It is doubtful whether all the items on their full agenda are in fact the most important to some of our immediate problems. This applies particularly to performance measurements, where the information yielded is not generally too useful for the programmers who would have to apply it. For example, last week we saw an extensive report which utilized more than an hour of machine time and considerable programming to answer the trivial question "should GECOS try to re-read tape 25 times?" One could have found several experienced people around who would have said "no, try 3-6", and have been saying this for several months.

I am carefully limiting these comments to performance measurements. In contrast, the GECOS III model was found very useful prior to actual construction. Basically I believe the emphasis is wrong. It is not as useful to evaluate performance to say what it is, as it is useful to know why performance is lower than the achievable, in order to better it. Putting it another way, the Cantrell/Ellison analysis perhaps should have been done in this group. In addition, TIPO has provided us with measurement tools which show performance for the real, rather than theoretical, world.

Continuing this argument, it should be pointed out that the work of Cantrell and Ellison has been aimed at finding inefficiencies and conflicts in the master mode. No work has been done yet in the slave mode to analyze the penalties and frequencies in binary-to-BCD conversion, editing, GEFRC usage, etc. Nor has the object code from our compilers been analyzed to see how they cause particular machine instructions to be used, for efficiency. This work should be going on now, providing for analysis of the actual way our customers utilize the machine (which is out of our control), not the way we think they do or should use it. Presently there is no evidence as to whether we should offer BCD arithmetic in the 600, as IBM does in the series 360. Weil and Couleur believe we should not, but I have seen no evidence one way or another. I know that IBM makes exhaustive analysis of customer programs.

If it should be decided that Mr. Dobbie's group cannot undertake this, perhaps it would be useful to enlarge the Cantrell/Ellison function with some additional top quality programmers to do some of this work in parallel. In the apparent absence of an advanced development function in CED, perhaps this could lead to some innovations in technique and input to hardware designers not presently yielded by Dobbie's operation. This leads to the question as to whether ASTO might wish to participate.

3658
C-76

P. A. Quantz
E. R. Vance

EARLIER THROUGHPUT IMPROVEMENT

January 20, 1967

E. R. White, Manager
Engineering

The return to aggressive selling of the 600 is still arrested. I am concerned by the sequential lags which will occur in firing up first our sales management and then the salesmen in the field. While much in favor of applying stringent quality control procedures to Change Letter 10, I am afraid its delivery time is somewhat out of phase with what we would hope to achieve.

I have already discussed with Mr. Vance and Mr. Gillette the possibilities of sending out those major throughput changes scheduled for Change Letter 10 via patches which can be installed safely in Change Letter 9. I do not believe any action is required on your part, but I wanted to be sure that you know the rationale for such action. Naturally, only a portion of these throughput changes can be applied to Change Letter 9 in this fashion, but they can have a substantial effect on customer satisfaction, perhaps two months earlier. This can be fed back to ISMO.

Of course, the primary aim is to demonstrate as quickly as possible that phenomenal improvements are being achieved on Change Letter 10. Mr. Cantrell suggests that this hope might be enhanced by inviting representatives of our most qualified customers to run some sample work loads on the prototype software system here in Phoenix. In this case it wouldn't hurt to have district sales managers as witnesses.

R. W. Bemer

RWB:cbb

MARTIN COMPANY

1966 September 23

- PRIVATE -

TO: L. E. Wengert

FROM: R. W. Bemer

It has been passed around that some top management of I.S.D. has effectively written off the Martin decision with a feeling that we were committed beyond our capacity to deliver performance, and therefore well out of an expensive commitment. No good money going after bad.

However, I am disinclined to think that the requisite performance could not be reached, subject to sufficient discipline, ingenuity and open-mindedness. Not on schedule, of course, but perhaps within a time period which Martin could have been persuaded to grant.

Was there a perhaps deliberate plan to achieve the present result? Certainly IBM, with a desire to retain a major customer and prevent deterioration of image, would have made a fierce effort with top management personnel.

If this is correct, may I suggest that you ask ISMO to put our 600 customers in two groups, thus:

1) Like Martin, we have oversold and overcommitted in contracts and are therefore happy if they cancel or return the equipment. TRW is certainly in this class, for example.

2) Others, validly committed.

I suggest this because an effective sales organization should be able to make this judgment prior to being thrown out; afterwards invites a suspicion of rationalization. Such a list could be quite illuminating. At T.I.P.O., for example, a Paramus taxi pulls up every night at 5 P.M., picks up work for delivery at 8 A.M. the next morning. It's done on ITT's 7094 Mod 1 at $325/hour, which perform (on a time basis) better than the T.I.P.O. 625 at approximately $900/hour. CEIR in NYC is bidding at $260/hour. I hear that Mr. Eaton plans to speak to you on this.

If such an assumption is not correct (we did not wish to lose Martin), could we set up some safeguards in our handling of customers?

I hope you realize that the foregoing was written in some vacuum as to divisional plans and motives. If I am wrong, ignore it and consider it only as insurance to make sure all inputs get to the right decision level. I'd rather speak up and take my chances than have G.E. lose through any oversights.

RUSH DELIVERY

THE THREE GENERAL ELECTRIC 600 SYSTEMS AT MARTIN COMPANY'S BALTIMORE, DENVER AND ORLANDO DIVISIONS WILL BE RETURNED TO GENERAL ELECTRIC WITHIN THE NEXT SIXTY DAYS. THIS DECISION HAS BEEN REACHED JOINTLY BY GENERAL ELECTRIC AND MARTIN AFTER EXTENSIVE CONSIDERATIONS.

THE ACTION ARISES FROM REQUIREMENTS, SOME OF WHICH ARE COMMON, AND SOME OF WHICH ARE UNIQUE TO BOTH COMPANIES. IN VIEW OF THESE REQUIREMENTS, IT WAS CLEARLY IN BOTH GENERAL ELECTRIC'S AND MARTIN'S BEST INTERESTS TO WITHDRAW FROM A BUSINESS RELATIONSHIP WHICH WAS NOT FULLY SATISFACTORY TO EITHER COMPANY.

IN RESPONSE TO QUERIES YOU MAY RECEIVE FROM CUSTOMERS, YOU SHOULD INFORM THEM THAT THIS DECISION REINFORCES GENERAL ELECTRIC'S DETERMINATION TO ACT IN THE COMPUTER INDUSTRY IN A BUSINESS LIKE AND MUTUALLY SATISFACTORY MANNER WITH ITS CUSTOMERS. WHEN A ECONOMIC REQUIREMENTS OF A BUSINESS TRANSACTION REMOVES THE PROSPECT OF REASONABLE REWARDS FOR THE SUPPLIER, IT IS GENERAL ELECTRIC'S POSITION THAT THE BASIS FOR AN ENDURING AND SATISFACTORY RELATIONSHIP NO LONGER EXISTS. IT IS THEN IN THE BEST INTERESTS OF BOTH COMPANIES TO EFFECT AN APPROPRIATE TERMINATION.

WHILE THIS STATEMENT DOES NOT FULLY DESCRIBE THE SITUATION WITH THE MARTIN CO., GENERAL ELECTRIC BELIEVES THAT THE ACTION TAKEN IN THIS CASE WAS CONSISTENT WITH SOUND BUSINESS PRACTICE. IN NO WAY DOES GENERAL ELECTRIC IMPLY THAT THE MARTIN CO., OR ITS PEOPLE, HAVE ACTED IMPROPERLY OR IN BAD FAITH. RELATIONS WITH THE MARTIN CO. FROM THE OUTSET HAVE BEEN CORDIAL AND CONSTRUCTIVE, AND THIS PLEASANT CIRCUMSTANCE CONTINUES.

THE PRESENT DECISION, ALTHOUGH REGRETTABLE, HAS BEEN REACHED WITH THE BENEFIT OF FULL UNDERSTANDING ON BEHALF OF BOTH COMPANIES. IT IS ENCOURAGING THAT THE CLIMATE PREVAILS FOR SOUND, CONTINUED COMMERCIAL RELATIONS, INCLUDING THE PROSPECT OF GENERAL ELECTRIC SUPPLYING THE MARTIN CO. WITH COMPUTER EQUIPMENT AND/OR SERVICES TO ACCOMMODATE ITS CONTINUING NEEDS.

AS WITH ALL THIRD GENERATION COMPUTER EQUIPMENT, DIFFICULTIES ARE BEING EXPERIENCED. PROBLEMS IN IMPLEMENTING LARGE COMPUTER SYSTEMS, AND THE INDUSTRY HAS MANY EXAMPLES, DO CONFRONT GE ON THE GE-600 LINE, BUT THEY ARE MANAGEABLE AND IN THE PROCESS OF CORRECTION. WHILE SOME OBSERVERS OF THE INDUSTRY MAY WISH TO SPECULATE THAT GENERAL ELECTRIC IS NOT AGGRESSIVELY BACKING THE GE-600 PRODUCT LINE, OR THAT THIS EQUIPMENT WILL NOT PERFORM SATISFACTORILY, SUCH CONJECTURE IS AT VARIANCE WITH THE FACTS. GENERAL ELECTRIC IS CONTINUING TO PERFECT AND ADD CAPABILITIES TO THE GE-600 COMPUTER PRODUCT LINE, AND CUSTOMER ACCEPTANCE, EXPRESSED IN ORDER BACKLOG, IS AT A RECORD HIGH.

THE GENERAL ELECTRIC PEOPLE INVOLVED IN SERVING THE MARTIN COMPANY'S DURING THE PAST TWO YEARS DESERVE OUR CONGRATULATIONS. THEIR EFFORTS WERE CHARACTERIZED BY DILIGENCE AND COMPETENCE AND I COMPLIMENT EACH OF THEM FOR THEIR FINE PERFORMANCE.

VERN COOPER
COMPUTER

-27

3658

C-76

W. Sullivan
D. Marden

1966 July 27

TO: P. A. Quantz

FROM: R. W. Bemer

The attachment shows that, in the 600 Sort mailbox on July 25, Item 22 is courtesy of an unknown IBM representative. Shown later are other items from the mailbox after being cleared, and you will notice under item 6 that the IBM representative is quite a decent fellow. I think we owe him a word of gratitude, which might even be passed along before we change the password as indicated in item 6.

May I suggest that you put out a policy statement on the usage of all mailboxes, to include:

1) Mandatory change of all passwords, at least quarterly.

2) Strict prohibition against posting these passwords at any site.

3) If adequate security is not achieved by these two measures, a further restriction for acceptable user numbers (in addition to the password) should be applied to the mailboxes.

I think Mr. Marden handled the present situation quite well.

+AB
-16

C-76

R. B. Curry

600 SOFTWARE

1966 July 18

TO: L. E. Wengert

FROM: R. W. Bemer

You have asked how to shorten the time to get the 600 software in acceptable condition. Having been asked, I believe that I can furnish an answer. The clue was given in recommendations 1 and 2, page 6 of my June 21 report to Dr. Weil (copy attached to my July 13 memo to you on COBOL).

A doctor will often delay an operation until the patient is stronger. The poor condition of 600 software at this late date stems directly from long operation upon a weak machine. The proper action would have been to stop software checkout completely, get the hardware functioning adequately, and then restart the checkout at a higher rate of effectiveness.

This also applies to the repair and improvement of software. It is said that we cannot shorten our schedules by adding emergency personnel (for example, in GECOS) because of the complexity and training period required. TRUE - but why? Because the software is, as a result of former lax production disciplines, patched and repatched, and possibly not documented well enough for anyone but the initiates to comprehend in the short time which remains to us.

My solution is therefore to forcibly stop those responsible for the software units from doing any other work until these units are reassembled to one clean source form to take as a reference point. Multiple copies of listings and flowcharts to be furnished as required. Until this point is reached, we are in the position of having the author's rough notes in jumbled form, which makes it hard for a second party to comment or make suggestions, But if a properly sequenced draft is distributed, many people can contribute.

This base point is reached when the software system is re-established in clean form and revalidated to be roughly equivalent to Change Letter 7, and will even show a modest thruput improvement (SWAG - 5%). Now personnel may be assigned to study independently the operational methods of the software units. Modifications and improvements may be tried. If successful, we profit; if not, try again.

These added personnel may be:

1) Phoenix programmers temporarily reassigned from other projects,

2) 600 programmers from inhouse G.E. operations (SIPD, TIP, etc.), or

3) Top consultant personnel with 600 programming experience.

Their time will seldom be wasted, for there are many peripheral improvements which can be made independently and will benefit operation and thruput from a human engineering standpoint, such as:

1) Finer breakdown of the abort codes so the user knows why it happened, was it his fault or not, etc.

2) Putting basic startup configurations permanently on disc.

3) Giving GECOS heavier armor, for it is presently designed to a 100% functioning of hardware (i.e., not lenient or forgiving).

4) Store dump to tape rather than forcibly to printer on aborts.

etc., etc.

I remain convinced that this one relatively simple strategem will allow us to put additional strength on a very serious problem, whereas this is virtually impossible under present conditions. As another analogy (perhaps superfluous), it is wiser to fix the flat tire and then drive fast to make up time, rather than limping along at a much slower rate.

In the software microcosm this may be considered as operating inefficiently, and as such repugnant to its management, who rightly judge that this is a less stable condition and more difficult to control. To this we must reply that we are now at such a point of lost rental and present emergency that it will nevertheless be a more efficient overall solution.

/cac

600 COBOL ANALYSIS

E.M. Koeritz
J.W. Weil
P.A. Quantz

1966 July 13

TO: L. E. Wengert, Deputy
Division General Manager

FROM: R. W. Bemer

Following Dr. Weil's instructions to conclude an in-depth analysis of 600 COBOL by August 1, it is apparent that my portion of the study is complete. A mechanism is established for benchmark and continuing analysis of 600 COBOL performance under the line organization.

Various attachments are provided and referenced here.

As a general summary, I found the compiler to be of a very reasonable quality and the group that produced it to have been excessively maligned. The major problems and solutions are given here:

PROBLEM	SOLUTION
1) Late delivery in operable condition.	None - This state may be attributed primarily to very poor performance of the checkout hardware, as well as excessive GECOS malfunctions.
2) COBOL reported to malfunction excessively.	Field personnel have not complied adequately with the reporting system of software notes. I have drafted an insert for the mailbox (attachment 1) which indicates to field personnel their responsibilities. In addition, this may be sent out for general distribution. Note that 353 reports have been reduced to zero!

PROBLEM	SOLUTION
3) Both compilation and running programs stated to be inefficient and slow.	a) Tests were run to determine better strategy. Timing facilities have been added to the processor. An AE Aid (attachment 2) has been produced to equip field personnel with the necessary knowledge to achieve better performance. b) There was a bad design decision in COBOL being forced to run through GEMAP for assembly. This penalizes elapsed time in compilation from 40 to 200% and is in my opinion absolutely unjustified. Mr. Quantz will have to make a decision about building an integral assembler. c) A substantial proportion of performance complaints may be laid to rumor and invalid comparison. Performance report forms have been prepared (attachment 3) and also placed in the 6.45 COBOL mailbox for faster reporting. No performance complaints will be heard except via this method. d) COBOL throughput performance is heavily dependent on the clean-up of other major software elements (initial report to Weil, attachment 4).
4) Stated that 600 COBOL will not be competitive in '67.	Our maximum measured compilation is (so far) about 700 statements a minute, compared to 1000 on the 7094 and 3000 on the 1108. An integral assembler and the 1 microsecond store can bring this to 1200. The IOC-B HICCUP or the IOC-C can provide substantial improvement because printing time is the longest compilation task. The DSU-250 could add another improvement but we have no measurements yet. With respect to running programs produced by COBOL, I have seen no evidence to indicate that normal improvements during the life of the compiler will not yield running programs competitive with any others operating under the GECOS system. In other words, any running time inefficiencies in COBOL object programs will be primarily due to GECOS and hardware limitations.

The 600 COBOL compiler is "state of the art" except for the GEMAP limitation arbitarily imposed. The design (which is largely table-driven) appears to be very effective for rapid maintenance and facilitating customer usage. Until recently the morale of this group has been quite low. I am glad that I was given this assignment since the results and benefits of my investigation appear to have improved this situation.

However, the 600 COBOL processor will require continuous improvement and tuning over the next 18 months. Such processors normally require original personnel for about 3 years! Mr. Quantz may wish to consider completion awards for personnel sticking it out satisfactorily.

/cac
Attachments

GENERAL ELECTRIC

Computer Equipment Department
Phoenix, Arizona

DIAL COMM 8*433
MAIL DROP C-80

SUBJECT

COPIES: P. Quantz

June 15, 1966

Memorandum to R. W. Bemer:

As you are aware, we have been engaged for three years now in the development of a very large COBOL compiler for the 625/635. We have spent the last year in the final throes of debugging and introducing this compiler to the field. During this period of time, we have been repeatedly told by our project personnel, and by several outside consultants or organizations, that we had planned and executed one of the most advanced COBOL compilers in the industry, both with respect to features included and its internal construction.

Two recent events make this situation somewhat suspect:

1. Comparisons by the Martin Company on compilation times indicate that COBOL is significantly slower (a factor of perhaps 2 or 3) than the COBOL compiler on the 7094.

2. We continue to have a good deal of dissatisfaction with the ability to actually get work thru COBOL. While a great deal of progress has been made, it is not clear that enough progress has been made.

May I ask you to undertake a special assignment. Would you please make an audit of the 600 COBOL compiler, making use of your background in this area, to arrive at an objective appraisal which will identify deficiences or problems, as well as those areas in which you believe our efforts to have been strong and effective. If you require any consulting assistance in analyzing the 600 COBOL compiler, please let me know.

COBOL continues to be of intense interest to our 600 users and the indications of disappointment over its performance, both here and abroad, add to an already difficult situation. May I ask that you make a preliminary report to me on the morning of Wednesday, June 22nd and that you be prepared with your final report when I return from vacation early in August.

John W. Weil

JW/ll

John W. Weil

GENERAL ELECTRIC

Computer Equipment Department
Phoenix, Arizona

DIAL COMM 8*433
MAIL DROP C-80

SUBJECT

COPIES: P. Quantz

June 15, 1966

Memorandum to R. W. Bemer:

As you are aware, we have been engaged for three years now in the development of a very large COBOL compiler for the 625/635. We have spent the last year in the final throes of debugging and introducing this compiler to the field. During this period of time, we have been repeatedly told by our project personnel, and by several outside consultants or organizations, that we had planned and executed one of the most advanced COBOL compilers in the industry, both with respect to features included and its internal construction.

Two recent events make this situation somewhat suspect:

1. Comparisons by the Martin Company on compilation times indicate that COBOL is significantly slower (a factor of perhaps 2 or 3) than the COBOL compiler on the 7094.

2. We continue to have a good deal of dissatisfaction with the ability to actually get work thru COBOL. While a great deal of progress has been made, it is not clear that enough progress has been made.

May I ask you to undertake a special assignment. Would you please make an audit of the 600 COBOL compiler, making use of your background in this area, to arrive at an objective appraisal which will identify deficiences or problems, as well as those areas in which you believe our efforts to have been strong and effective. If you require any consulting assistance in analyzing the 600 COBOL compiler, please let me know.

COBOL continues to be of intense interest to our 600 users and the indications of disappointment over its performance, both here and abroad, add to an already difficult situation. May I ask that you make a preliminary report to me on the morning of Wednesday, June 22nd and that you be prepared with your final report when I return from vacation early in August.

John W. Weil

JW/ll

John W. Weil

600 COBOL MAILBOX

$1. MAILBOX CLEARED 7-13-66 09:55HRS MST.....

"MILESTONES"

1. TODAY, JULY 13, THERE ARE ZERO SOFTWARE NOTES INHOUSE OUTSTANDING AGAINST 600 COBOL. WE ARE AWARE THAT SOME FEW ARE IN THE MAIL.

2. TO DATE 353 COBOL SOFTWARE NOTES HAVE BEEN ANSWERED AND/OR CORRECTED.

3. AS OF 12:30 TODAY THE INHOUSE COBOL SYSTEM IS NUMBER 58, A CORRECTED AND SOMEWHAT FASTER VERSION OF SYSTEM 55. THIS WILL BE RELEASED TO OUR USERS AT THE END OF JULY.

4. AN AE AID IS PRESENTLY BEING PRINTED, ENTITLED "EFFICIENT USE OF GE-600 COBOL". WE FEEL THAT THIS INFORMATION WILL FACILITATE USAGE AND IMPROVE COMPILATION AND EXECUTION TIME. WITH SUCH AN ADVANCED COMPUTER SYSTEM AS THE 600, IT IS DIFFICULT TO ACHIEVE OPTIMUM PERFORMANCE WITHOUT SOME GUIDANCE OF THIS TYPE.

5. THE EFFECTIVENESS OF THIS MAILBOX SYSTEM CAN BE MAINTAINED ONLY IF NOT ABUSED. MALFUNCTIONS WILL OCCUR IN ANY COMPILER OF THIS COMPLEXITY, BUT WE BELIEVE THAT CUSTOMERS ARE ENTITLED TO BE AWARE OF THESE SO THEY MAY AVOID THEIR DANGERS PRIOR TO REPAIR. THE LACK OF OUTSTANDING NOTES INDICATES THAT 600 COBOL IS NOW IN EXCELLENT CONDITION. HOWEVER, USAGE MUST BE OBJECTIVE, FACTUAL AND CONSTRUCTIVE. IT WOULD BE MOST UNFORTUNATE IF PERSONNEL FROM OUR CUSTOMERS OR EVEN OUR COMPETITORS WERE TO HAVE, THROUGH ACCESS TO THIS FILE, A DISTORTED PICTURE OF OUR SOFTWARE. IN OTHER WORDS, ABUSE COULD CAUSE THIS TOOL TO BE WITHDRAWN, AND I AM SURE YOU DO NOT WISH THIS TO HAPPEN.

P. A. QUANTZ, MANAGER
PROGRAMMING OPERATION

$2. END OF PHOENIX TRANSMISSION.....

READY

SUBJECT: Efficient Use of GE-600 COBOL

Introduction

It is a well known fact that compilation and execution times for COBOL programs are subject to wide variations depending upon how the program is written as well as the operational environment in which the compiler works. The recommendations given below are based upon the findings of a study recently conducted in Phoenix for the purpose of determining techniques for operating the GE-600 COBOL compiler at the high level of performance for which it was designed. These recommendations are presented in two groups. The first group deals with efficiency of compilation; the second is concerned with improvement of execution times for the object program.

Compilation

1. Preliminary findings indicate that the following file and equipment configuration should be used for most compilations:

   - *1 file assigned to magnetic tape (use a $TAPE card)
   - *3 file assigned to drum (use a $DRUM card)
   - G* file assigned to magnetic tape (use a $TAPE card)
   - COBOL should be on disc not on tape

   The amount of core made available to the compiler has a very significant effect on speed of compilation. In determining the optimum amount of core to specify in a $LIMIT card for the compiler, the following formula should be used:

   $$\text{Optimum store} = 24K + \left(\frac{\text{No. of lines of data div.} - 400}{75}\right)K$$

   That is, 24K should be used with Data Division of 400 lines or less and 1K should be added for each 75 lines over 400.

   When the optimum amount of core store is used, the use of the drum for the *3 file, although still desirable, becomes less necessary.

2. The GMAP list should not be requested except in unusual cases. It is usually not used and costs a great deal of machine time to produce. In many cases, even the COBOL list may be dispensed with.

3. Some installations use the $DATA US card in every deck they compile. This card is not only unnecessary, except when patches to the compiler follow it, but it costs a considerable amount of compilation time.

4. Even though "good programming practices" dictate the breaking of large programs into segments, many programmers still fail to appreciate the savings available to them and refuse to segment. Program segmentation not only increases the ease with which a program can be checked out, but by employing the multi-programming capabilities of GE-600 equipment, total compilation time can actually be reduced since other programs can be running at the same time.

Execution

The following are some points of efficient COBOL programming which pay large dividends in reduced execution times. Some are rather widely known, some are not.

1. The use of Commercial Collating Sequence should be avoided wherever possible. If it is used, a table look-up must be performed by the compiler whenever a pair of alpha characters fail to compare. This takes time.

2. Subscripts should be used sparingly or at least limited to only one level. If more than one operation is to be performed on a subscripted item, it should be moved to Working Storage to avoid repeated retrieval.

3. Use ACCEPT and DISPLAY wisely. These can be great time wasters in an actual operational environment.

4. An alternate area should always be reserved for high frequency input and output files.

5. A series of simple conditionals is generally preferable to an involved compound conditional. Although there is little difference in the amounts of object coding generated, use of lengthy compound conditionals interferes with efficient check out.

6. When information is written on tape, it should be "blocked". The practice of writing one short logical record to a physical record on tape is absurdly wasteful. The same thing applies to writing on discs and the drum.

Further points of efficient COBOL programming practices are discussed in Technical Information Bulletin #600-91.

PREPARED BY:

Wallace Hainlin
Consulting Sales Specialist
GE-600 Line

APPROVED BY:

Charles V. Hoge
Product Manager
GE-600 Line

/dew

6

6/27/66

COBOL PERFORMANCE REPORT

Hardware Configuration		Software System	
Store _____ μsec	# Drums _____	Cobol Version	_____
Store _____ k	# Discs _____	Date of Version	_____
Tape _____ BPI	# Tapes _____	Date of Run	_____

Source Program Run Ident. # Stmts - E & D Div. - Proc. Div. - Total											
U/M - Programming Alloc. - Source Prog. - * 1 - * 3 (scratch) - G* - COBOL - Core Store											
GEMAP Included? GEMAP List On? Copy Module In? COBOL List On?											
Timing		Proc.	Elap.	Proc.	Elap.	Proc.	Elap.	Proc.	Elap.	Proc.	Elap.
COBOL - Translation	Sec										
- Analysis	Sec										
- Generation	Sec										
- Make List	Sec										
COBOL - Total	Sec										
GEMAP	Sec										
Total	Sec										
COBOL-Total	Min										
Total (with GEMAP)	Min										
Throughput	Min										
Stmts/Min of COBOL Elap.		X		X		X		X		X	
Stmts/Min of Total Elap.		X		X		X		X		X	
GEMAP ÷ COBOL Time											
# Tape Alerts # Reader Alerts # Printer Alerts # _____ Alerts											

Installation _____ Reporter _____

GENERAL ELECTRIC

Computer Equipment Department
Phoenix, Arizona

DIAL COMM 8*433_____
MAIL DROP_____

SUBJECT
- 600 COBOL Performance Study, Preliminary

COPIES:

1966 June 21

TO: J. W. Weil

FROM: R.W. Bemer

Your letter of June 15 requested me to undertake this study. Due to previous commitments of the people concerned, Mr. Quantz was not able to call the first meeting until June 16 at 1530. This is the preliminary report you asked for by June 22 at 0900.

For the testing plan I enlisted the aid of Clyde McGuffie, Leroy Ellison and John Wertz did all the actual work. Both are excellent programmers and very cooperative. Wally Hainlin from 600 Sales assisted in the evaluation. Despite a clear definition from you of the relative machine time priority for this project, we were able to obtain a reasonable amount of time over Saturday and Sunday, though not at optimum hours.

Experiments Conducted

1) Using a single customer (Martin Denver) program, primarily in the uniprogramming mode, vary the facilities employed by the several elements of the process among disc, drum, tape, store size allocated, and with various processor features operative and inoperative. Runs 3, 3B and 3C are identical to establish where non-reproducibility might occur and in what amounts. Mr. Dobbie had warned of variance due to hardware performance degradation, which could invalidate comparative measurements.

2) Using a single program, add the procedure division to itself several times to correlate compile time to linearity of procedure size.

3) Using a single program, process multiple copies concurrently by multiple copies of COBOL. Vary by a) 2 sources of disc, b) source each on disc and drum, c) 3 sources on disc. This was designed to detect queueing problems for further investigation, as well as indicate what advantage might be obtained by segmenting and doing concurrent COBOL processing.

4) Using DTEST (written by G. Stephens), get the comparative compilation, run, and load-and-go times for 7094, 360-50 and 600.

GENERAL ELECTRIC

Preliminary Findings

1) The drum is so cluttered with software that it preempts space which may be used better for working store. This is a most important requirement. Ellison stated in his 1966 Jan. 27 memo that "Our use of *3 as a random file was planned with very clear management understanding that unless COBOL has *3 on drum, its performance will be lousy."

   Cases 1B and 2B are identical (thus the same processor time) except that 1B has *3 on drum, whereas 2B has it on disc. On drum, the elapsed time is reduced from 12.42 to 5.48 minutes. The workers at EDF report that .55 minutes can be saved by having the COBOL processor itself on drum, but this is clearly of lesser importance than having the working files there.

   Amelioration

   a) Programmers have taken advantage of the capability to make object code patches via the same relocation software utilized for the basic processor stored on drum. Each patch takes two units of the store by granularity. This causes exorbitant wastage of valuable drum store for basic software and leaves little, if any, for working space (we had difficulty allocating scratch files to drum to even demonstrate this point). Yet these same processors are clean in the store. It is not difficult to read the clean copy from the store to the drum! Gillette includes this in his project, and it should be aided and accelerated.*

   b) No facilities exist presently to put source files on disc and leave GECOS on drum (confirmed by Gillette), thus wasting further drum space without reason. A source program does not need drum speed because its intermediate and final output are usually (and desirably) much larger than the original, and each must be processed several times. Gillette is also fixing this, and priority will be desirable.

2) The total COBOL system is of archaic design in that the output is forced to run thru the standard assembly program GEMAP. So far we have measured up to an 80% additional penalty for this requirement!

   We can assume that the assembly can be performed integrally in equivalent (or perhaps less) time to that required for forming GEMAP source statements in string form for that processor. Comparing cases 3C and 3H, identical except that the GEMAP process was not included in the latter's elapsed time, we find a 79% penalty. 3C takes 180% of the running time that it should! 335 statements a minute vs. 600.

   Amelioration

   a) This decision can be reversed and an integral assembler written for COBOL. A preliminary estimate of 18 man-months and 8-10 months elapsed time has been given by Ellison. (Note that this same problem applies to FORTRAN).

* Of course, even under the present system this can be eliminated by keeping the processors in correct source form, as they should be.

GENERAL ELECTRIC

Preliminary Findings (Contd)

Amelioration (Contd)

b) However, note in that same case that the processor time was only 42% additional. We require more evidence, but it could be that this reflects the fact that COBOL files are largely buffered, whereas the GEMAP files are not. This may reflect an original intuition that buffering is not necessary in a multi-programming system. I feel that we can demonstrate that they are necessary.

3) Elapsed time is rather sensitive to various combination of file allocation. Compare worst case 8 (17.64 minutes) against best case 3H adjusted by .55 for COBOL on drum and .11 saved by not bringing in the COPY module unless the customer states that he used this function in his program (2.46 minutes). Even in these few cases we then have a variation of over 7 to 1! 760 statements per minute vs. 105!

In particular, compare 5 to 1B for only 25% improvement going from 24 to 40K when *3 is on drum vs. at least (6.12 figure is high) 100% improvement with *3 on disc, SHOWN COMPARING 6 TO 2B.

Amelioration

a) Let the A.E's understand these effects and advise customers for more productive practices (with a new type of system like the 600, it is likely that customers will not understand these different balancing techniques). Publish broad recommendations. Leave the timing subunits in the COBOL processor and add to other processors.

b) Make further studies to measure queueing effect for several files of the same process on the same device. It may be that some form of dispersion control is necessary.

4) There is a certain penalty for the added intelligence and annotation in this COBOL processor. There is 20-30% more file bulk and this costs elapsed time.

Amelioration

Offer (to customers still complaining after other items are fixed) to remove these facilities if they desire. None will.

5) Significant degradation of elapsed time was noted whenever SYSOUT or input media conversion was a concurrent program. Note different timings for identical pairs EDMOVE runs 1-2, 3-4, which were loaded sequentially in card reader.

Amelioration

Clean up drum and put hardware fix for printing blocks of lines. The UNIVAC 1107 required only 30 seconds of processor time per 8-hour shift to drive a printer!

GENERAL ELECTRIC

Preliminary Findings (contd)

6) Using full processor facilities for every compilation significantly degrades throughput.

Amelioration

a) Cut off object program deck production in early compilations. It won't be useful anyway.

b) Cut the assembly list, even if it were to be integral in COBOL. It is not good practice to lean on this listing; it is for emergency only.

c) In certain cases even the COBOL list can be cut (i.e., one final change which can be noted on the previous listing).

d) The COBOL listing may be abridged (requires additional programming) to become an analyzer, listing only statements with mistakes. Correct all before turning COBOL list on again.

7) Certain modest and easy corrections and improvements are now known for the processor itself, and this is a suitable time. This could lead to 10-13% improvement in processor time. The subtimings used for these tests have sparked new buffering ideas in the COBOL group.

8) We were completely unable to do the third experiment. The startup deck turns out to have been maladjusted so there was an extra copy of GECOS on the drum. (Every user has been taking this 30-link penalty). Whether or not this was the cause, as Gillette surmises, we were unable to run in a multiprogramming mode. This test will be run later.

9) We do not have results of the fourth experiment worth showing. There was insufficient time to match up conditions for measuring alike on the several machines. The New York office of the Service Bureau Corporation does not know how to run COBOL on the 360-50, let alone provide facilities to measure performance and elapsed time.

GENERAL ELECTRIC

Other Observations

1) GECOS is thinly armored and can be destroyed easily, causing a time-consuming reboot of the system from the source to the drum operating form. $SELECT did this three times in a row (an obvious malfunction, reported).

2) GECOS has no machine time accounting facilities of any adequacy.

3) GECOS does not indicate what facilities were used for temporary or permanent storage of files. This is desirable both for measurement purposes and for knowing which file may be bad on an alert or abort.

4) GECOS logs the real usage of facilities, but does not log what is reserved but unused (thus forbidden to someone else, so shouldn't the reserver pay?)

5) If printed output is bad, operator cannot stop the printing operation until run-out.

6) Hardware alert messages, unless specially provided for, can be embedded in working listings, without information about what file was affected.

7) COBOL has a maintenance test module, but no other component of this system does.

8) SYSOUT will drive only 2 printers, although there are 4 on each system in our shop. I understand that this bottleneck has been with us for a year and a half, during which time the tape and disc versions of SYSOUT have been re-written twice - surely unsuccessfully both times if they drive printers at only 500 LPM. The hardware fix may help this considerably.

9) Someone should measure read alerts for Globe Ticket stock in comparison to other.

10) The only 519 available to programmers is reported in continuously poor condition, for at least two years. The word is out to use the 1004 instead if you can get it. Another 519 would not be too expensive.

11) $SELECT apparently only works once for a given program (obvious malfunction, reported). Some other combinatorial uses of this destroyed GECOS.

12) Multiprogramming sometimes fails to result.

13) I estimate we got only 30% useful time in our testing due to system flaws.

GENERAL ELECTRIC

Recommendations

1) Give top priority (even over new system production) to Gillette to force acceptance of only clean processors on drum and allow more flexible allocation.

2) Order a general cleaning up of all major processors to a clean source form with correct and matching listings as of a particular date. Reassemble and incorporate after suitable testing as Change Letter 8. These patched forms are lowering overall performance.

3) Set deadline for this work based on estimate from Gillette.

4) Continue present testing, expanding to FORTRAN and other processors. Use selected, knowledgeable A.E's under direction of W. Hainlin to give field a feel of the balance.

5) Compare effects of this recent mode of evaluation with the simulation type of evaluation being done by Dobbie.

6) Since this recent study did not get many measurements on object time performance (which cannot be done without comparisons on other machines), set up a program for outside machine time. Garnett Stephens of Evendale has 12-15 programs running on both 7094 and 600 which he feels are suitable for comparison (of a total of 84).

7) Accelerate the formation of an inhouse Quality Assurance group under a really competent technician constitutionally suited for high performance as his main goal.

8) Bring Marketing into picture to produce general guidelines to users on Good Practice and how to get the most efficiency out of a new type of system. Aren't AE's supposed to educate the customer? Where do they get educated in this area?

9) Differentiate malfunction (vulgarly, bug) reports from performance complaints. Control these complaints by a formal reporting method to filter out rumor, prejudice and apples-oranges comparisons. React to spurious and system-imbalance complaints by prompt advice to customer.

R Bemer

COBOL PERFORMANCE STUDY

SOURCE PROGRAM		MARTIN-DENVER EA-003 →									
TIME CONFIG - STORE		128 K →									
- TAPES		16 @ 800 BPI →									
- DRUMS		1 →									
- DISCS		2 (200) →									
RUN IDENT		1	1B	2	2B	2C	3	3B	3C	3H	3I
U/M-PROGRAMMING		U →									
ALLOC - SOURCE PROG.		DRUM →									
ALLOC - #1		MT →									
- #3 (SCRATCH)		DR	DR	DS →			DR →				
- G*											MT
- COBOL		DS →									
- CORE STORE		24 →					32 →				
# STMTS - E+DO		1038 →									
- PROCEDURE		837 →									
- TOTAL		1875 →									
GEMAP ?		YES →								NO	YES
GEMAP LIST?		YES	NO	YES	NO →						
COPY MODULE?		YES →									
COBOL LIST?		YES →									
TIMING - C-TRANS	S	69.9	69.8	71.8	71.2	SAME CONDITIONS	44.9	44.8	44.7	44.5	44.1
C-ANAL	S	28.4	28.3	28.6	28.4		28.8	28.5	28.4	28.3	27.9
C-GENR	S	22.9	22.9	23.1	22.9		23.3	22.9	22.9	22.9	22.8
C-MAKE LIST	S	18.0	18.0	17.9	17.9		19.5	19.3	19.4	19.3	19.4
C-TOTAL PRT	S	139.2	138.9	141.4	140.5	140.3	116.4	115.4	115.5	115.0	114.3
GEMAP PRT	S	71.0	98.6	71.3	48.5	48.7	48.8	47.9	48.7	.2-	48.4
TOTAL PRT	M	3.50	3.13	3.55	3.15	3.15	2.75	2.72	2.74	1.91	2.71
C-ELAPSED	S								←	3.12	
GEMAP-ELAPSED	S									0	
TOTAL ELAPSED	M	6.18	5.58	13.20	12.42	13.14	6.00	5.70	5.58	3.12	4.20
THRU TIME	M							13.0			
STMTS MIN OF C ELAPSED		300	335	140	150	145	310	330	335	600	445
" " " TOTAL "											
GEMAP PRT / COBOL PRT		.51	.35	.50	.35	.35	.42	.41	.42	-	.42
GEMAP E / COBOL E									.79	0	
# CARD READER ALERTS		0					4				
# PRINTER ALERTS		0		2			1				

COBOL PERFORMANCE STUDY

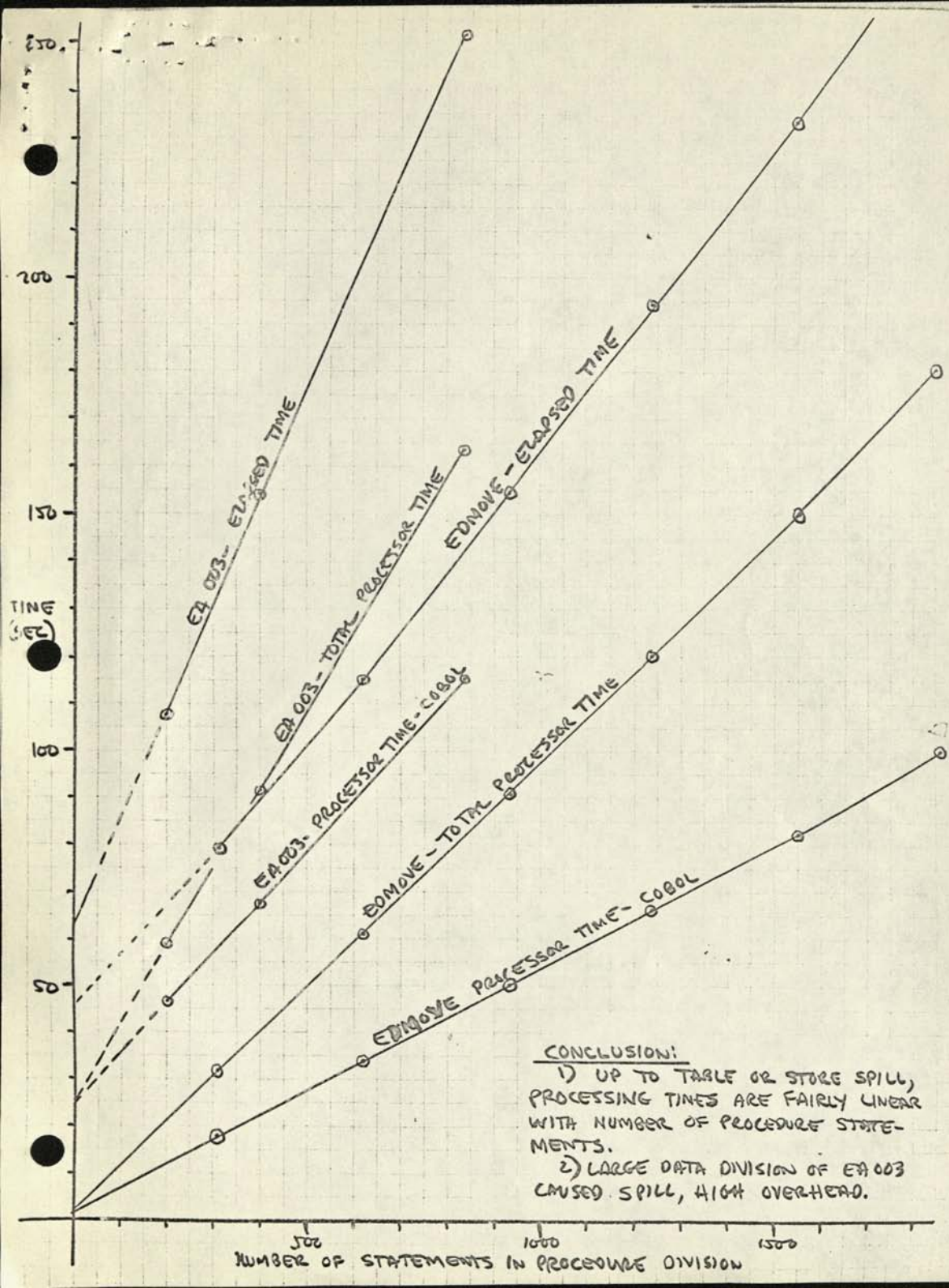
SOURCE PROGRAM		MARTIN-DENVER EA-603 →									
HOME CONFIG - STORE		128K →									
- TAPES		16 @ 800 BPI →									
- DRUMS		1 →									
- DISCS		2 →									
RUN IDENT		3J	3N	3φ	4	4B	5	6	7	8	9
U/M-PROGRAMMING		M*	U →			M*	U →				
ALLOC - SOURCE PROG.		DR →									
ALLOC - *1		MT →									DS
- *3 (SCRATCH)		DR →			DS →		DR	DS	DR	DS →	
- G*		MT				MT					DS
- COBOL		DS →							MT	MT	DS
- CORE STORE		32 →					40 →		24 →		32
* STMTS - E + DO		1038 →									
- PROCEDURE		837 →									
- TOTAL		1875 →									
GEMAP ?		YES →									
GEMAP LIST?		NO	~~YES~~	~~YES~~	NO →						
COPY MODULE?		YES	NO	NO	YES →						
COBOL LIST?		YES →									
TIMING - C-TRANS	S	44.6	37.5	SAME CONDITIONS	45.3	(NO TIMES IN MT VERSION FOR MULT)	44.9	44.8	(NO TIMING IN MT VERSION)		
C-ANAL	S	28.1	28.3		29.0		28.5	28.5			
C-GENR	S	23.0	22.9		23.1		22.9	22.9			
C-MAKE LIST	S	19.4	19.5		19.1		20.1	20.1			
C-TOTAL PRT	S	115.1	108.2	108.1	116.5		116.6	114.3			116.9
GEMAP PRT	S	48.4	71.1	71.3	48.7		48.3	48.3			53.8
TOTAL PRT	M	2.72	2.99	2.99	2.75	2.80	2.75	2.74	3.34	3.37	2.84
C-ELAPSED	S										
GEMAP-ELAPSED	S										
TOTAL ELAPSED	M	4.62	4.62	4.74	6.00	5.34	4.44	6.12	11.52	17.64	6.78
THRU TIME	M	~~4.62~~			~~X.00~~	~~5.34~~					
STMTS MIN OF C ELAPSED		405	405	400	310	350	420		160	105	280
" " " TOTAL "											
GEMAP PRT / COBOL PRT		.42	.66	.66	.44	.44	.43				.46
GEMAP E / COBOL E											
# CARD READER ALERTS					3		0	MANY			
# PRINTER ALERTS					2		0				

COBOL PERFORMANCE STUDY

SOURCE PROGRAM		M-D EA023	→	→	→	EA05	BONES
HOME CONFIG - STORE							
- TAPES							
- DRUMS							
- DISCS							
RUN IDENT		1D	3D	3E	3G		
U/M - PROGRAMMING		U	→	→	→	→	U
ALLOC - SOURCE PROG.		DR	→	→	→	→	DR
ALLOC - #1		DS	MT	MT	MT	MT	
- #3 (SCRATCH)		DS	DR	→	→	→	DS
- G*		DS				MT	
- COBOL		DS	→	→	→	→	DS
- CORE STORE		24	32	→	→	→	32
# STMTS - E + DD		1038	→	→	→		32
- PROCEDURE		837	200	400	837		7
- TOTAL		1875	~~1238~~	~~1438~~	~~1875~~	3672	39
GEMAP ?		YES	→	→	→	→	YES
GEMAP LIST?		NO	→	→	→	→	YES
COPY MODULE?		YES	→	→	→	→	YES
COBOL LIST?		YES	→	→	→	→	YES
TIMING - C-TRANS	S	71.4	27.0	33.4	44.6	152.2	1.0
C-ANAL	S	28.8	6.0	12.3	28.3	54.7	.1
C-GENR	S	23.6	6.7	11.9	22.9	52.5	.4
C-MAKE LIST	S	18.1	7.2	10.1	19.4	37.2	.2
C-TOTAL PRT	S	142.0	46.9	67.7	115.2	296.6	1.7
GEMAP PRT	S	53.9	12.5	24.1	48.7	116.3	3.3
TOTAL PRT	M	3.26	.99	1.53	2.73	6.88	.08
C-ELAPSED	S						
GEMAP-ELAPSED	S						
TOTAL ELAPSED	M	9.48	1.80	2.58	4.20	11.22	.72
THRU TIME	M						
STMTS MIN OF C ELAPSED							
" " " TOTAL "		250	690	560	445	327	54
GEMAP PRT / COBOL PRT		.38	.27	.36	.42	.39	1.94
GEMAP E / COBOL E							

COBOL PERFORMANCE STUDY

SOURCE PROGRAM		EDMOVE								
HOWE CONFIG - STORE		128K								
- TAPES		16								
- DRUMS		1								
- DISCS		2								
RUN IDENT		1	2	3	4	5	6	7	8	9
U/M - PROGRAMMING		U								
ALLOC - SOURCE PROG.		DR								
ALLOC - *1		MT								
- *3 (SCRATCH)		DS	→	DR						
- G*				MT						
- COBOL		DS								
- CORE STORE		32								
# STMTS - E + DO		62								
- PROCEDURE		310			→	620	930	1240	1550	1860
- TOTAL		372			→	682	992	1302	1612	1922
GEMAP ?		YES								
GEMAP LIST?		NO								
CORY MODULE?		YES								
COBOL LIST?		YES								
TIMING - C-TRANS	S	5.0	4.9	4.9	4.9	8.4	11.9	15.6	19.2	22.9
C-ANAL	S	5.0	4.9	4.9	4.9	9.6	14.4	19.1	23.9	28.6
C-GENR	S	5.9	6.0	6.0	6.0	11.5	17.1	22.6	28.2	33.8
C-MAKE LIST	S	2.3	2.3	2.3	2.3	4.5	6.7	9.0	11.3	15.1
C-TOTAL PRT	S	18.1	18.0	18.0	18.0	34.0	50.1	66.3	82.6	100.3
GEMAP PRT	S	14.3	13.8	13.8	14.1	27.2	40.6	53.9	67.5	81.1
TOTAL PRT	M	.54	.53	.53	.53	1.02	1.51	2.00	2.50	3.02
C-ELAPSED	S					70.2				
GEMAP-ELAPSED	S					41.4				
TOTAL ELAPSED	M	1.50	1.26	1.74	1.32	1.92	2.58	3.24	3.90	4.62
THRU TIME	M					↳1.86 w/o MAG TAPE ALERT				
STMTS MIN OF C ELAPSED		250	295	215	280	355	385	400	415	415
" " " TOTAL "										
GEMAP PRT / COBOL PRT		.80	.77	.77	.80	.80	.81	.81	.82	.81
GEMAP E / COBOL E						.59				
# MAG TAPE ALERTS						1				


250
200
150
TIME (SEC)
100
50
EA 003 - ELAPSED TIME
EA 003 - TOTAL PROCESSOR TIME
EDMOVE - ELAPSED TIME
EA003 - PROCESSOR TIME - COBOL
EDMOVE - TOTAL PROCESSOR TIME
EDMOVE PROCESSOR TIME - COBOL
500
1000
1500
NUMBER OF STATEMENTS IN PROCEDURE DIVISION
CONCLUSION:
1) UP TO TABLE OR STORE SPILL, PROCESSING TIMES ARE FAIRLY LINEAR WITH NUMBER OF PROCEDURE STATEMENTS.
2) LARGE DATA DIVISION OF EA 003 CAUSED SPILL, HIGH OVERHEAD.

Applied Programming
DSDHQ
December 18, 1959

R. W. Bemer

Supervisory Languages and Systems
Meeting

The ACM Committee on Programming Languages has set up a sub-committee on Supervisory Languages and Systems, chaired by R. W. Bemer. It is planned to have a symposium to discuss and extend invited papers which, by that time, will have been published in the Communications of the ACM.

It is felt that an interchange of ideas on the general subject of supervisory systems between various interested IBM programming groups, prior to establishment of an industry-wide effort, is desirable.

Such an internal meeting will provide ample opportunity to acquaint each other with our individual aims. An attempt will be made to formulate a standard approach to supervisory systems. The even more basic question of their desirability and scope will be open for discussion.

This meeting will provide the background for IBM's participation in the Supervisory Languages and Systems Symposium.

In view of the purpose of the meeting, the discussion should be fairly specific rather than general. It has therefore been decided to hold down the number of attendees. We call upon your discretion in inviting any of your qualified people.

The meeting will convene at 10:00 a.m. Monday, 28 December, in Mr. Bemer's office on the fourth floor of the Time-Life Building in New York City.

Please consider this your invitation to attend.

The panel will be composed of the following or their representatives:

R. W. Bemer	I. C. Liggett
E. F. Codd	D. L. Mordy
C. E. Diss	F. A. Williams
W. P. Heising	H. B. Williams
M. Held	A. S. Wolf

F. A. Williams

faw/ep

ElectroData
DIVISION OF BURROUGHS CORPORATION
460 SIERRA MADRE VILLA PASADENA, CALIFORNIA

October 24, 1958

Mr. Robert Bemer, Manager
Programming Systems
International Business Machines, Incorporated
590 Madison Avenue
New York, New York

Dear Mr. Bemer:

I am pleased to reply to requests for information from the office of Techniques Editor of Communications concerning Burroughs automatic coding tools and simulators.

Contrary to your impressions regarding program librarial facilities of DUO and Burroughs Corporation, I believe you will find that a very well monitored exchange provision does exist. May I be of assistance to remove this impression to better represent the support efforts available to Burroughs customers.

Included below is a list of programs in operation or under development known to this office that may interest Communications readers:

BURROUGHS 205	AUTHOR
Datacode: A 3-address compiler	Burroughs Corporation
Purdue Compiler: A formula-translating compiler	Purdue University
Dow Compiler: A 2-address compiler	Dow Chemical
SAC: A numeric assembler-compiler	Burroughs Corporation
STAR 0: An alphanumeric assembler-compiler	Burroughs Corporation
Dumbo: A numeric assembler	Babcock-Wilcox Company
UGLIAC: An alphanumeric assembler	United Gas Company

SPAR: An alphanumeric assembler-compiler	Cooperative Wind Tunnel (California Institute of Technology)
Shell Assembler: An alphanumeric assembler-compiler	Shell Development Company
Shell-Bell Interpreter: A 3-address interpreter	Shell Development Company
ANCP: An alphanumeric assembler	Naval Ordnance Laboratory
* FORTRAN Compatibility	Naval Ordnance Laboratory
* Debugging Generator	Burroughs Corporation

SIMULATORS

The following information is based upon the listed questions by Mr. Helt:

1. Approximate speed ratio of simulator program to machine being simulated.
2. Program limitations
3. Purpose of the program
4. Authorship
5. Completion date

A. BURROUGHS 205 on BURROUGHS 220

1. Simulation proceeds at one to two times the speed of the Burroughs 205. This can be increased to from five to fifteen with minor 220 machine-language program substitutions, ignoring resultant input-output requirements, if any.
2. None

---

* Under construction

3. Production runs during changeover

4. Burroughs Corporation

5. August, 1958

B. BURROUGHS 220 on BURROUGHS 205

1. Simulation proceeds at 1/170 the speed of the 220. 140/200 INSTR/MINUTE

2. Data input from paper tape only. Magnetic tape is not simulated.
SIM USES 1200 WORDS, CV 2800 FOR 220 PROG, NEED FLT PT FOR FLT PT.

3. Program testing prior to Burroughs 220 installation

4. Burroughs Corporation

5. February, 1958

C. IBM 650 on the BURROUGHS 220

1. Simulation proceeds at one to two times the speed of the IBM 650. This can be increased to from five to fifteen with minor machine-language program substitutions, ignoring resultant input-output requirements, if any.

2. RAM units, magnetic tape, and inquiry station instructions.

3. Production runs during changeover

4. Burroughs Corporation

5. August, 1958

D. IBM 650 on the BURROUGHS 205

No pertinent information from the customer-author is available.

Programming research and development at Burroughs is being increasingly emphasized. Beyond the aspect of customers service, most certainly it is our intent to contribute such developments to the entire profession.

Would you, to insure proper liaison between Burroughs Corporation and the Techniques Department, kindly inform me of advance subject matter so as to permit Burroughs to contribute where possible and, at the same time, remove the apparent communications barrier.

Sincerely yours,

C. Ricker

C. L. Ricker, Manager
Applied Programmer

CLR:ja

Distribution: OEMI Headquarters
All Members of X3 Sectional Committee
All Subcommittee Chairmen of X3

OFFICE EQUIPMENT MANUFACTURERS INSTITUTE

777 FOURTEENTH STREET N.W., WASHINGTON, D. C.

Reply to: Data Processing Group
Room 2814 - Graybar Building
420 Lexington Avenue, New York 17, New York

August 18, 1960

Mr. H. S. Bright
Westinghouse-Bettis Laboratory
Post Office Box 1468
Pittsburgh 30, Pennsylvania

Dear Mr. Bright:

Thanks for the ACM material you sent me. I am distributing to all X3 members and subcommittee chairmen copies of this for their information and coordination.

With this, I am sending you copies of: (1) Data Processing Group Engineering Committee Minutes re formation of X3 Sectional Committee and (2) Minutes of initial X3 Sectional Committee Meeting.

You will see that ACM was invited to be represented officially on X3, that Mr. R. W. Bemer was designated by ACM as "pro tem" representative until your Council selects its representative and that he attended the X3 meeting.

You will note that a glossary and definition of terms is included in the assignment to a subcommittee of X3. Also ASA procedures provide for liaison between and active coordination of all ASA committees working in related (or even similar) areas - such as a DP glossary. There need be no friction nor lack of cooperation for we stand ready to work actively with any group interested in any aspect of data processing standards, in fact we welcome it.

Sincerely,

Joseph W. Barker
Consultant and Acting Director

JWB:ie

Reprinted from The Computer Bulletin, Vol. 6, No. 2

AUTOMATIC PROGRAMMING LANGUAGES FOR BUSINESS AND SCIENCE

Reported by Daphne Kilner

[*A Conference under this title was held on* 17–18 *April* 1962 *by the Mathematics Department of the Northampton College of Advanced Technology in co-operation with the British Computer Society. The following is a summary report on the Proceedings which will be published in full in the Computer Journal.*]

Aims

What do we want from these Automatic Programming Languages? This is a more difficult question to answer than appears on the surface as more than one participant in the recent Conference of this title made clear. Two aims are paramount: to make the writing of computer programs easier and to bring about compatibility of use between the computers themselves. Towards the close of the Proceedings one speaker ventured that we were nowhere near achieving the second nor, indeed, if COBOL were to be extended any further, to achieving the first.

These aims can be amplified. Easier writing of programs implies that they will be written in less, perhaps in much less, time, that people unskilled in the use of machine language will still be able to write programs for computers after a minimum of training, that programs will be written in a language more easily read and followed, even by those completely unversed in the computer art, such as business administrators, that even the skilled in this field will be relieved of the tedium of writing involved machine language programs, time-consuming and prone to error as this process is. Compatibility of use will permit a ready exchange of programs and applications between installations and even of programmers themselves (if this is an advantage!), for the preparation of programs will tend to be more standardised as well as simplified. Ultimately, to be complete, this compatibility implies one universal language which can be implemented for all digital computers.

History and Development

A Potted History of Automatic Programming was circulated at the Conference and is reproduced here as an Appendix. This shows that early ideas on a universal language were first voiced in 1956–57, but that since then the line of development has forked into two such languages: mathematical and commercial. Whilst many languages of both types have been specified during the course of this development (FORTRAN and AUTOCODE among the former, FLOWMATIC, FACT and NEBULA among the latter), only two were universal by design, the others being always orientated towards certain computers: these two are ALGOL, a universal mathematical language, and COBOL, a universal business language. Both have been the subject of continual revision (e.g. COBOL 60 and COBOL 61 designate successive annual versions of COBOL) and their advent has caused the spawning of ALGOL-like and COBOL-like languages.

The Potted History refers to the time lapse between the specification of these, the Source Languages, and their implementation in actual programming systems to produce the Object Language (generally and ultimately in the actual computer instruction code but intermediately this could be in another pseudocode before final conversion into machine code). A certain play was made of this time lapse feature also in the titles of a few of the papers presented. The missing link which accounts for it is the development of the Compiler: a source language cannot be used to the full until a compiler has been written to implement it for a particular computer. The success of an automatic programming system depends on both the source language and the compiler: incomplete definitions in the former will limit the efficiency of the latter and the latter serves to define more closely specifications given in the former. So there were papers at the Conference on the source languages themselves and other papers on the development and use of compilers: there were views expressed on the ideal source language and other views on the different methods of writing compilers.

The Ideal Source Language

What are the properties of an ideal source language? Which are the important aspects to consider in an Automatic Programming Scheme? In establishing such a scheme we are trying to shift the burden of problem-solving from the human being to the computer. Therefore, such a language should be easy to learn and simple to use. Not all the existing published languages have achieved this double purpose, so much so that the paradoxical statement was made, and supported, that at present it is the machine code which is easy to learn but hard to apply and the automatic programming language which is hard to learn but eventually easy to use.

Moreover, such a language must be general: a source language must apply to the widest possible range of problems to be of the greatest value, although at the moment no satisfactory way has been found of attempting to place both scientific and commercial problems within the framework of one universal source language. This ideal source language must be concise with as few words as possible and using identifiers as far ranging as possible. It must do a maximum amount of work with a minimum of commands. Through this it will in fact tend towards symbolic languages but not in such a way as to make its resulting simplified programs difficult for the layman to follow. It should obey the cardinal programming rule of absolute precision but at the same time should avoid any unnecessary complications which will make

the construction of the compiler more difficult. Finally, it should be as natural and unforced a language as possible. A comparison was made here with the way mathematical notation had grown up over the years. It had not been especially invented, but it had evolved into the highly apt vehicle of expression that it now was. Something of the same long evolution might have to take place before the far subtler ideas of automatic programming could be naturally expressed.

This necessity for natural expression applies to the description of the types of information involved as well as of the problem-solving procedures. These two fundamental aspects have to be considered in framing any kind of programming scheme, but, in addition, two others arise through trying to use an automatic programming scheme: how is communication obtained with the object program after it has been compiled, in particular, how is it to be debugged? and will the compiler work on the actual machine the user wishes to employ?

The types of information involved will be the input data themselves and the final results to be obtained from these, together with any necessary intermediate results. The ideal source language must be capable of expressing any plan for the input of data, including the layout of records in the associated backing files and also any form of presentation of results. The basic properties available for data division are the chief source of power in an automatic programming scheme and are probably the chief causes of the distinction between mathematical and business source languages. Are data to be expressed as fixed length units, easy to move, or as variable, although static, length units, which are more difficult to move? Or perhaps as dynamic variable lengths (i.e. changing in the course of the problem), more difficult to move still? Records in files have to be broken down through various levels to individual items which may be of only one character and these must be conveniently identified. Does a universal source language therefore need facilities for individual character manipulation? Likewise, the presentation of final results requires detailed output facilities in the source language, facilities for indicating style and layout, repeats and alternatives, for consistency checks on the information, and many other features necessary for succinct reporting.

Procedures for problem-solving are dominated by the types of command available and the rules of grammar employed. [The word procedure is used advisedly: despite some claims it was felt that most languages, including COBOL, were still procedure-orientated only and none had yet reached the happy state of being problem-orientated, that is, assisting "the expression of problems without describing the precise means of solution" (d'Agapeyeff: see Appendix).] Commands should be variable, flexible and general in type. The difficulty here arises in incorporating new functions into the language. Not all users need the same types of command, some may even need special-purpose commands, and so there must be a basic repertoire of commands with simple facilities for adding to it. Most present source languages have a lack of such adequate facilities, or employ cumbersome and difficult methods in their stead. ALGOL is the exception here.

Scant attention has been paid to the problem of communication with the object program after it has been compiled and programs produced automatically are notoriously difficult to debug. The only solution is to provide means of debugging the program in the language in which it was originally written, i.e. the source language. Manufacturers are wont to say that any fool can now write programs for their machines but it still takes a different type of fool to debug them.

A source language must be general, as has been said, in that it must apply to the widest possible range of problems. But it must also be machine-independent, it being left to the compiler to take account of the features and configurations of individual computers. Most source languages include a section which describes the facilities necessary in the computer on which the Object Program is to run and the compiler translates the source program in terms of these. No one compiler can cover every machine and the question is how wide a range of machines can it fit. Compilers are difficult to make, and probably as difficult to keep up to date. New facilities and new equipment are continually added to existing computers so the compiler must be continually adapted to incorporate these, otherwise the object program is less efficient than it should be. The success of the compiler is measured by the efficiency of the object program, and this efficiency will vary with every different configuration of equipment employed. The job of the compiler-writer is never done.

It was objected that an ideal universal source language along these lines was impossible. If it were truly general for greater ease of use, it could not fail to be difficult to learn, its compilers would be complex and difficult to make and it could not be effective over the whole range of machines. Can a source language of great general power be effective on a small machine, for example, without loss of generality? It was suggested during the course of the Conference that the importance of the source language known as LANGUAGE H was that it was designed for use with a small machine. But does this mean that there must always be a range of local languages to take over where the large universal language leaves off?

Development of Particular Source Languages

The development of source languages has been dominated by controversies first over ALGOL and later over COBOL. There was early opposition to ALGOL, for example from IBM, but a more stable state of affairs was reached with the establishing of ALGOL 60 for which a primer of programming has now been published.* Three examples of the implementation of ALGOL 60 were described to the Conference: that at the Mathematisch Centrum, Amsterdam, the development of the Elliott ALGOL 60 compiler for use with their 503 computer and the work of English Electric at Kidsgrove in relation to their KDF 9.

At Amsterdam, the aim was to make an automatic programming system which worked and initially they were not greatly concerned with its efficiency. A small Electrologica X–1 computer was used, with a 4,000-word core store and no backing store, and slow punched paper tape input and output. A special Flexowriter was adapted to the production and correction of source program tapes in ALGOL 60 and this is now on the market. There was no form of syntax checking in the compiler and no printing out of the object program: all corrections must be made to the source program. Neither was any provision made for the assembling of an object program partly from an established library of subroutines in machine code and partly from ALGOL: this

* DIJKSTRA, E. W. "A Primer of ALGOL 60 Programming." Academic Press, N.Y., April 1962. 114 pp. 30s. 0d.

technique of embedding ALGOL programs in surrounding machine code subroutines, however advantageous in speed of assembly, was felt to be most unwise.

The Mathematisch Centrum had had no previous experience with autocodes before implementing ALGOL 60. Nevertheless, their translator has been very well received and has been found to fill a need. They now offer a 4-day course on ALGOL 60 programming, in which so far there have been some 240 participants, some with and some without previous programming experience (and some indeed with no mathematics). In general they have found that handwritten programs are about 10% shorter than their compiled equivalents (although two cases were reported where the ALGOL version proved to be the faster!). This success itself raised difficulties because it opened the way to attempts to program problems too large for the store; this they solved by ordering another 4,000-word module of core store!

The Elliott ALGOL 60 compiler had the distinction of being available for work 6 months before the machine for which it was designed. Since their 503 and 803 computers have the same instruction code all the experimental running could be done on the latter. Their compiler was designed with an eye on their service bureau work where the aim was to run a maximum number of programs to the hour. The ideal for this was a translator, which one could leave permanently in the store, to translate successive source language programs as they came along.

Elliotts ran their first ALGOL program on 15 February 1962, and since then most such programs have run correctly the first time, although some have uncovered errors in the translator or in the dynamic housekeeping routine governing the time-sharing facilities. The compiler has some 8,000 instructions which includes 400 instructions for this housekeeping routine. Two early decisions had been taken in its regard: first that there would have to be some loss of ALGOL generality in order to avoid too cumbersome a compiler and second that no optimisation of the object program was to be attempted except of the most rudimentary kind. The use of recursive techniques, recommended where no optimisation is needed, makes it easy to print out and find errors. The speed of the source program input, 1,000 char./sec. on punched paper tape, was faster than the production of the resulting object code, a disadvantage that would probably not have much effect upon the service work envisaged. Finally, to illustrate the complexity of compiler writing, it was noted that it had taken three people $1\frac{1}{2}$ man-years to make the Elliott ALGOL 60 compiler, with an additional $\frac{1}{4}$ man-year for discussion and planning and another $\frac{1}{2}$ man-year in associated pursuits.

English Electric set out with the aim of implementing the full ALGOL 60 report. At the Kidsgrove service bureau they expected to have a wide range of users and of problems and therefore they needed a language which could be universally understood and implemented, as well as eventual fast and efficient object programs. In fact, they have developed two compilers, one at Whetstone, a fast compiler of 3,000 instructions operating on the one-pass load-and-go principle but with no special attempt at efficiency, and the other at Kidsgrove, a larger affair of some 20,000–30,000 instructions, designed to be more efficient with exceptionally good procedures but involving several machine passes. Both are expected to be ready for full use by the end of 1962. The Whetstone compiler is suitable for one-off programs but that at Kidsgrove is designed for production programs which require a greater efficiency.

These two compilers have certain properties in common. They accept almost identical versions of ALGOL, and that almost the whole of ALGOL and certainly nothing but ALGOL. For both there is an available library of procedures in ALGOL, and in both one can communicate with the machine only in ALGOL. But the Kidsgrove compiler has greater facilities for checking and amending the source program, and also for the final printing out of programs. Finally, English Electric endorsed the opinion that ALGOL was not difficult to learn: their experience was that, including recursive procedures and side effects, a course would not last even the Amsterdam time of four days to an audience of experienced programmers.

To contrast with this experience of ALGOL, a paper was presented on the experience at AWRE on several IBM machines with another, older, mathematical programming language, FORTRAN. Interest at AWRE in this language began about four years ago, but their early experience of FORTRAN I was not very satisfactory due partly to the natural conservatism of mathematicians and partly to the unreality of the compiler (it needed 750,000 words of core store and AWRE had only 8,000 on their IBM 704!), but chiefly to the poor quality of the object programs. While less efficient object programs are nearly always a feature of any automatic programming scheme, one cannot always ignore this inefficiency. FORTRAN I, for example, produced object programs with 10% or 15% more instructions and using 5% more machine time than any machine-coded program. Further, it lacked the ability to create subroutines for incorporation into the source program: one had to write the whole program without subroutines and consequently in correcting it one had to recompile the whole program.

FORTRAN II overcame this latter difficulty and, furthermore, its real value lay in its additional embedding facilities, whereby an object program could be assembled from both FORTRAN and machine code sources. This takes longer and is not the load-and-go principle, but it is invaluable in an establishment already having a good library of machine code routines. FORTRAN II has also been found invaluable in time-limited problems and in those which otherwise could not have been done at all, and AWRE has now adopted the policy of writing all future programs in FORTRAN II. They are not waiting for ALGOL for the simple reason that the compiler for FORTRAN II is available and that for ALGOL is not.

Interesting comparisons were made between ALGOL and FORTRAN II. Both were scientific languages, but both could be used for other purposes by small extensions. ALGOL was the more advanced but nevertheless FORTRAN had greater input/output and embedding facilities and permitted clearer segmentation of programs. In regard to syntax they were two different things altogether: unlike ALGOL, FORTRAN has no definite syntactic rules and the programmer is guided by precepts, his choice very often following a set of precedents, and sometimes the only way is to try it on the compiler and see! This construction allows of very simple statements and can include the use of recursiveness, resulting in a language of considerable power, but its extreme generality can cause difficulties to the average user.

The FORTRAN II compiler, of 40,000 to 50,000 instructions, was definitely slow, chiefly on account of its optimisation features, but it nevertheless now gave a very good output. However, it was an illuminating commentary on the difficulties of making an absolutely reliable compiler that even after three years' use AWRE were still coming across errors in it. They had also drawn some guarded conclusions on the economics

of using FORTRAN II. Every object program instruction derived from it cost one penny to compile, as against ¼d. for such an instruction derived from a handwritten program. Nevertheless, the human costs involved in writing machine code programs far outweigh the extra cost of compilation. The reduction in writing time between the two methods was something of the order of 2 or 3 to 1 for an average programmer: the costs of debugging a FORTRAN II program were also said to be less. All in all one could conclude that good quality programmers were still needed and would save money in an installation even when it was working solely from the FORTRAN II language, but that bad programmers would lose money faster.

While there has been probably even more controversy over COBOL than over ALGOL, it has ultimately gained acceptance quicker, at any rate in the USA. This acceptance was certainly aided by the attitude of the US Dept. of Defence, under whose sponsorship the body responsible for constructing COBOL had been set up (CODASYL—Conference on Data Systems Languages), and who attempted to enforce its use by insisting that all the computers which it purchased (and it is by far the largest owner of computers in the USA) should be able to implement the COBOL language.

In the UK, however, the attitude to COBOL was lukewarm and its acceptance slower for many reasons. The distance between the potential implementers and the source of the COBOL reports led to a lack of drive in obtaining acceptance, as to a certain apparent vagueness and lack of definition about the language itself. Furthermore, the descriptions of COBOL were tape-orientated and in this country the use of tape had been pared to a minimum so that the equipment to implement COBOL was not available and, contrariwise, there was vast capital investment in the present equipment. And this is a conservative country anyway.

A significant point in the history of this acceptance of COBOL in the UK came about 18 months before when a Working Committee of the British Computer Society, set up to study COBOL in detail, advised that they could not recommend the language as it then was, and that experience could as well be gained from the manufacturers' own languages. ICT was the only manufacturer to ignore this advice, the others pursuing in the main, until recently, their own developments.

ICT made a study of COBOL in relation to its readability, its simplicity and its internationality, a study which bore fruit in RAPIDWRITE, for use with their 1301 and 1500 computers. They found COBOL easy to read, an essential feature for data processing managers and systems analysts, but also verbose and difficult for programmers to write. By cutting out about 10% of its marginal, little used, facilities and redundancies, e.g. using the verb compute to replace the four separate words for the arithmetical functions, and by transferring all the superfluous material on to pre-printed forms, thus preserving the readability whilst leaving little for the programmer to write, they reduced the essential features of COBOL to the simpler language of RAPIDWRITE. By doing this the language could be described in an 8-page manual and a training course takes 2 days only as against 10 days for a COBOL course or one month for a machine code course. The Compiler, of some 40,000 words, was expected to start making translations at the end of May 1962.

Can a language be said to be universal when its readability depends on English? The ICT attempt to get over this hurdle was the only one described to the Conference. RAPIDWRITE has three features which can be prepared in any language using the 26-letter alphabet: pre-printed stationery, a format dictionary and a synonym table, the two latter being built into the computer store. By making the appropriate substitutions, a program can even be converted from one language to another. Thus it is available to non-English speaking users, a point of some importance to a manufacturer with a world-wide market.

The Conference heard also of the 6 months' operating experience of COBOL in the English Electric Service Bureau at Kidsgrove. A compiler of 60,000 instructions was ready for use on the KDP 10 in October 1961 (this machine is the English version of the RCA 501 in the USA, the computer upon which COBOL was first used). The compiler has made a feature of dealing with this difficulty of debugging the object program: the system operates in two sections: first the compiler accepts the source program, finds its mistakes and prints out a list of errors in the COBOL itself; then, secondly, after two or three other such machine runs it produces the final object program and, in addition, an edited copy of the source program. The whole procedure is well documented, the computer giving good print outs of both source and object programs.

English Electric have found that COBOL can be learned in three days although two weeks is a more suitable time. Program writing time was reduced by a ratio of 4 : 1 and other side benefits accrued also from the use of COBOL: operating and running procedures become generally simpler with some exceptions (e.g. there is no provision for sorting in COBOL 60) and it comes into its own on file processing problems. There are also options to allow people to put in their own sections, e.g. in this country such a section might be sterling conversion. Nevertheless, it is still true that the best machine-coded program is better than the best COBOL-coded program. English Electric maintain that COBOL is easy to learn, but it is also still true that the better the programmer the better the program.

The advent of COBOL has not stopped the flow of production in other source languages, either here or in the USA, although there was very little experience of their actual use yet in either country. In USA experience in the mathematical field was the greater, with wide experience of FORTRAN although little as yet of ALGOL (in the development of which Europe was regarded as the leader), but as far as commercial languages were concerned, there has been little actual use of FLOWMATIC or of COBOL, and while FACT (the Honeywell language for their 800 and 1800 computers, described in some detail to the Conference) has been in customers' hands for over a year, this was still experience of no great significance as yet.

FACT was started a month before COBOL and developed independently of it: it has many features not found in COBOL. The key to understanding FACT lies in its handling of files to which special attention has been paid. No other language has handled bulk files yet, and this has given FACT a significant role in the development of these commercial languages. In most file structures information is dealt with at various logical levels, groups pertaining to one subject coming under one group heading, and these related group headings being themselves collected under a larger group heading at a higher level. Thus an information hierarchy is arrived at and FACT is constructed to deal with file information on this basis.

In addition to this, FACT is loaded with convenient ways of doing things, including features not available in other languages: for example, handling of punched card input, with particular regard to ease of editing and checking (it is

claimed to have the most powerful input/output facilities of any automatic programming language so far), sorting on magnetic tapes (40% of data processing is reckoned to be made up of sorting and FACT has influenced the later development of COBOL in this respect), and ease of description in reporting.

The FACT compiler has 220,000 3-address machine instructions. It is being improved all the time and will run on the 1800 three times as fast as it does on the 800. Cases were cited where FACT programs were already written and working satisfactorily: one of 30,000 words for payroll, for example, was written in three man/months and had completed eighteen runs since last January. Of the Honeywell 800 customers, five use nothing but FACT, twelve use a mixture of FACT and machine code, and one actually chose a machine to take FACT because it handles paper tape so easily. Experience has already shown that for a compiler of this size, more equipment is needed for the implementation, but it does nevertheless serve to replace unavailable manpower in programming.

The Conference heard of some other commercial languages developed by manufacturers, who did heed the advice given by the British Computer Society, and who had other good reasons for pursuing their own developments. LANGUAGE H, for example, produced by National-Elliott, was commendable because it operated in a region where COBOL was weak, i.e. on a small machine. Little effort is being made to provide such languages for small machines and yet the small user needs them every bit as much as the large.

The first compiler for Language H was produced in August 1961 for the National-Elliott 405M. Its revision, expected to be ready by May 1962, would contain 23,000 1-address instructions. Operational procedures are reduced to a minimum, the instructions being as fully boot-strapped as possible. Comprehensive checking is done on the first run through with the Compiler, after which the object program is printed out together with a list of errors and a "map" of the immediate access store, invaluable in debugging. It was pointed out that this number of instructions would be the equivalent of 11,000 3-address instructions in the FACT compiler, i.e. it was about 1/20 of the size of FACT, but for this it would do 1/18 as much work. Another interesting comparison was made of the expansion ratio between the number of instructions in the source language and the resulting number of instructions in the object language: Language H has an expansion ratio of 4, FACT one of 35.

Ferranti considered that it was essential for their ORION customers to have a good auto-coding system and that COBOL 60 lacked some facilities they regarded as necessary. COBOL tended to be bound by business requirements in the USA, e.g. it is orientated towards character rather than binary machines, and if they had accepted it they would be penalising their customers for a language over which they had little control, besides the actual delay in getting the COBOL decisions through.

Their language NEBULA is similar to COBOL but gives a greater freedom of choice of input/output media with choice of format within those media. It can, for example, take all existing punched card codes and there is nothing comparable in COBOL to provide for layout presentation as there is in NEBULA. A neat solution to the problem of debugging the object program had been found in developing two versions of the compiler, the production and program testing versions. An input parameter chose one or the other. The present compiler for ORION, of some 30,000 instructions, is expected to be ready by the end of 1962 and that for ATLAS is in the course of construction. Both compilers owe a debt to techniques developed at Manchester University.

Some preliminary information was also given on CLEO, the language designed by *LEO Computers Ltd.*, for use on LEO III. This, the most recent of these programming languages, is impressive in that its goal is to cover both mathematical and commercial problems, an attempt which is long overdue. It has features akin to both FACT and ALGOL. The first version of the compiler, now in the last stages of completion, is written in an intercode, using one language to write another. While it is premature to give such information the number of orders in the Compiler is expected to be about 25,000–30,000.

Some other languages were also mentioned in passing to complete the picture of the present situation. In addition to FORTRAN for scientific work, IBM also produced a business language, the COMMERCIAL TRANSLATOR, which has been used with IBM 705, 7070 and 7090. Simple and elegant, with an excellent manual (not always true of other languages) it has, for once, adequate means of defining new functions but it also has the disadvantage of having no means of defining variable length fields. FILECODE, a Ferranti language for PEGASUS and SIRIUS, has the distinction of being the first commercial compiler to work in the UK. Like Language H for similar size machines, it is primitive, but it does work and it has good facilities for data description.

The primary characteristics of existing source languages for both business and science, were summarised towards the close of the Conference. They were totally different and ranged from the specialised to the naïve. Generally speaking, business languages tended to be sophisticated in data input and output, although there was still a great deal to be done in abstracting a general data structure for use in such languages, but, at the same time, they tended to be naïve in procedures. The reverse was true of ALGOL and other mathematical languages whose procedures were sophisticated but whose data description was naïve. Business languages were, in general, imprecise in definition and they avoided the use of symbols altogether. They used a great many *ad hoc* devices and usually arrived at a certain efficiency in the object program but not in the translation process. Mathematical languages, on the other hand, usually had extreme precision in statement with a high degree of symbolism, suitable to the mathematically inclined and much abstracting and generalising by designers but no *ad hoc* additions. Their translators, therefore, tended to be considerably smaller.

There are some areas of weakness in both types of language. Business languages are weak in specifying operating procedures, in updating, correcting, loading etc., i.e. in all aspects of actually instructing a computer. This weakness is reflected in the manuals, which causes confusion. ALGOL has helped to clarify this, but it and other mathematical languages in their turn have the serious disadvantage of always assuming the computer to be a serial processor with a large store: they do not allow for a computer with a different structure. The early weakness of not taking into account the operating system of the computer itself is still reflected in the present languages. This bears out how closely tied up is the design of programming languages with the design of machines.

Present Situation

The situation in programming languages today is very like that of computers ten years ago: few are actually working,

there is the general fascination of new ideas, which progress at a faster rate than the projects themselves, and a great many unresolved problems in the construction of languages and of their compilers. None of the commercial languages presented to the Conference had really solved the major problems of the debugging of the object program and of the ability to incorporate readily new functions into the source language. The running time of object programs, in general, is greater than that of machine-coded programs. The cost of this becomes really significant on machines such as ATLAS, IBM 7094, etc., and for programs in constant use. The optimising of these object programs by a skilled machine-code programmer would still seem to be necessary. Moreover, the efficiency of the object program will vary for each type or even configuration of machine, so that it is not easy to determine the optimum efficiency of any one compiler.

A good source language available for use is still only part of the programming problem. There is still the problem of writing the compiler, a problem bound up with the nature of the object program as well as with the design of the hardware. The aim is to shift the hard work of programming from the programmer on to the machine and on to the compiler, with the result that to date some have reached a horrifying size, requiring a tremendous amount of equipment and becoming very difficult to maintain. There is room for development in compiler techniques, and of special, machine-independent languages in which to write compilers. One such, from Manchester University, was described at the Conference, a method of writing compilers for phrase-structure languages, resulting in a general system orientated towards the use of programming expressions; this is at present being tried out using 1,000 words of fixed core store now working in ATLAS.

The news of the errors still found in FORTRAN after several years' use gave rise to a discussion on how much checking is appropriate in a compiler. There is a similarity here to the discussions in the early days on how much checking was appropriate in the machines themselves. It was, and is, a question of how short a mean free period between errors can be tolerated. Early machines incorporated an enormous number of checking facilities with the result that their actual production time was considerably reduced. Later the machines themselves improved so much that the checking circuits were removed. Similarly, present compilers generally incorporate a large number of checks because of present thinking about their construction but later as these checks are shown to be more and more inappropriate they may be removed. There will probably emerge an irreducible minimum of checking for incorporation into compilers and, while this was unknown at present, it was still important to determine it.

Even after the source language and its compiler were established, the troubles were not over, for the problem of maintaining them was real and continuing. Not only does the language, and its compiler, continually need further definition and interpretation, it has to keep up with the new options on equipment and the more complex the new hardware the more difficult the language becomes to maintain. An example was given of the effect on the Honeywell FACT compiler of the advent of magnetic discs (random-access) for the Honeywell 800. The compiler could have been altered to take account of the potentialities of this new equipment but this would be a most difficult thing to do for a complex language such as FACT, with the result that at the moment no one is actually using these discs on the 800 with their proper flexibility and power because it is so difficult to adjust the automatic programming language. The general problem is to permit within a machine-independent language statements which are explicitly machine-dependent: at present only minor variations in environment can be accommodated. Even ALGOL is to a certain degree machine-orientated.

However, the most fundamental of these unresolved problems in source languages and their compilers springs from the desire to make them as general, if not universal, as possible. There seems to be an unavoidable conflict between their generality and some other desirable characteristics. The more general the language strives to be, the larger and more cumbersome it tends to become, and consequently it becomes difficult to learn, even if it is easy to use: contrariwise, attempts to provide languages which are easy to learn as well as to use have usually resulted in a loss of generality. Likewise the more general (and larger) the language the more equipment needed to implement it and then what becomes of automatic programming for the smaller-type computer? If a universal language is eventually accepted *can* it be implemented on the whole range of machines, both large and small?

These distinctions may not always be valid but they appear to be so now and at the Conference they generated much discussion on the generality and commonality of languages. The pressing need for common source languages was accepted, but there was no conviction as yet that there need be only one, or possibly two (to cover separately the mathematical and business fields) and certainly none that these two should be ALGOL and COBOL, neither of which could be implemented in their full generality on a small machine. If, in addition to being unavailable on small machines, general languages were going to become so complex as to be difficult to grasp, their whole end was defeated and it would seem, for both these reasons, that many programming languages will still be needed. Their diversity is probably as necessary at present as is the diversity of equipment; only by actual use will their worth be proved and a choice made. To single out now one common programming language was stigmatised by one speaker as being an entirely superficial view.

This leads on to the vexed question of the international standardisation of universal languages and to the present state of ALGOL and COBOL, its chief subjects. At present ALGOL is incomplete as regards data description and this lack of definition limits the efficiency of the compilers. ALGOL must develop but unfortunately no organisation was set up in 1960 to look after this, a situation which was remedied recently by the setting up of an IFIP working party to take charge of ALGOL maintenance and development. This contrasts with the situation in COBOL where the original committee is still working on the next revision.

From March 1960 to May 1961 was a period of clarification of COBOL after which COBOL 1961 was produced. By December 1961, twelve USA and three UK manufacturers were implementing it on thirty-five different machine models. Since May 1961 a great many moves have been made towards maintaining and extending the language and making it more precise, the main extensions covering optional arithmetic features, sorting (a great step towards commonality for no language has as yet dealt entirely successfully with sorting procedures) and report writing. Other extensions cover Table Handling and Bulk Files with, significantly, some for the benefit of small machine users. In fact there is so much activity here that no COBOL 62 report will be issued, the next being COBOL 63. Within this committee there is a small

group who want all optional features to become required features, a fitting commentary on the difficulties of producing a general language.

Comment was made on the profusion of committees now engaged in considering the standardisation of commercial languages (up to 13 have been counted). ECMA, IFIP and ISO are the three chief international organisations concerned. This is a more than usually difficult subject for standardisation, and two of the committees set up, those within the British Computer Society and the British Standards Institution, are remaining content for the time being to study different aspects of the subject and to be forums for the exchange of opinions, rather than for drawing up any actual standardisation proposals.

The Future

What is the goal of standardisation of programming languages? Is it literally one language for everything *or* one language with permissible varieties, *or* different languages for different fields and so on? The whole subject roused a certain antagonism among members of the Conference. Some felt it better to veer away from the subject altogether and others that the pursuit of standardisation should not go so far as to stultify the development of the languages themselves. There were heavy commercial interests involved in standardisation since the development of a programming language eventually influenced the development of equipment. An instance of this has already been noted where the new Honeywell 1800 had been designed to take FACT, a language designed for their previous machine, Honeywell 800. Will a standard source language lead to a standard machine? And is this a desirable end? Standardisation has its dangers. Certainly the aim should be to avoid at any rate, capricious variations which hinder the exchange of information and an ideal language may come more quickly if a little is given on equipment but anything in the nature of compatibility of machines tends to frighten both manufacturers and users. To do away entirely with the diversity and ingenuity of a variety of machines would be a horrible result.

The progress of any international standardisation of programming languages was bound to be slow. A sense of perspective was induced at the Conference by noting that while there were 1,000 American Standards in existence there were only 100 International Standards altogether. At the moment the ISO/TC 97 Working Party E (with a USA Secretariat), responsible for this work, had had only one meeting, that at Stockholm, 8–10th May 1962, for which a survey of programming languages was begun but by no means completed. There was bound, too, to be an interaction on this work of other international studies, in particular the work on Coded Character Sets.

In looking to the more remote future one sees how very much the introduction of universal programming languages is bound up with machine design. An inappropriately designed computer will render powerless the features of a language it cannot implement, however well-planned they are. COBOL, for example, has extreme generality in the matter of the use of variable length records but many machine designs, both in respect of logic and size, would stultify this feature. It seems that designers of machines and designers of compilers should work together. The importance of the new Burroughs B 5000 computer, designed from the ground up to accept programs written in both ALGOL and COBOL, was emphasised here. The problems of compiler writing should also ease in the next five years: already in the USA programs in the language NELIAC (a version of ALGOL) have been successfully shifted from one computer to another without reprogramming.

While standardisation is clearly on its way, the form this will eventually take is not yet certain. It is also clear that COBOL will become prevalent in Great Britain as it has done in the USA. It has improved greatly since the BCS study of it 18 months ago and, in fact, is being foisted on manufacturers by customers and not the other way round. A "Teach Yourself" COBOL manual has been prepared and the possibility of constructing a teaching machine for a simpler version of COBOL, such as RAPIDWRITE, has been discussed. The inevitability of COBOL will not lighten the difficulties of standardisation, nor of the maintenance of the language (even now the different versions are confusing), nor will it deal with the problem of the commonality of machines. Will these other well established languages, such as FORTRAN and FACT, have to give way eventually before ALGOL and COBOL? One cannot tell. Already an even more general and very large language, JOVIAL, has been produced as an extension to ALGOL. In fact, the whole field was still very uncertain and one of increasing scientific technicality.

The prospect of providing just one universal language seemed very distant. These languages have a high intellectual content, needing informed experts to draw up the specifications and one of the troubles was that, so far, there were experts in either ALGOL or COBOL but not both. There was a suggestion that ALGOL, generally accepted as being a more elegant and precise structure than COBOL, could be adapted for commercial purposes, giving thereby a greater commonality of machines. The breakthrough here may come through the pursuit of problem-orientated languages, languages which were a genuine aid to a systems man in problem documentation. At present languages are merely procedure-orientated, just an aid to programming but not to problem definition. The significance of the two other groups which like the COBOL Committee come under CODASYL, lies here. The first is a Systems Group, concerning itself with tabular languages: one of their studies, TABSOL, is heading for publication now. The second is a Language-Structure Group, studying algebraic languages with a view to determining the relations between the data elements in data processing: its report has appeared in the Communications of the ACM.

As far as this country is concerned, it was suggested that there was far too great an apathy in this subject and that not enough was being done to justify our calling ourselves a home industry. Home products were either not working or not on a par with the American. In spite of the lead originally given in Europe over ALGOL, we were in danger of becoming an American satellite, not only in equipment but also in programming languages, so much so that the situation was neatly summed by one speaker saying that manufacturers should stop selling solutions to problems in incomplete languages with compilers not yet written for computers not yet working! He further looked forward to the day when we could produce a compact home language that would outshine some of these overseas monsters.

Note: Due to the time lapse between the specification and implementation of programming systems, any dating is bound to be somewhat arbitrary. The following, which is only intended as a guide, is accurate to within a year either way. The selection of items is also arbitrary and in the later years refers to those of particular interest in the UK.

APPENDIX

(Reproduced by kind permission of its author, A. d'Agapeyeff, of *Computer Analysts and Programmers Ltd.*, London)

A POTTED HISTORY OF AUTOMATIC PROGRAMMING

–1950	Ideas of Turing, von Neuman and others. Machine coding in use, First Input Routines devised for ENIAC and EDSAC.
1951–55	Early ideas voiced on compilers. Assembly programs in use, particularly on IBM 701, UNIVAC I and Ferranti MARK I. Development of interpretive processes.
1956–57	Early ideas voiced on a "universal" language. First non-computer orientated languages in use—FLOWMATIC (business) and FORTRAN (mathematical). Growing dominance of translator rather than interpretive techniques in the USA.
1958–59	First specification of a "universal" mathematical language (the ACM-GAMM proposal later known as ALGOL). First full languages specified in UK—AUTOCODE (mathematical), CODEL* (business). Specification in USA of COMMERCIAL TRANSLATOR, LISP and IPL-V (symbol manipulation).
1960	First specification of a "universal" commercial language—COBOL. Several other commercial languages specified—e.g. FACT and NEBULA. Revised version of ALGOL and proliferation of ALGOL-like languages.
1961	Revised version of COBOL and proliferation of COBOL-like languages in the UK and USA. First "advanced" level commercial compilers work in the USA—e.g. COMMERCIAL TRANSLATOR and COBOL. First simple commercial compiler works in UK—FILECODE.

* This language was in fact never implemented, being replaced by COBOL.

UNWIN BROTHERS LIMITED, WOKING AND LONDON