1976

FINAL
PROGRAMME

DB/DC
The State of the Art

London
November 8–12 1976

State of the Art Conference
On-Line Data Bases

State of the Art Tutorial
Distributed Data Bases

November
8
Monday

November
9
Tuesday

November
10
Wednesday

November
11
Thursday

November
12
Friday

State of the Art Conference

On-Li
Data

Excelsior Hotel, Heathrow (London

State of the Art Tutorial

Distri
Data

Excelsior Hotel, Heathrow (London

ne

Bases

Airport), November 8-10 1976

State of the Art Conferences

Conference speakers

Conference programme

uted

Bases

Airport), November 11-12 1976

State of the Art Tutorials

Tutorial speakers

Tutorial programme

State of the Art Conferences

Conference speakers

Conference programme

Session 1, Day 1
9.30-10.45

Overview

Why on-line data bases; benefits and pitfalls; relationship between DB and DC software; user interface; problem definition; controlling costs; performance and reliability.

Speaker: E E Tozer, Software Sciences

Session 2, Day 1
11.05-12.20

Software selection for DB/DC systems

DBM packages; TP monitors; interfaces – integrated vs stand-alone; application program interfaces; terminal control; recovery/restart; performance factors; selection methodology.

Speaker: R Rustin, Chase Manhattan Bank (USA)

Session 3 and 4, Day 1
13.40-14.55, 15.15-16.30

Two case studies

Two large users' experience in moving to DB/DC: motivation; system design criteria; software selection methodology; choosing a DBMS and a TP monitor concurrently; products available; ancillary software; interfacing problems; cost/benefit; implementation experience; variations from expectations; operational behaviour of chosen systems; lessons of experience.

Speakers: M Gurr, BOC, and J M Sykes, ICI

Session 1, Day 2
9.30-10.45

Concurrent access

Protection and access control; contention; mutual exclusion; deadlock and deadly embrace; duration and scope of data locking; performance overhead; data base design implications.

Speaker: I Macdonald, CACI

Session 2, Day 2
11.05-12.20

Back-up and recovery

Error detection; fault location and containment; checkpointing; audit trails; message logging; image logging; back-out; delayed updating; recovery and restart.

Speaker: E E Tozer, Software Sciences

Session 3, Day 2
13.40-14.55

Tuning on-line data bases

Access path optimization; data structures; retrieval/update trade-offs; security/recovery overhead; file placement; direct access storage; communications aspects.

Speaker: L J Cohen, Performance Development Corp (USA)

Session 4, Day 2
15.15-16.30

Self-organizing data base systems

Motivation; overall structure; processing components; design and implementation; usage measurement; self-optimization; localized control mechanisms.

Speaker: P Stocker, University of East Anglia

Session 1, Day 3
9.30-10.45

On-line data bases on minicomputers

Software availability; standard access methods; memory constraints; architectural features; communications; back-end processors; distributed data bases; data bases on micros.

Speaker: J Gross, Data Logic

Session 2, Day 3
11.05-12.20

On-line enquiry systems

Types of enquiry language; binding techniques; file structures; execution considerations; admissability of queries; non-procedurality; cross sectional processing; ergonomic aspects.

Speaker: W King-Gillies, CII-Honeywell-Bull (France)

Session 3, Day 3
13.40-14.55

Alternatives to the generalized DBMS

Relational data model; text processing; enquiry systems; data retrieval; ease and efficiency of use; overhead of generalization; small low-cost data bases; distributed intelligence.

Speaker: R W Bemer, Honeywell (USA)

Session 4, Day 3
15.15-16.30

Forum

An open discussion session at which speakers answer questions from the floor and elaborate on points raised during the conference, while delegates contribute their own experience and expertise to the discussion.

State of the Art Tutorials

Tutorial speakers

Tutorial programme

Session 1, Day 1
9.30-10.45

Introduction/Architecture

Growth of interactive processing; emergence of distributed systems; distributed systems versus networks; distributed data bases. Horizontally distributed systems; hierarchically distributed systems; partitioned data bases; replicated data bases.

Session 2, Day 1
11.05-12.20

Advantages and problems

Greater interest at local levels; tailoring to local requirements; flexibility; economies; complex interactions; technical resources; manufacturers' support; security.

Session 3, Day 1
13.40-14.55

Processing philosophy

Centralised vs distributed; degree of interaction; time of applying amendments; amount of inconsistency tolerated; possible solutions for a given application.

Session 4, Day 1
15.15-16.30

Workshop in system solutions

Examination of solutions possible in terms of architecture and processing philosophy for a given exercise.

Session 1, Day 2
9.30-10.45

Support for distributed data bases

Hardware: technology developments, processors, mass storage; software: network software (SNA, DECNET etc), data base software for small machines (TOTAL, IDMS, SIBAS, INFOS etc), interfacing problems.

Session 2, Day 2
11.05-12.20

System design objectives/Data analysis

Selection of appropriate mix of performance, economy, implementability, availability, flexibility. Identifying the functions; where is the data required; where are the functions required.

Session 3, Day 2
13.40-14.55

Selection methodology

Selecting an appropriate architecture and processing philosophy; what kind of system has to be built; where to allocate how many copies of which files; how to retrieve the allocated files; what data structures must be provided.

Session 4, Day 2
15.15-16.30

A case study

A detailed example of a distributed data base system, the key decisions taken in its design, and the methods of organization and implementation.

State of the Art Conferences

Conference speakers

E P Magnuson
Consultant, CACI

Eric Magnuson's wide experience includes work with the US Armed Forces, a major manufacturer, assignments with large users, and now consultancy. He has worked in the USA and in Germany, as well as in the UK, on the design, evaluation, and implementation of on-line data base systems.

Chairman

R Rustin
Manager of Data Base Planning, Chase Manhattan Bank (USA)

Leading US exponent of the data base approach, Randall Rustin's experience in the field includes academic, commercial, and consultancy activities. He has recently led an in-depth study, sponsored by Chase Manhattan, of currently available DB/[illegible] software.

Session 2, Day 1

M Gurr
System Planning Consultant, BOC

Active in computing since 1957, Mike Gurr has been deeply involved in data base technology both in the academic and the commercial environment. Joining BOC in 1972, he now acts as data base consultant within the entire BOC group.

Session 3, Day 1

J M Sykes
Central Management Services, ICI

With nearly twenty years experience in computing, Mike Sykes has played a leading role in ICI's move to on-line data bases. He was a member of the evaluation teams that examined available TP/DBMS software and led to the selection of ICI's current software.

Session 4, Day 1

I Macdonald
Consultant, CACI

Ian Macdonald has wide practical experience in data base technology, having worked in this field for two major manufacturers as well as for CACI, Europe's leading specialists in DB/DC. His most recent assignment was to advise one of the UK armed forces on its commitments in this area.

Session 1, Day 2

E E Tozer
Senior Consultant, Software Sciences

Currently on long-term government assignment relating to several major DB projects, Ed Tozer is vice-chairman of the Data Administrator Group of the Codasyl DDLC. He has extensive experience in the design and implementation of on-line data base systems.

Session 1, Day 1, and Session 2, Day 2

L J Cohen
President, Performance Development Corporation (USA)

Implementor of eight data base systems and author of the report *Data Base Management Systems: A Critical and Comparative Analysis*, Leo Cohen is an internationally recognized authority in the data base and performance measurement fields.

Session 3, Day 2

P Stocker
Professor of Computing Studies, University of East Anglia

Peter Stocker's experience with computers began in 1953, since when he has worked in the aircraft industry, in government establishments, and for a major computer manufacturer. Since moving to the University of East Anglia, his researches have centred on data base technology.

Session 4, Day 2

J M Gross
Manager, Database and TP Group, Data Logic

With extensive DP/TP experience with two major manufacturers and two consultancy firms, Jeremy Gross's work on minicomputer-based DBMSs dates from 1973. He has recently completed a study of centralized and local dedicated data bases for order-processing applications.

Session 1, Day 3

W King-Gillies
CII-Honeywell-Bull (France)

After eight years' experience in user installations, Will King-Gillies joined De La Rue Bull in 1968 to specialize in integrated data base design. His experience within this company and its successors makes him one of Europe's leading exponents of the data base approach.

Session 2, Day 3

R W Bemer
Staff Consultant, Honeywell (USA)

Bob Bemer's distinguished career in computing includes experience within IBM, Univac, Bull, and General Electric, as well as Honeywell. In recent years he has headed the development within Honeywell of an interactive enquiry system based on the relational model.

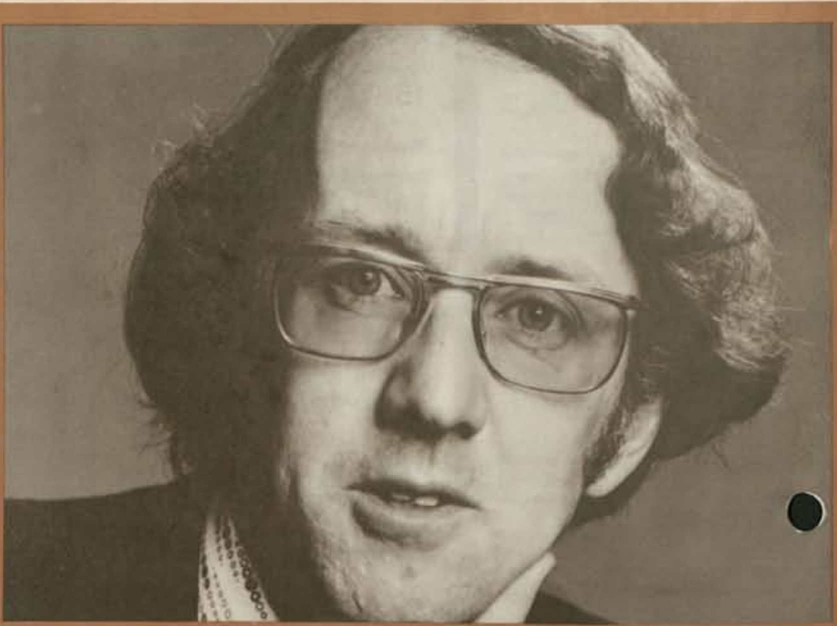
Session 3, Day 3

Forum

Session 4, Day 3

State of the Art Tutorials

Tutorial speakers



I Palmer
Manager, Data Base and TP Group, CACI

Ian Palmer, one of the world's leading experts on data base technology, heads the Data Base and TP Group of CACI, Europe's foremost specialists in the data base field. On behalf of CACI's overseas affiliates, Ian Palmer has provided DB consultancy services throughout the world, but especially in the USA. He has been active on the Codasyl Data Description Language Committee since 1972 and is currently Vice Chairman. He is equally active on the Codasyl/BCS Data Base Administration Working Group. He is perhaps best known for his book *Data Base Management*, a third edition of which has recently been released under a new title, *Data Base Systems – A Practical Reference.*

Ian Palmer has been actively involved in large data processing systems for many years using a variety of hardware and software. Recent data base assignments have included a large management information, financial control, and operations research system, and the design of a data base to support financial accounting and on-line inventory control. Possibly his most challenging data base project was a detailed study of the policies that should be adopted with regard to the use of data base and DBMS within the multiple installations of one of the largest of the nationalized industries in Britain.

R A Davenport
Consultant, CACI

Bob Davenport is a lecturer in Systems Analysis at the London School of Economics and a CACI consultant. He has been concerned with the design and implementation of transaction processing systems for several years working for an English consultancy company.

E P Magnuson
Consultant, CACI

Eric Magnuson's wide experience includes work with the US Armed Forces, a major manufacturer, assignments with large users, and now consultancy. He has worked in the USA and in Germany, as well as in the UK, on the design, evaluation and implementation of on-line data base systems.

The Infotech State of the Art

Conferences

The Infotech State of the Art Conferences are designed to present, through a selection of authoritative sources, the state of the art in subjects of current concern. They are designed to meet the requirements of technically orientated personnel who are involved in the subject area and wish to ensure that their knowledge of the subject and its practical application are at the highest possible level. The objective of each Conference is briefly to review the theory of a subject to the level necessary to understand its practical application and then to expose the lessons of hard experience gained by leading practitioners. Ample time is allowed for discussion so that delegates can state their own concerns and obtain feedback from the speakers. Within the Conference framework, delegates have the opportunity for discussion with their peers from major organisations throughout Europe, all of whom are involved in the same subject. Each Conference provides a sounding-board for delegates' own knowledge and experience and a source of new insights into problem areas.

State of the Art Conferences

The Infotech State of the Art

Tutorials

The State of the Art Tutorials are a major development within the Infotech State of the Art Project, introducing a new format complementary to the State of the Art Conference. The tutorials are designed to provide participants with the benefit of exposure, over a concentrated two-day period, to a single world-renowned authority in a specialized area of computing. The Tutorial framework allows the speaker to devote an extended period to a comprehensive unified view of both the current state of the art and future developments in his field. At the same time, interspersed discussion periods provide participants with extensive opportunity to gain immediate, authoritative feedback on those topics of particular concern to themselves. Top names figuring in the Tutorials have included Gordon Bell, Gene Amdahl, Terry Baker, James Emery, Algirdas Avizienis, Chinnoor Ramamoorthy and Peter Denning, as well as Ian Palmer himself.

Gene Amdahl

Algirdas Avizienis

Terry Baker

Gordon Bell

Peter Denning

James Emery

Ian Palmer

Chinnoor Ramamoorthy

State of the Art Tutorials

November
8
Monday

November
9
Tuesday

November
10
Wednesday

Registration:

On-Line Data Bases

State of the Art Conference
November 8–10 1976
Excelsior Hotel, Heathrow (London Airport)

Company/organization

Full postal address

Booking placed by (name/position/telephone)

Delegate's name/position	Fee £205

Cheques payable to Infotech International Ltd
Infotech reserve the right to change the programme without notice
Send to the Registrar, Infotech International, Nicholson House, High Street, Maidenhead, Berkshire, England.

Net fees	
+8% VAT	
Total fees due	

November
11
Thursday

November
12
Friday

Registration:

Distributed Data Bases

State of the Art Tutorial November 11–12 1976
Excelsior Hotel, Heathrow (London Airport)

Company/organization

Full postal address

Booking placed by (name/position/telephone)

Delegate's name/position	Fee £145

Cheques payable to Infotech International Ltd
Infotech reserve the right to change the programme without notice
Send to the Registrar, Infotech International, Nicholson House, High Street, Maidenhead, Berkshire, England.

Net fees	
+8% VAT	
Total fees due	

IBM

IBM ÖSTERREICH INTERNATIONALE BÜROMASCHINEN GESELLSCHAFT MBH A-1020 WIEN, OBERE DONAUSTRASSE 95

AAA PHA 0050 NOV 01 76 ZZZ
ZZZ PLEASE REFILE TO : IBM LABORATORY VIENNA
ATTENTION: PROF. H. ZEMANEK
IBM-HAUS, A-1020 VIENNA/AUSTRIA
TELEX 7 4481 IBMVIE A

APOLOGIES. TWO REFERENCES ARE: R. A. MC LAUGHLIN, EQUITY FUNDING: EVERYONE IS POINTING AT THE COMPUTER, DATAMATION 19 NO. 6, 1973 JULY, 88-91; ALSO EQUITY FUNDING: THIS SWINDLE COULD HAPPEN HERE, THE ECONOMIST, 247:10, 1973 APRIL 14. RE LEGAL ASPECTS - HONEYWELL COMPUTER JOURNAL PRODUCED ORIGINAL PUBLICATION. ALSO AVAILABLE FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VIRGINIA 22151, AS -QUOTE- COSATI 73-01 -UNQUOTE- (ONLY TITLE). CLAMONS SUGGESTS FOR STARTING LAST PARAGRAPH FIFTH PAGE: -QUOTE- ICH MOCHTE IN ALLER KLARHEIT VORAUSSCHICKEN DASS ICH COMPUTERS GERNE HABE - UNQUOTE-.
THANK YOU.

R. W. BEMER MD C61 4:10 PM MST HIS/PHA

Enclosed you will find the translation of your paper "Computers and our Society" as it shall appear in "Elektronische Rechenanlagen". You can check the translation.

Please have a special look at the literature. Unfortunately, your 70 ACM Volume has temporarily disappeared - probably put on the wrong spot after our move from Parkring. If you could improve any of the quotations, please do so.

Do you have a better specification of the US Government Report on "Legal Aspects"?

On the Equity Funding scandal we found no reference, but European readers may not be that familiar with it - in fact, I am not. Could you find any article on it that could be easily found by the European computer specialist - say in the Communications of the ACM or in a similar journal?

I hope you are satisfied with our work!

Sincerely yours,

Heinz

Prof. H. Zemanek
IBM Fellow

RBemer

REPORT

18th Annual One-Day Symposium

June 6, 1976

Americana Hotel, New York City

The Next 30 Years In Computing

by
Fred Gruenberger

Engineering Center
School of Engineering and Computer Science
California State University, Northridge

List of attendees for the June 6, 1976 symposium

Paul Armer	Center for Advanced Studies in Behavioral Science
• Robert Bemer	Honeywell Information Systems
Barry Gordon	IBM
– Irwin Greenwald	Xerox Data Systems
• Fred Gruenberger	California State University, Northridge
• Daniel McCracken	Author
• Clarence Poland	IBM
Richard Tanaka	California Computer Products
Robert White	Informatics
• Joseph Weizenbaum	Laboratory for Computer Science

Every year since 1958, a one-day invitational discussion session has been held to discuss current topics in computing. Paul Armer and Fred Gruenberger have co-hosted these affairs, which were sponsored until 1968 by the Rand Corporation, and since then by California State University, Northridge.

The 1976 session will probably be the last of these symposia, on the grounds that they have outlived their usefulness. Each year, the attempt was made to gather a homogeneous group of experts, and this tended toward the senior citizens of the field. There is some evidence that the results were fruitful, but that the impact, if any, has attenuated over years. Like any good vaudeville act, it seems wise to get off the stage while the audience might still be enjoying the show.

It seemed like a good idea, therefore, to change the theme for 1976; to avoid the burning questions of the day and consider what are likely to be controversial topics of 2006. In the announcements of the meeting, the thought was offered that it was not likely that in 2006 the computing fraternity would still be debating the relative merits of Fortran vs. PL/1. On the other hand, it would still be true that one shouldn't calculate constants inside the loop. The questions to be discussed, then, are the things that might endure. Of course, as you might expect, several of you disagreed with me on the things that I listed that we wouldn't be debating 30 years from now.

GORDON: I didn't know that we were debating Fortan vs. PL/I now.

GRUENBERGER: You can get any ten computer people to spend hours hotly debating that one.

BEMER: We should clear up our terms of reference. The physical evolution in computing is visible and tremendous, and we can reasonably expect more of it in the next 30 years. But look at the people in computing. There I don't see any evolution, or much improvement. People will learn one language, like COBOL, and they are willing to live with that for the rest of their life. Tony Pizzarello gave some talks at our place on structured programming and P-notation languages and things like that, and found that only about 10% of the people could be retrained at all. They don't seem to care. They will, indeed, argue the merits of COBOL, PL/I and Fortran for years.

POLAND: But with any luck, those people will die off soon enough and a new breed will emerge.

WHITE: Except that the old ones propagate. Fortunately, COBOL is not hereditary.

GREENWALD: We may be ignoring the laws of economics. The old dogs may die, but outside of the areas of hardware and systems programming, the investments that have been made in programs prevent people from being innovative; you have to be compatible. The present programs control very large data bases, and you can't come in and change things fast. You may evolve, but it will be a slow process.

WHITE: You can change rapidly once in a while; I've seen it happen. I agree that it's not common. The problem is tied in with the statement that people who don't pay attention to history are condemned to repeat it. We don't have a written history, much less any history that is being taught. Most of our history is in the heads of us old dogs, and the new people are making all the same mistakes all over again.

WEIZENBAUM: The trouble is even more fundamental. In computing, there is a sinusoidal phenomenon with respect to all sorts of developments. Some topic is controversial for a while; then it disappears, only to reappear later in a new guise. Structured programming is one example. Perhaps a better one is interpretation vs. compilation. At one time it was debated. Then (around 1959) it was regarded as settled, in favor of compilation. Later on, when conversational computing came in, interpretation was again useful, and the controversy starts again. Today, with the microprocessors coming in, we see all the old issues being explored as though they were new. We see people right now going through the stages from absolute octal to symbolic coding, to assemblers, and on up. You can all remember when the writing of a compiler was held to take many man-years of work, and soon after that it became an exercise for a college junior--and the microprocessor people are busy repeating that history, too.

Most of this knowledge is stored in the form of finished products (which is not documented), or in lore, which is in people's heads. In my own early days, I was essentially a journeyman, moving from one interesting job to the next, and carrying the essential information with me.

Think back to the efforts we exerted at one time toward getting useful work done with small memories. When larger memories came along, we all said we'd never have to face that problem again--but they're doing it.

GRUENBERGER: I'm reminded of a long debate we used to have at RAND. The topic was: if everything got wiped out in a nuclear war, how long would it take to rebuild civilization as we know it? One side, headed by John Williams, said 50 years or so. The other side said 5 years or so, on the grounds that we would know exactly what had to be built. For example, we wouldn't have to wait for nylon to be invented; we would know of its uses and advantages (and its makeup) and be able to head for nylon production directly. We would aim directly toward computers, similarly, and 4th generation machines right away. No one would propose going slowly and painfully through the vacuum tube stages. In other words, the plan is all there and everyone knows exactly what to do; moreover, they know where the eventual profits lie. Now, won't the same thing happen here? The people with the tiny machines are facing ancient problems, but at least there are people around who can guide them through the early stages fast, knowing where to head. Won't that help them get there much faster?

WHITE: Perhpas, but it doesn't seem to be working.

GREENWALD: They face another ancient problem you may have overlooked. They can get people off the street to do the work very cheaply. Hence the ones they're getting to do the job aren't the good ones.

POLAND: No, they're very good; they're just totally inexperienced. There's a big difference.

GREENWALD: So they have to learn the same way we did?

GORDON: They're just not interested in learning from other people's experience. Even if they were interested, they can't, because it isn't documented well enough.

POLAND: That's the key. Postulate any group of competent and bright people, whom you want to move into a new area. How do you get them started? There are few, if any, books, periodicals, white papers, and technical reports, on the stuff you want them to learn. You have to put them in direct contact with someone who already knows it.

GORDON: It won't work, as we've seen in IBM over and over. The new group calmly assumes that the experienced people don't know what they're talking about and they invoke their right to start from scratch, which they then do, and we see the same mistakes made all over again.

GREENWALD: But even if they should ask for help from experienced people, what they get is "How I solved that job," rather than how they should tackle the new job.

POLAND: Worse than that; they don't hear "How I tried and failed to solve that problem."

GRUENBERGER: Some of this is simply fun to watch. The 701 had a 5-bit op-code, and 32 instructions, and we quickly learned what chaos can result from not having some redundancy built in at that point. This glitch was repeated, to my knowledge on only one other big machine (the GE 225). But the micro machines are now busy on the same path. They have a 6-bit op-code and they cram 64 instructions into the logic, and of course reap the same reward. But my earlier point is still valid, I think. Even though they're repeating history, they will recover much faster than we did, because someone can point out immediately where the trouble lies. The fact that the knowledge is there must operate to speed up the evolutionary process.

WEIZENBAUM: I think we are being a little to harsh with the old-timers. The programming profession is, and always has been, in performance mode. The thing to do is to get something working. It turns out that it is usually easy to get something working reasonably well, by which I mean that you are not worried about the maintainability of what you've produced, or about documentation. But you can deliver the product to the customer and claim that it has the required input/output behavior. It isn't that there aren't books; there are plenty of them. It's that books, and deeper knowledge, are simply unnecessary in order to get just past the threshold of mere performance. It's an expression of greed for quick profits, and a failure to pay attention to longer term goals on the part of everyone. Things like quality and maintainability--these are also sinusoidal in nature, because those things have been preached from the very beginning.

GREENWALD: I'd like to call attention to the latest issue of Software Engineering, in which someone has analyzed 120 commerical PL/I programs for such things as clarity (they were terrible), comments (there were none), indentation (hardly ever used), the use of IFTHENELSE (it was IFTHENGOTO).

WEIZENBAUM: There is a similar study done at General Motors, with, unfortunately, the same results.

GREENWALD: It all supports your statement: the name of the game is "get it working."

WHITE: That was our attitude at Informatics five or six years ago. At that time, we could produce a new feature for Mark IV in from four to six months from the time of inception. The motto then was get it working and get it out. Of course, that caught up with us. Today, to do much the same thing, but to do it right, with internal and external documentation, proper design and total controls--a maintainable product--takes 18 months minimum. It takes at least three times as long between doing it and doing it right and I doubt that there is much difference in the level of the people involved. It is a matter of having the right controls and doing the right thing.

GORDON: You still have people who should know better who are putting statements into print like "First make it work, then make it pretty."

POLAND: I think we're talking about the wrong topic. If we're serious about talking about the art of computing in the year 2000, I think we're talking about art as equivalent to circuit design. I think the bulk of computing will come from people who do not understand "computer science" and never had a course in it. Call them end users if you wish. They will not think in terms of loops or data bases. They shouldn't have to. But that's where the business is going to come from, in terms of dollars or in terms of where computing is.

GORDON: That's where the business is now. I must confess that I don't know what you mean by "computing." Over the years, most of you have become elder statesmen, while I've just become a senior citizen. Compare, if you will, an old-time pilot of an airplane, who really flew the plane, with the business-suited man of today who spends an hour with a checklist and then "flies" the plane by pushing buttons to invoke servomechanisms and autopilots. The whole thing is mechanized and routine. When you guys talk about computing, you seem to be talking about the goggle and white scarf guys. I feel there's a big difference between computing in academia and data processing out in the real world. In the latter world, people worry about shifting from one disk drive to a different one, and how they will conserve their data. They are concerned about things like their investments in applications code.

WEIZENBAUM: Your analogy with airplane pilots is not too good. To be sure, the modern pilot does much of his job with gadgets. But he is trained--right down to the aerodynamics--to take over when the gadgets break down, and then he needs skills and knowledge that the white scarf guy never dreamed of.

POLAND: The end user at a boob tube is pushing payroll buttons, not even COBOL buttons.

WEIZENBAUM: It might be quite possible to train someone in six months to fly a 747 and he could indeed fly one for years without getting anyone in trouble, provided that everything works. But in our business, that's not how we train our pilots. I think what is going to bug us is what I call incomprehensible programs. We are getting very large computing systems, and larger ones are coming, and more of them. They are becoming increasingly more incomprehensible, in the sense that no one person understands the whole program, no team of people who understand the whole program and, indeed, no group of people who can be identified as being responsible for the system over a long period of time. The programs are essentially anonymous.

There is a threshold of complexity that we have already begun to cross, where the factor of incomprehensibility comes into play, and that's going to cause us a lot of problems. We are creating larger systems by adding patches to existing working programs, by tying programs together, by adding new data bases together, and by networking. The parts of these systems were working and installed, and probably irreversibly installed, in the sense that we couldn't go back to the former methods even if we wanted to. And for such systems, no one understands them.

GORDON: It's the Sorcerer's apprentice.

BEMER: As I keep telling my management, we have to do something before our key people die or retire.

GRUENBERGER: Before we get into the problems of large systems, let me go back one step. The people who use Altairs and similar machines have available a BASIC compiler, which occupies perhaps 65K bytes. Don't we have a difference between now and 12 years ago in that we can say "It's fine and it works, but it isn't pretty" and we can tell them exactly how to make it pretty? Can't we build on that difference? I find it difficult to accept the gloom I hear here, which seems to say that we're going to go on being dumb forever.

GREENWALD: Barry started to define our terms, and I'd like to go on with that. There's a big difference between computing, in the sense of scientific computation, and the data processing field, and the data processing user. The statement that "The way to learn computing is to compute" applies only to the first of those three. I think we can offer some suggestions to the people in the data processing field, which interacts with the needs and goals of the data processing user.

POLAND: You're right on. Consider the day when we might have electronic transmission of the mail. I'll still need a secretary, but she will need to know less about the technology she interfaces with than she does today about the postal system.

WHITE: She will know about line faults and dropped bits and things like that.

POLAND: Perhaps. All she will really know is that letters to me appear at her terminal, and that she can send letters through it.

WHITE: She won't know everything, but she will have some knowledge of new things that she doesn't have now.

POLAND: But that knowledge will be very limited. It will be comparable to the knowledge needed today to determine that the one-time ribbon has run out, or that the wrong type ball is on the machine, or that the space bar doesn't work. She will be able to deduce that there is a computer connected to the terminal. But after that, she quickly runs out of gas, by our standards. Yet it is remarkable how much can be done with such limited knowledge.

GRUENBERGER: But isn't that one of our goals, to enable people to utilize computers without being computer experts?

POLAND: Yes, but the distinction between the use and the technology that fosters that use is frequently overlooked, and leads to confusion in our thinking.

GORDON: One of the troubles is the dependency involved. When the system goes down, and people still have to get things done, perhaps by manual procedures, they are frequently completely helpless because they have learned to depend on the system for even the simplest things.

BEMER: My friend Pizzarello couldn't get a hotel reservation completed once because the system had his name garbled. He tried to fight that battle, but lost, and finally asked "Well, do you have a room available?" and was told "Oh, sure."

GRUENBERGER: Two of you took issue with the dictum "The way to learn computing is to compute." What, then, is the other way to learn computing?

GREENWALD: If you'll change the wording to "data processing," then I'll buy it.

WEIZENBAUM: It is true, I believe, that we (computer people) want to deliver systems to people such that they can make intelligent use of them without having to know what is going on behind the scenes technically, just as we enable people to make fairly complicated long distance telephone calls without knowing anything about message switching. So we can agree, I trust, that we want to spread the use of computers without requiring masses of people to take courses in circuit design and compiler construction. Then there's the matter of dependency that Barry brought out. But now look at another aspect.

My town, Concord, Massachusetts, decided to install a computer system to determine real estate taxes, taking into account many more "variables" than could previously be used, and apply sophisticated techniques like linear programming. I think it's a terrible mistake, inasmuch as events in this system (unlike process control) do not take place every few milliseconds, and hand methods would do the job nicely. But the pitch of the firm that sold the system to my town is that the method of determining the tax will be so much better, since it will invoke magic like factor analysis, so that the causative factors can be determined. When this system becomes operational, someone will ask how his tax was determined, and the tax assessor will have to say "I don't know." This is another kind of dependency, much different from the kind that requires you to know how a compiler works. The user will not know what theory is being applied, and will certainly not know the algorithm that has been implemented. It seems to me that the town will either have to hire someone who knows and understands the algorithms and can explain them to the citizens, or it will have to revert to the old hand methods. This strikes me as typical of the thing we're rapidly getting into. The abdication of responsibility is not a consequence of the technical incomprehensibility of the system, but is a consequence of the substantive incomprehensibility of the system. The real point is that there is no theory in such things. Someone made a system that worked, and then other things were patched on to it. It was generated in performance mode, and then the general manager asked "Can we also do that?" and the system grew. Even with the best will in the world, it is no longer possible to find out the basis on which decisions are being made.

POLAND: I agree with you, but I think the phenomenon is a transient. The reason is that today there is a belief that whatever you want done by computer, the first thing you do is get yourself a systems analyst and a systems programmer. Then someone examines the job to be done (who doesn't know beans about that job--he's a non-tax-assessor) and does a systems analysis. He is followed by a programmer, who programs something similiar to what the analyst describes. In situations where the person with the job, who is an expert in something, directs the computer without the priesthood intervening, then one tends to get right answers. The answers also tend to be meaningful. There will usually be inefficiency in terms of computing capability: CPU time, disk space, and other things that don't cost very much.

GORDON: Yes, that's the shibboleth I'd like to see questioned very strongly. At one time computers were a scarce resource and were very expensive. Perhaps it made sense then to tailor the problem solution to the available equipment, but it doesn't make sense today.

WHITE: It's the only thing that can be measured. You can count usage factors, but you can't measure utility and quality.

GREENWALD: Let's not criticize the priesthood too much on this issue. The reason that microsecond chasing still goes on is benchmarks. Every vendor has to do them against other vendors, and I don't see that changing.

POLAND: And you don't see benchmarks set up to time the installation and measure the maintainability of a new job.

GREENWALD: In competitive situations, we learned that the company that gets there first will probably win, because that company gears the benchmarks to its system. I agree that 30 years from now we won't worry about calculating constants inside the loops; we won't be able to afford to and still do the things we've been talking about. On the other hand, if there is still competition, we'll still be counting machine cycles.

WHITE: It is also unlikely that there will be a change in the attitude of the people who run the systems that the purpose of the man at the far end is to feed information to the computer, rather than having the computer furnish useful information to him.

GRUENBERGER: Did you say, Clarence, that one day the property owner will be able to come to the tax assessor and find out the algorithm that was used to calculate his tax?

POLAND: Yes; that's a desirable objective. They used to be able to, and they should be able to with computers.

GRUENBERGER: You're dead wrong. They'll be able to hide behind that computerized system, and they will.

WHITE: That's a characteristic of bureaucracy, not of computing.

GRUENBERGER: Sure it is, but the computer makes it so much easier. We see it already. Department stores, banks, motor vehicle departments--they have all learned very well that no one argues when they calmly say "The computer says so." Not only won't they give you the algorithm, they won't be able to, because after two revisions of the program, no one will know what the algorithm really is any more, as Joe pointed out.

GORDON; What's more, the new system was sold on just that basis. The salesman told them that they could implement what they did before, and also lots of new things. Again, few people ever then ask "Should we do these things?" They are done simply because they can be done.

WEIZENBAUM: The firm sold the town a system that included all sorts of sophisticated things like factor analysis. The town wasn't even talking to a programmer, trying to explain how they assess taxes. But there is also a political context to be considered, with all the federal rules that towns must observe. The tax assessor would like very much to be able to escape the responsibility of making very tough decisions that are bound to make one segment of the community or another unhappy. The salesman from the computer firm offered him a way to avoid such responsibilities. The pitch was that the hard decisions would be made automatically, logically, scientifically and--best of all--"by computer." In buying it, the assessor has said, in effect, "and no one will be able to argue with me."

GRUENBERGER: Think of the 17 years or so of experience that has built up with banks using computers. They were one of the first groups to adopt widespread use. After all these years, if your bank slips $27 out of your account and you complain, you'll hear immediately "the computer did it." You can ask for the algorithm, but it will be a fight. All the way up to the level of the manager of a branch you'll only get the same idiotic reply. You may eventually get your $27 back, if you squawk loud enough, and then you'll find that the computer did that. You are not supposed to argue with these pronouncements. We're here already.

WEIZENBAUM: Some years ago the president of Allegheny Airlines ran into this when his reservation got screwed up, and when a clerk gave him that line "The computer did it," he said "I'm the president of Allegheny--and we don't have a computer."

GREENWALD: I agree that the phenomenon is not a transient. There will be massive troubles, and public outcries, and many laws passed, but the situation Joe and Fred describe will be here permanently.

GORDON: It will be "temporary" like the income tax.

GREENWALD: There will be licensing exams, and they'll give me a license thro ugh some grandfather clause, but the situation won't be cleared up.

WHITE: I can't agree with Fred as far as banks are concerned. There is a logical procedure, even if the teller or the manager don't know it, and it can get straightened out.

GRUENBERGER: Perhaps it can, but I have a peachy example of where it wasn't, after months of haggling. In the case I can cite, I wound up some $25 ahead, and I figured that I had earned it, at the rate of about $3 an hour of my time.

WEIZENBAUM: It goes back to that survey we cited from Software Engineering. That's the place to begin the attack. With all modesty, I think the place to begin is in the universities. We need some standard of quality and behavior, and a propaganda campaign to persuade end users not to try to get away on the cheap. We have to tell them that they may be able to make a programmer out of a high school graduate, but it's not the way to go.

GORDON: People will have to be burned in order to learn. I remember a discourse at a SHARE meeting, when we were challenged with "Why don't you guys deliver reliable systems?" and I replied "Because you guys don't want reliable systems." They hover over the systems we build with a stop watch, and, since we're not stupid, we deliver what they are willing to buy. When they really want reliability, and specify it, and get rid of the stop watches, we'll start building it. When they get sufficiently burned and understand about maintainability, comprehensibility, and reliability, there will be little improvement.

GREENWALD: There are some things you can measure besides CPU time. You can measure the time it takes to install a system, for example, but how do you measure quality? Can you measure maintainability? If you talk to ten different people, they'll give you ten different ways.

WHITE: That's the vendor's problem.

GREENWALD: Some of these things could be measured by other people's experience. Also, one could measure how long it takes to add a new feature that is compatible with the overall logic.

WEIZENBAUM: When you say you can't measure something, do you mean you can't measure in principle? Most acceptance tests today are based on input/output behavior; namely, does it produce the required output from given input? Perhaps we should require that the producer of the system be able to tell us how the system produced a given output.

POLAND: We're back inside the art again. Which "system" are you talking about? And is it a system, or an application? Are you referring to an inventory system or an operating system? A data base system, or a computing system? These "systems" are all different.

WEIZENBAUM: Let's say that you are the user of an integrated inventory control/purchase order system.

POLAND: No sir. The end-user is responsible for the inventory; he uses one of those funny systems.

WEIZENBAUM: OK. He goes to a vendor of computer systems and states his responsibility and asks for computer assistance. The vendor comes back with a complete package and promises that it will do everything. There may be a 30-day trial period of the system, and it turns out that the input/output behavior is correct. Now, the user can ask "If there's a change in the property tax laws, can I change your system to include that?" The vendor says "Sure." But now you have something you can time (with a calendar). And if it takes too long to do it, then you know.

WHITE: I don't know of any techniques yet in programming that allow you to get a very fast program and at the same time a very modular one. Your man who wants the inventory system is going to buy the fast one, when it's the modular one he needs.

WEIZENBAUM: Then he should give up the speed.

WHITE: Yes, he should, but he won't--not today.

POLAND: But there is a group of people who do. I know of an inventory manager who bought a subscription service for his work, and got what he asked for, together with a four page manual. The writer of the system arranged for him to have data base space, and also to dump his files periodically for protection (unknown to the user). Beyond that, he was in business. His manual told him the procedure that he could use, with no mention of loops or sequencing. He, the user, put in all the algorithms, and entered the data. Now, the system is inefficient; it takes much more CPU time than comparable systems that are not as well produced. It takes no application programmers. It requires no application programmers. It requires no system analysts, nor anyone whose function is not inventory control. Backing up the system are clever computer types, but they do nothing special for this user.

WHITE: Those computing center guys must be really dedicated to service.

WEIZENBAUM: We all know of the familiar trade-offs of time and space. It sounds here as though you have a trade-off involving comprehensibility.

BEMER: I recently got the go-ahead on a job at 2:00 p.m. The next morning there was a meeting to discuss what was going to be done, and it was already completed. It was inefficient, computer-wise, but the computer cost was truly trivial compared to the savings in elapsed time.

POLAND: In my inventory application, the user was invoking an IMS data base, MVS, and a programming language, all of which he knew nothing about.

GRUENBERGER: And you're saying that that's what they should have sold the Concord tax assessor. What are the chances of that happening? If the situation is competitive, it seems to me that it will always lose out to the vendor who offers a poorer system which contains lots of magic but very little common sense.

GREENWALD: Unfortunately, the guy who runs the warehouse is not the one who will make the decision; the decision will be made by his company's data processing people. In the manufacturing industry, at least, every manager is the captive of the corporate data processing department. They may not use the services, but they can't use anything else. That's the thing that is giving data processing a black eye. It is typical that an arrogant programmer says to a potential user, "I know what you want," rather than taking the trouble to find out what he really needs. You get a tax assessment program designed by a system analyst rather than by a tax accountant. The arrogance involved is common to all scientists, not just programmers; in fact, it's common to all people just getting out of college. It's killing us, if only in terms of our public image. And I see little chance that it will improve much in the foreseeable future. My hope is that the situation Clarence was talking about will come more and more as we begin to distribute.

BEMER: I sometimes have the feeling that programmers are blackmailers. They do dirty tricks (mixing up decks; moving things in the operating system) to make themselves indispensible.

GRUENBERGER: If there have been two continuing themes over the 18 years of these sessions, they are: (1) the universities are doing a lousy job and turn out a poor product, and (2) our industry has done a poor PR job. This session seems to be conforming to type. You're telling me that we have to sell a message to someone, and clearly we're not doing it. So what's to be done?

ARMER: It's governed by sheer gross dollars.

WHITE: And the improvements come so slowly.

BEMER: They always come slowly. We should try to discern the trends and then try to accelerate them.

WHITE: We've been selling that (well-designed programs) since 1968, and the rate of selling hasn't changed since then.

WEIZENBAUM: Two things have changed dramatically in the 18 years of these sessions: (1) hardware reliability is taken for granted, and (2) computers are no longer a scarce resource, and no longer expensive. That last fact may come as news to people who are spending a million dollars on an installation, but let's hope they are spending it appropriately and getting far more for their dollar than 18 years ago. At least we in the industry recognize that whatever scarce resources end-users have to wrestle with, it is no longer raw computing power. That fact has not taken effect in the outside world. The users are still sub-optimizing (that is, optimizing at too low a level), and we should work on that, but if there is one single message we should produce, it is that computer time is no longer crucial; it can be traded off against quality and reliability.

GRUENBERGER: But that message is already widespread among students, and their interpretation of it is "computing is so cheap, we can afford to do it sloppy." They believe devoutly that (a) computing power is a free good and (b) there is no point to using intelligence--we can overcome stupidity with sheer speed. There is nothing to be done about (a), because it is obviously true at most universities. The tragedy is that (b) is potently false, and they proceed to prove it, over and over, but still believe it. I haven't found a way to dispell that myth.

WEIZENBAUM: The first time I heard that sentiment (that the machine is so fast that I don't have to be bright) was from a programmer on the RAYDAC, around 1953. We should be careful, when we stress the low cost of computing today, that we don't re-sell that message. The real message is now we can afford to be as elegant as we wish, even if it costs you machine time.

GORDON: But even that is the wrong message. It is cheaper, overall, to do it right, right away. Compilers take minutes; loops take nanoseconds. If it never was true, it is still not true: CPU speed is never a substitute for good programming.

GREENWALD: Not to mention that data base recoveries take days.

WHITE: We have engineered Mark IV so it will not accept a piece of bad data, but every day someone asks us for a switch to shut off that feature so the program will run faster.

GREENWALD: We were forced to put just that switch in our operating system.

WHITE: There is one situation where it might make sense. That's when the customer has the biggest system he can afford, and his daily 24-hour work load takes 25 hours to run.

GREENWALD: A lot of this is the fault of the users, but a lot can be traced to the competition among vendors. There is no cure for that; someone will always be there to sell what people think they want. And if he's successful, the other vendors have to follow suit or come up with a better idea.

BEMER: The hope is that IBM will see the handwriting on the wall...

GORDON: Nonsense. The handwriting on the wall is in hexadecimal, which is the worst crime ever perpetrated on the users. Talk about kowtowing to the machine! A decimal chip costs no more than a hex chip. It's an outrage to the users, and as long as it exists, we have a nerve talking about doing anything to help them.

GRUENBERGER: How do they do it in other industries? It's certainly true that there is always someone around to sell you junk, but in other industries you can buy quality if you want to.

GORDON: You don't buy a car to drive it at 90 miles per hour all day. You expect it to be idle most of the time. Yet in our industry it's a crime to have the WAIT light come on. Where did that get started?

GRUENBERGER: It's a matter of scale. 747's don't sit idle, either, but Piper Cubs do. Similarly, big 370's are seldom idle, but IBM 5100's will be idle most of the time. It's a matter of how much you have to invest just to get the thing there at all.

WEIZENBAUM: And even if you don't drive the car 90 MPH all day, they make it to do just that--and you pay for it.

GREENWALD: And the automobile is also a personal status symbol to a lot of people.

POLAND: Today's personal status symbol is the hand calculator. The guy with the SR-50 is way above the guy with a little Casio. I have a plug-in-the-wall machine on my desk, which has near-zero status, but it's above the guy who has a mechanical Friden.

GRUENBERGER: Which I have on my desk.

GORDON: At one time we pictured the computer as being memory-oriented. Today, if you want to compute, you use a pocket calculator. Today's terminals have the power of a 709. The problem today is the handling of large masses of data. Thirty years from now, our industry will be characterized by memory banks, with terminals providing all the arithmetic you need.

GREENWALD: One big problem today is optimizing the I/O bandwidth so that there is something to do with the CPU cycles.

GRUENBERGER: About 35 years ago, the companies that make eyeglass frames pooled a million dollars to mount an advertising campaign to convince us to buy glasses with the hinges at the top, instead of in the middle, where Ben Franklin very sensibly designed them. That campaign lasted about two years, after which it wasn't needed, as I see that all of us have such glasses. No doubt they're about ready to convert us all back again. My point here is that you can sell the American public anything--no matter how stupid--with proper advertising. Couldn't our industry advertise some intelligent use of computers? Why can't we peddle quality?

GORDON: Your analogy doesn't hold. Those companies got together on their campaign; we aren't allowed to do that now.

GRUENBERGER: But for such a thing as quality, a group like AFIPS wouldn't be accused of conspiracy, or whatever is illegal. Why couldn't one company--say, IBM, for example--start selling quality, and maintainability, and all those good things?

WHITE: Because the competitors would go all out to sell CPU time.

GRUENBERGER: Then how do they sell all those Cadillacs?

WEIZENBAUM: "Quality" in that field is all tied up with "image" and other factors. Let me try a different comparison. Suppose you go to a competent surgeon and ask him to cut off your finger. Unless he's convinced that it is appropriate to some medical problem you have, he just won't do it. He has some ethical standards to go by. There is essentially no free market for what you think you want.

Now, in our industry (and in engineering in general) we have just the opposite situation. The customer doesn't even tell you his problem; he only asks for a system that has a specific behavior, and we seem to be willing to do whatever the customer is willing to pay for. (For example, if there is military security involved, the customer may only describe an equivalent system to the one he wants.) I think that that is the root of the problem. We need a code of ethics so that a professional in our field can exert some responsibility.

WHITE: But the analogy breaks down. The doctor knows the business he's in; he is well qualified in it. We don't; we're not experts in tax assessing or inventory control, or anything but the tool we offer.

GRUENBERGER: What's more, every doctor knows that every other doctor must subscribe to the same code of ethics. In our business, which is now well established with no such code, it will be difficult, if not impossible, to get every single person to subscribe to any set of standards all at once.

WEIZENBAUM: You're quite wrong about the doctor being well qualified. But we know certain things that we know are wrong but which we are willing to do. For example, we are willing to optimize CPU time when we know not only that that isn't the problem, but that it is positively harmful. Everyone says that he must, or his competitors will and will take away the business. That's an ancient excuse for immorality: "If I don't do it, someone else will, so it might as well be me." If you agree to that, you can go out and rob banks.

GREENWALD: But it isn't always easy to tell just what is right. A customer may have a real-time problem that forces him to by-pass all the error checks; he has simply agreed to live with errors as the price he has to pay to get running at all.

Another place where Joe's analogy breaks down is the lack of competition in the professions. They are usually not involved in competitive bidding.

WEIZENBAUM: I realize that what is "right" is seldom calculatable. But I'm advocating copying the first part of the Hippocratic oath: "Do no harm." I'm saying that we should abstain from the things we know are wrong; we need not have to prove that we're right.

GORDON: You are asking us to adopt Asimov's laws of robotics, the first of which says that robots should never harm humans.

WEIZENBAUM: But I keep hearing that we should tell the users to mend their ways, and I'm saying that first we should mend our own ways.

GREENWALD: Do you think that the people who sold Concord the tax program thought they were doing harm? It may be that they are not aware of the implications of what they sold.

WEIZENBAUM: Of course, people can do harm out of ignorance. I can perform minor surgery on my children that a doctor may have to correct later. Let me go back to the medical analogy. Suppose I ask a doctor to do something that is not harmful, but also does no good--he should refuse that, too. In our business, consider the automating of election returns. Even assuming that it does no harm, what good does it do? Why should it be done at all? Is it done simply because we know how to do it? In the case of the real estate taxes in Concord, whatever needs to be done could be done by hand rather efficiently--nothing changes on a millisecond basis.

POLAND: I must challenge you on that tax program. A manual system requires judgement for every decision; this is an advantage. The disadvantage in that there is no way to bring in a third party (on a regular and convenient basis). The tax assessor formerly arrived at a figure by judgement, and you could get a new decision by providing him with new input data, or by pointing out to him who you were (e.g. the town's leading employer). When you automate a system, two things happen. First, the judgements become explicit (not perhaps visible or comprehensible, but nonetheless explicit). Second, the judgements become uniform.

GREENWALD: I could easily write an algorithm that would recognize and act on specific names.

BEMER: You may recall a recent instance in which a piece of property valued at $140,000 one year was valued at $300,000 the following year, in which it was sold, netting Mr. Nixon a rather large profit. Now, you can go to the IRS with a dummy case, using just those figures, and call for an explanation, and that was done. (As it happened, nothing much was done about it.) But the point is that when such things are done by computer, there is a factor of replicability that is totally lacking in pure manual procedures. True, a program can be rigged to respond to certain names, but then it won't reproduce its actions properly with a hypothetical case, and is thus open to public test.

WEIZENBAUM: When you bring in political actions, you're in a different world again. Let's stick to the computer aspects. When you say "explicit," just what do you mean? If all our law books were written in Chinese they would be explicit, but what good does that do you? Jay Forrester's models are explicit, he says, and if you quarrel with the output, then you are reduced to questioning his input data or with the structure of the model. But this doesn't help anyone, like a labor leader, who wants to question the limits of growth; he can't read or understand the model.

Now, you pointed out that a computer program is impartial. This is precisely what is meant by "equality under the law." The law applies, in effect, algorithmically, independent of irrelevant parameters (such as color of skin, wealth, and so on). But the law is also a living thing, and parameters that were thought to be irrelevant when the law as written might be important later. But in any case, one doesn't get justice out of a system by cutting off those parts that make judgements possible. I'm suggesting that in the Concord system there is an abdication of responsibility, which has a side effect that it applies equally to everyone.

GORDON: There is clearly no need for speed in the Concord system; the job could easily be done with a set of rules and a pocket calculator. By putting the process in a computer, the algorithm has been buried and made inaccessible to human recourse. The impartiality is now a by-product of the process.

WEIZENBAUM: For every task we should ask "Why do it at all? What urgent problem does it solve? Why should we spend society's resources on doing this?" For example, take the automation of election returns. The proponents argue that if the polls close at 7 the results will be known at 8. Even if that were true, we should ask "so what? Who needs it? What's wrong with 8 the next morning?"

WHITE: The newspapers need it.

WEIZENBAUM: But they don't need it. Society doesn't need it.

GRUENBERGER: But society is certainly willing to pay for it.

WHITE: They don't need it; they want it.

WEIZENBAUM: I recall hearing Margaret Mead on a television show...

POLAND: Oh? You have a TV set?

WEIZENBAUM: Yes.

POLAND: Do you need it?

WEIZENBAUM: No. Maybe I saw it in a hotel room. Anyway, Margaret Mead was commenting that she had just been in a conference of TV anchormen and she said "I've never been with so many people who all claim to 'give the public what it wants' with no thought of exerting any responsibility of their own." Everybody is simply serving the people, as though there were some meter somewhere that can measure what people want.

GORDON: There are always hidden costs to computer systems. The election return tallying, for example, gives us something, but for that we give up something, in this case the ability to have a write-in vote. It can still happen, but we have made it almost impossible. Supermarket automation is another case in point. We gain speed and efficiency, but we lose some freedom to do comparison shopping. If there's an error, it becomes almost impossible to rectify it.

WHITE: But presumably it's a free market choice. You can choose to patronize an automated market or not.

GREENWALD: Many states are requiring that prices be marked on items so that the consumer can read them. But even so, where the idea has been tested, it turns out that people don't care.

WEIZENBAUM: EFTS is a better example. When it comes and has operated for a while, the step will then be irreversible; you couldn't go back if you wanted to. For most such systems, people do not have a choice. If you, personally, elect to refuse the new system, whatever it is, your life becomes very difficult.

WHITE: Try to write a counter check today.

WEIZENBAUM: Or try to declare yourself completely independent of banks--you just make your own life difficult.

GREENWALD: Every one of us was asked for a credit card when he checked into a hotel here. What would happen if you didn't have one?

WHITE: You might not get a room (even with a reservation), or you might be asked for a cash deposit...

WEIZENBAUM: Which they might not be prepared to accept. The point is that many new computerized systems are introduced with the qualification that you can take it or leave it, but in most cases that soon becomes a fiction.

WHITE: To a large extent, you are trading your degrees of freedom. Before the widespread use of credit cards, for instance, it was difficult to rent a car without a large cash deposit.

POLAND: The credit card is today's credential for strangers. A century ago it was the way you dressed. There is always some device to establish your credibility, if not your credit, in a strange town.

GORDON: One advantage of cash is its anonymity. I'd like to be able to prove my solvency without revealing my identity, for whatever reason. I'd like to be able to buy my wife a present as a surprise for her, without having it appear on our joint account statement.

BEMER: I'd like to comment on Joe's notion of explicitness in the program. I recently wrote a program to compare the old General Electric pension plan with the current Honeywell plan, as it affects the individual employee. When the program is run, it queries the employee for input data (which it subjects to all sorts of reasonableness tests), then calculates the comparison, and prints out all the data, the intermediate calculations, and the results. The employee can see just what was done and how. I think if this overall format were followed in more DP applications, we'd be a long way toward making the algorithms really explicit.

POLAND: Many stores now send you a bill that simply says "Balance due:..."

GORDON: Yes, they have removed a degree of freedom, where before they used to itemize the bills and send you the carbon of the sales slips.

WHITE: But that trend is reversing, and they are again itemizing. In fact, it may even be better now than it was before.

GREENWALD: Some of my bills are itemized all right, but I can't read the codes. I have to save all my sales slips and try to match them up with the bills. This will all get worse with EFTS, I think. They tell me that in 1975 the banking industry processed 34 billion pieces of paper, and they expect to to go to 47 billion by 1980. I don't think I'll be able to pay them what they are going to charge me for that service.

WHITE: But the point is that you can't do without the service anymore. And you won't have a choice.

GORDON: There's another factor in all this that seems to have been overlooked. On the right side of current American Express charge slips, there is a column that is labelled "Delayed Charge; Revised Total." When you sign for a purchase, you are, in effect, signing a blank check, because they can add items in that column after you've signed it.

BEMER: I think we are all agreed on the inherent danger in EFTS. Let me go back to something else. Fred sent us the computer science curriculum from his university. I find there is just one course relating to the storage, processing, care, and feeding of large data structures (out of 22 courses). It seems to me to be out of balance.

WEIZENBAUM: Those numbers may be misleading; it may be one good course, which would be sufficient (and would be one more than there is at M.I.T.). There may very well be 22 distinct facets to computing.

BEMER: But this one strikes me as far more important. I recently played with a tape I got from a government agency; an active data base that they were using. I found, among other things, that it was loaded with extraneous blanks which made the file about twice as large as it needed to be, and that it was in bad shape (e.g., one man listed three times with different spelling of his name). With a good text editor, that file could be cleaned up quickly. This is the subject that should be taught today. I can think of three courses in data bases that should be in every curriculum.

GREENWALD: I prefer to have my name misspelled in various files, so that the data on me can't be correlated.

WEIZENBAUM: Bemer has raised an important point. Do we know enough about handling data bases to be able to teach techniques to beginners?

POLAND: We know the techniques for handling large files and data bases, and we could teach them if we elected to. The techniques use computer time and generally involve a tradeoff of computer time for people time. How many compilers will accept misspelled words? Not very many. How many text search programs will search for misspelled words? Not very many.

WEIZENBAUM: But wait a minute. If your compiler will accept misspelled words, it will accept THAN for THEN and you may not want that.

GORDON: The important thing is that it tell you what it has done, and 95% of the time it will be what you wanted.

GRUENBERGER: All you need is a little feedback in the system.

WEIZENBAUM: I don't know how "little" that should be.

BEMER: The reason I bring all this up is that I think the present methods of dealing with data bases (involving lists and indeces and pointers) are wrong. I want content addressable data bases, so that the content will indicate things like the security and privacy levels, the reliability of the source...

WEIZENBAUM: And you know how to do all that?

POLAND: We've known how to do that for many years, but recently it was forgotten. Even before computers, a bookkeeper knew how he got his results (whether they were right or wrong). One of his fundamental rules was: never erase. He would either correct, with a line drawn through an entry, or he would make a reverse entry. Moreover, he retained all the figures that led to his final balance.

In computing, we threw out this fundamental piece of folklore, not recognizing that it was an excellent idea. It is only just recently that we are getting around to restoring it.

GORDON: The low point in that history was the op-code on the 705: WRITE ERASE. It seemed like a good idea to the designers (you could clean up the output area of storage as you used it), but it made it very difficult to correct errors.

POLAND: Yes, and that was the last machine to use it, too.

GRUENBERGER: Then someone invented READ-AFTER-WRITE, which was a much better idea.

POLAND: The key point is that in the art of trading off computer cycles and computing expense, we went through one phrase from 1955 to 1965 with some validity, but then continued on that same path from 1965 to 1975 falsely. We have lost information, in the fundamental sense of information. I don't think it's necessary, and I see the trend reversing. If I can look into the future, I think we're going to get into transparent computing; that is, computing that is no longer mysterious. It may do things in a mysterious manner, but what it does will be visible. When your account balance changes, the calculations that led to the change will be apparent.

BEMER: And this will be the effect of cheap storage.

GORDON: That WRITE ERASE command was the ultimate in cycle-saving. It enabled you to re-clear your output area at zero cost in cycle time, thus effecting an apparent saving. What it cost you was phenomenal. Today we can see that the extra machine cycles are unimportant; we can put things together so they won't kill us when something goes wrong. Cheap storage is one element in all this, but notice that if we capitalize on it, we promptly get accused of pushing unnecessary storage..

BEMER: One solution is prefix text processing, wherein every line of text, for example, could be an instruction in some language, and the prefix allows me to say that that line participates in versions 1,2, and 3, but not in 4 or 5. Therefore, the records are not physically destroyed as a program is modified. They are only logically, destroyed, and the program can be executed at the level of version 4. Thus, I have an audit trail of my program. The same thing can be done with data.

GORDON: There are a number of ways of doing that, and each way carries a cost. The cost may be extra storage, or the time involved in skipping over unused portions. No technique is free; there are always trade-offs. The important thing is to get people to emphasize the "correct" way of doing the job, to emphasize reliability and security and comprehensibility.

WHITE: And it will be interesting to see the new problems we'll have when we do get them convinced.

GORDON: No doubt. But two things I think will emerge: memory centrality and reliability vs. cycle time. Since these things have been known for a long time, and observing the rate at which wisdom propagates in our field, we might expect some real progress in 30 years.

GRUENBERGER: Who is supposed to start this ball rolling?

POLAND: It is rolling.

GREENWALD: The user has specified what he wants. In any kind of transaction processing system, the users insist on having every transaction journalized; he insists on the recoverability of his data base, its integrity, and its traceability. They will still make benchmark comparisons, of course.

GORDON: The user now recognizes that these things cost them something. IBM's system called IMS tends to be recovery-oriented, and so it's not fast; it works in a reliable, leisurely fashion, which led to the comment "IMS is a long day's journal into night." Another system, CICS, ran much faster, since it did not journalize (the user had to do it himself). CICS has now slowed down, by virtue of adding journaling and backout and recovery. So I tend to agree with Clarence; the trend is there, but it will take a long time to become prevalent.

WEIZENBAUM: Meanwhile, the mini and micro processor boys are getting active at about the level we were 20 years ago, and their machines are becoming interconnected with larger machines.

GREENWALD: For example, the people working on EFTS are using minis as the front-end processors to the system, and using techniques that are over 20 years old.

GRUENBERGER: Do I now detect a note of optimism in this room? You were all painting a very gloomy picture a few minutes ago.

GREENWALD: The places where we are now doing things considerably better are the places where we were forced by user requirements. The world of data processing (as opposed to the world of computing) may force us to do more of the things we've been talking about. Part of our troubles stem from the fact that us senior citizens generally came out of the area of scientific computing.

WEIZENBAUM: We may also be in the same position as the generals who figure out how the last war should have been won. We all grew up with large, expensive machines. Maybe the impact of the new little machines will alter people's thinking.

BEMER: But there's little difference. You can attach a billion-bit store to a microcomputer.

WEIZENBAUM: The real point is whether the minis and micros are going to spread as free-standing systems, or whether they will be connected and communicating with each other, if only through a data base.

[Dick Tanaka arrived here.]

GRUENBERGER: Let me brief you. On the one hand, there has been a lot of gloom here: things are in terrible shape and about to get worse, and no one listens to the wise men. Stupidity is rampant, and everybody is out to repeat all our mistakes for the next 30 years. On the other hand, I've been hearing how well things are going, and how we can design large systems so much better than we used to. I find it very confusing.

GORDON: The real message is that things are in miserable shape, but most of us are too old to be around when the blowup occurs. Seriously, the gloomy view is that things will not get better until terrible things happen to force it (such as an EFTS system that goes down nation-wide and no business gets done for a week or so). The optomistic view is that things are getting better and will continue to do so without a calamity. This view says that people in our trade are becoming aware of the real problems before having to experience a disaster.

BEMER: But not enough of them

POLAND: I think a better statement is that they are experiencing calamities on their own terms, and they are taking corrective action to reduce the incidence of them.

WEIZENBAUM: I must register a vigorous dissent. Some years ago, on the Nova TV show, they showed a western desert with the bulk of a ship in it. The desert had once been a navigable waterway. The Army Corps of Engineers had been busy, over 75 years, building dams for various purposes. Each dam solved some local problem, but the global effect over the years was adverse. Now, I think what you are describing is analogous to this in our industry. We are keeping large systems going by adding patches and thus narrowly averting disaster, but at the expense of making the systems even more complicated. Each patch makes its system more incomprehensible. It may be that the disaster can be postponed, perhaps indefinitely, but at continuously increasing social and financial expense. Furthermore, it becomes impossible to back out; you can never say "this is fundamentally the wrong way of doing business."

WHITE: Or, worse, never being able to go ahead. Today it is getting very expensive to move from one version of the operating system to the next one, regardless of its capabilities and functions.

WEIZENBAUM: We started out here saying that 30 years from now people would no longer be debating Fortran vs. PL/I. Perhaps they will; perhaps there is no way of backing out even of that. The disaster we've already experienced in this connection is the language BASIC, and that is now irreversible.

GRUENBERGER: I doubt that we could have something as disastrous as a complete breakdown of EFTS, but what I think is likely is something like having the credit rating of all the veteranarians in the country wiped out one day.

WHITE: No, there could well be a complete breakdown, through a chain reaction, much like the chain reaction that took out the northeast power grid a few years back.

GORDON: And that could get worse, too. The power grid is now going national, so that when the next blackout occurs, you guys out west can participate along with the rest of us.

WHITE: They patched up the local troubles by going national, in other words.

GORDON: We have a large telephone system, and we've never had that kind of national failure.

GRUENBERGER: No, it just has local failure continuously.

WEIZENBAUM: As a matter of fact, it did break down on November 22, 1963 (the date of Kennedy's death).

GORDON: The system is such that you wouldn't notice a national breakdown. You don't get a dial tone; so what else is new?

WEIZENBAUM: There are many times when you can't dial Manhattan from Boston directly; you get only busy signals.

GREENWALD: How much is reliability worth? It's a question of tradeoffs again. There is probably much more effort going into the electronic switching system of the phone company, which costs a lot more, but it's worth it.

GORDON: Reliability gets to be worth more the more you've been burned by the lack of it.

GREENWALD: Maybe it's a question of education. You have to educate people to their dependency on computer systems. Without them, their business stops.

BEMER: You have to hold fire drills.

WEIZENBAUM: It goes back to quality control. It's like Clarence's warehouse manager, who should be able to query the inventory control system and ask "What happens if?" and get back an answer that he can understand and use. If he can't do that, then even though the system exhibits the specified input/output behavior, it is not acceptable. The word "transparent" seems quite appropriate.

GORDON: Concerning that particular system: is anyone in the corporation looking at that system and comparing it to others for overall effectiveness in doing the job, and is anything being done to write up the experiences and conclusions reached, so that the news will spread?

POLAND: The answer is "yes" to both questions.

BEMER: We have it in Frank Cary's own words: "As we reach out for these opportunities, we know that we have to make the computer acceptable to people with no knowledge of how it works, but who can benefit in their daily lives from what it can do for them."

GRUENBERGER: We seem to have stumbled on to something good. Give me a scenario of how a fire drill should be conducted.

BEMER: You set things up like a counter-insurgency team, with instructions to create a disturbance in the system; then the people operating the system can see how to react and how to recover.

WHITE: The Bell people did that for the electronic switching system by having the destruct team stuff Chore-Girls into the terminals to create random shorts. That's a fire drill.

BEMER: You get teams of people to put super overloads on the system, to see what happens.

POLAND: With any kind of terminal system, there is a breakdown that occurs (perhaps by stupidity) when someone gets to a terminal and types something like

I WANT TO KNOW...

How does the system react to that? Some systems try to parse that, compile it, and do it. Others respond by USER NOT LOGGED ON. There can be many such types of response, and there are subtle variations. I have personally been in this Mad Bomber category, through a simple programming error. I managed to crash a triple 168 system--twice. Before I did it a third time, I telephoned the system's manager to say "I think I'm the guy; am I?" Such crashes exist all the time.

GREENWALD: But I'm thinking of a different kind of fire drill, where you deliberately take a system down.

POLAND: Yes, that's like a ship's fire drill where you actually put the people into the lifeboats and sink the ship. It's a little bit destructive. We should be able to test a system without going that far.

GREENWALD: You're interested in showing them how their system can break down; I'm interested in showing them how much they rely on their system. Until I show them how important the system is to them, I can't convince them to pay for the reliability they need.

GORDON: The function of a fire drill is not to show you ways to make a system more secure (that's a different problem), but to convince people of the seriousness of the system going down.

POLAND: If that's the case, then one should be careful to distinguish among groups. The non-data-processing person will be more easily convinced than the DP professional.

GRUENBERGER: I conducted a fire drill for Clarence many years ago. He had proudly shown me what SABRE would do. Then when SABRE went on line, I found myself in the Newark airport with a ticket to Los Angeles, and I decided to tickle the system. I had the agent type in the proper information, and the system promptly reported that there was no such flight (it was already posted on the notice board over the agent's head) and even if there were, I wasn't on it. I persuaded him to give me the printout, which I mailed off to Clarence. How's that for a fire drill?

WHITE: It isn't--it's a test of the system. A fire drill is supposed to show what to do when the system comes unglued; that is, when a fire occurs.

GREENWALD: It's not the errors of the system; it's the dependency on the system that is in question.

WEIZENBAUM: I recently had the situation where I had the wrong set of tickets for a complicated trip, and the agent at the airport in Vancouver had to create a new set for me. He was concerned that he do everything correctly, because he knew that if he made a mistake anywhere along the line he'd have to start all over. I would think that the designers of his system would eventually take care of that, and write a recovery procedure for him. That sort of prophylaxis could be practiced the way pilots practice flight procedures.

GORDON: We have two different problems here. The first kind, where the whole system goes down, we don't have to worry about--it will happen normally, and they'll learn how to take care of it. The second kind, of the type Joe is talking about, is inherent inside the computer system, and we are responsible for it, and we should know how to deal with it.

WEIZENBAUM: But we are the only ones who can tell them that they should have some feel for how dependent they are on the system.

GORDON: We don't have to do that for the pioneers; for those who installed the systems that are now 15 years old and are thoroughly shaken down. We should worry about the tremendous horde of new users who are less sophisticated and who are not yet aware of the dangers and pitfalls.

WHITE: They don't realize what is going to happen with their automatic checkout systems until two years later, after they are completely dependent on them, when the automatic checkout system goes down.

GORDON: But the systems I thought we were talking about are already so unreliable that they've had to suffer the consequences; you're talking about systems that have gotten good enough so that their users have developed that sense of dependency.

GREENWALD: Xerox Computer Services is a service bureau, running presently 13 Sigma 9's for customers who each have terminals. If a single CPU goes down, the user will be up on another machine in, say, 5 minutes. If a disk goes down, it will take a little longer. But what if the whole system goes down, for any reason? I'd like to know that all those customers are aware of the consequences. When I think of the complex systems we are now creating (for EFTS, but even for Sears Roebuck), you can imagine a situation in which one node goes down, and the resulting chain reaction could bring most of the business activity of the country to a standstill.

BEMER: Bob Patrick tells of an installation that maintained copies of its files off-site for protection, but the only copy of the run book with the procedures for handling those files was kept in the machine room. A properly conducted fire drill would have helped there. The operating personnel should be locked out of the machine room ("It has just been destroyed by fire") and told to recover with the backup files, at which point the weakness in the system would quickly reveal itself.

GREENWALD: This discussion began with the question how do you convince the customers to pay for the features they will need and not rely on stopwatch measures of efficiency?

GORDON: We installed disk storage for United Airlines' reservation system. They were interested in operating speeds, and so on, but they were also interested in time-between-failure and the time to recover if a disk crashed.

WEIZENBAUM: That isn't it. The question is, if a major portion of the system were to be effectively destroyed, what protection do you have?

GREENWALD: Assuming that you can define the quality of the system (in particular, the quality of the software), the question is, how do you get them to pay for that quality?

GORDON: But we keep talking about external disasters. How about the sort of thing Clarence talked about, where the system can be crashed by keying wrong information at a terminal? Then the questions are: How long does it take to notice the trouble?; How long does it take to patch that weakness?; and How many new troubles are introduced by that patch? It's that kind of quality in the software that you want people to be willing to pay for.

POLAND: In our discussion of fire drills, we have concentrated on the theme of demonstrating to end-users what it means to be without their computing services. Now, we have agreed that methods and techniques exist to make an application system reasonably reliable, even though it appears to be quite difficult to get end-users interested in wanting that reliability. We are agreed also, that all it costs to achieve reliability is money. I have not seen anywhere a discussion of how the end-user is to determine what it is worth to him to increase the reliability of his system. How much should he spend to get his system more reliable than it is?

GRUENBERGER: Maybe the thing to do is to show him what it might cost--or will cost--not to increase the reliability.

POLAND: That might lead to the same answer; we don't know. What we want to demonstrate is the effect on a department, like Accounts Receivable, if the computing center just plain lost all their data. Datamation had an article on that about a year ago, describing just what did happen. That's the only case I know where the subject has been investigated.

BEMER: You could find out more if you'd pay Bob Patrick for it; that's one of his stocks in trade.

GRUENBERGER: It has been done on our campus. For most work (i.e., student problems) the answer is that many users wouldn't even know about it, or at least the cost is merely that of some keypunching. For the administrative work (grade records, library records, and so on), the result would be costly, but the dollar value is only someone's rough estimate. But that raises another question: How do you motivate users to make these estimates?

GREENWALD: For most users, the answer to Clarence's question is "I'm out of business," and I think that's the only answer you can get.

POLAND: A computer center usually serves many departments. The answer from all of them will be "I will be out of business," but for at least one of them, I will bet you, the real cost will be 75¢ or less.

GORDON: It's like any insurance problem; it's an expected value problem. What is the cost of that happening, and what is the likelihood of it happening?--it's the product of those two. It's also a function of time: what is the cost for every day that the data is lost before they recover? For example, what is the cost to American Airlines for every day that SABRE is down?

McCRACKEN: That figure is a lot easier to get than the probability of it happening. The strike that United had last year would furnish a rough cost figure, for example, but it would be more difficult to calculate the chances of their reservation system going out for x days. It hasn't happened yet, so how do you figure the probabilities? How would you figure the probability, for example, of someone skimming off ten million dollars from the EFT network ten years from now?

POLAND: That's a standard actuarial problem, like insuring an actress' legs. The insurers play with their secret formulas and come up with a premium value that they judge to be large enough. They're still in business, so they must guess right most of the time.

GORDON: I think it was an IBM Board Chairman who said that anyone can get the right answers given all the data; the trick is to get the right answers when you lack some of the data. We have a question here that users have to answer, and most of them are answering it by doing nothing. They are acting as though it will not happen, or that it will not be too costly if it does happen.

WEIZENBAUM: There's a flaw in the analogy. When the members of Lloyd's price a risk, the members pledge their personal fortunes on it. If they guess wrong, they are personally responsible. The decisions made in business aren't that way; the ones who make them are not personally liable; they are risking corporate funds.

GORDON: That sounds like a good point, but I don't see its relevance.

GREENWALD: He's saying that managers are as irresponsible in their fields as we in data processing are in ours.

TANAKA: Suppose we carry this through, and actually get figures for the chance of disaster and its consequent cost, for any given situation. How is a manager supposed to use that information? We would tell him that for N dollars you can get so much protection and for 10N dollars you get so much more, and so on. There is no point unless you state the situation in terms of choices that the manager can make. At some point, the cost would put the company out of business right away.

BEMER: I feel that all the software fixes you'll ever need cost less than going out of business.

GREENWALD: But there is no real assurance that, for extra money, there will be any real improvement. We've attacked this problem from both ends. We started by asking how you define quality, how do you measure it, and what does it buy you? Then we shifted to the user's need for quality, and how do we sell him on buying it? We then have to prove our case, and for that we're back to defining quality and measuring it--we're in a vicious circle.

GRUENBERGER: It's like selling burglary insurance; what you need, at regular intervals, is some good burglaries.

TANAKA: But even given that, you have something you want to sell, and you tell the customer he can have the plain model or the deluxe model. He then asks how much more the deluxe model costs and what does it buy him? You can't just say "Trust me, baby; it's better."

WHITE: But it isn't quite that way. Nobody worried about auditing computer files until Equity Funding came along. That was the same kind of circular argument--until it happened.

WEIZENBAUM: One reason we're in a circular argument is that we're trying to put the responsibility on the end user, when it is our responsibility. It's like the automobile industry. The industry wasn't interested in safety, nor was the safety council; they put the burden on the driver and on laws. Part of the argument was along the lines of "If we don't make unsafe (and hence cheaper) cars, someone else will." With that kind of morality, you can do anything.

WHITE: But they're not making cars safer because they want to; they were forced to, by legislation. And the same thing may happen to us.

WEIZENBAUM: But we shouldn't wait to be forced; we should agree to sell only quality products.

WHITE: But suppose some auto maker, ten years ago, had decided to make non-polluting cars that would cost 70% more than those of their competitors and had appealed to the users, as rational right-thinking people, to buy only their cars--they would have lasted, at most, about a year.

BEMER: But that comparison is unfair. We can make quality software and do it at less cost than what we're doing now by doing it right. We don't do it now because of ignorance of software management. It's not willful ignorance; they just can't comprehend the problem.

GREENWALD: List for me the characteristics of safe software.

BEMER: That's tough. I picked up the word "safe" from the auto analogy.

WEIZENBAUM: Let's try the word "comprehensible," or "maintainable."

WHITE: And if that is to be competitive, then we have a whole new set of problems.

BEMER: Software costs are a one-time situation; the replication costs are near zero.

WHITE: Not true. Software changes, every day.

BEMER: I agree, but we should plan on those changes.

GREENWALD: I'll change my question. Give me a list of characteristics of software that is comprehensible.

BEMER: First, it should be made up of piece-parts.

GREENWALD: All right. How big are the pieces?

GORDON: One page.

WHITE: A comprehensible size.

POLAND: How many (head count) system programmers did it take to make it? To maintain the program, does it take one part-time man, say four hours a week, or two full-time men, or 14 full-time men? I believe that this furnishes a scalar measure.

GREENWALD: When you say "maintain," are you talking about bug fixing, or enhancement, or both?

POLAND: Both.

WHITE: The number of people needed depends on the functional complexity of the program. If you need more functional capability in your system, it will require more manpower to maintain it.

GREENWALD: I have a better criterion; namely, the amount of time it takes to train someone to be a maintenance programmer. I'm trying to point out that we don't have many measurable criteria on which to evaluate software. Many times it reduces to a matter of taste or style.

POLAND: Let me return to the inventory example I used earlier. There is another such system, in another plant, that is hard-coded in PL/I. That one requires two men full time to maintain it; to keep up with the changes that occur. The other system (the one I cited earlier) requires zero people to maintain. Both systems purport to do the same job. I submit that there is a big difference in maintenance between those two programs, which is largely a difference in comprehensibility to the user.

WHITE: There probably is a difference in that particular case, but it may not be true in general. In either case, you have a percentage of the available time (somebody's available time) spent on maintenance whether it be a "programmer" or a non-programmer. There may be a crucial difference as far as maintaining the programs is concerned, but not as far as the company is concerned. It still may take two people at the same salary level.

BEMER: No, he's not talking about that. Clarence is talking about practically an order of magnitude difference in ease of maintaining the better program.

GRUENBERGER: Clarence has a nice clean case: two different programs that do much the same job. We don't often get a chance to compare things that nicely. But to go back to Irwin's point: isn't it comparable to being able to recognize a good omelet without being a chef? There seems to be a passion for assigning numbers to things, as though that nails down some basic truth. Would word counts or other mumbo-jumbo establish that Dan writes better books than other people? Quite apart from the technical content, they are recognizable as well-written books. Do we have records of brush-strokes/hour for Rembrandt and Vermeer? Would such data help us in any way to establish that the paintings by those men are better, in any sense, than those of a thousand other painters? My point is that there are means available for certifying a program as good (comprehensible, maintainable, portable, modifyable--whatever you please) without having to attach numbers to it. Of course you can count the GOTOs, but such schemes will only reveal the extreme cases in either direction; they cannot discriminate in exactly those cases where you need a discriminator.

BEMER: I did an application in two days that had been turned down on the grounds that it had a cost estimate of $10,000.

ARMER: Now, was that because of the tools you used, or because of Armer's Law which says that the cheapest way you can get a job done is to hire the most expensive people you can find to do it?

BEMER: Well, I do cost more than those guys, but mainly it was due to the tools I used.

GREENWALD: But didn't you--the high priced guy--help to create those tools?

BEMER: No, they were bootlegged.

GREENWALD: I'm worried about Clarence's criterion of quality (the number of maintenance people required). If that were applied generally, we would start to get terrible maintenance from our vendors because they would start to say "I used only one person."

ARMER: You'd have to hold constant the level of maintenance.

GORDON: Clarence didn't say the number of maintenance people being used; he said the number of maintenance people required.

POLAND: Those two PL/I men have to be there even if they have nothing to do, because they are preserving a body of knowledge that exists only in thier heads. You dare not assign them to something else, or let their number get down to one. Their turnover rate is always higher than you want. With the other approach you don't have that problem. For one thing, the turnover is within the operating department to whom the application belongs. The critical thing is not the number of man-hours, but where they're spent.

WEIZENBAUM: Let's try a different test of quality. Can the end user diagnose trouble in his own terms? In Clarence's inventory application, for example, if the program reports that an item is short by 50,000 and the user sees the normal 400 items on the shelf, he knows that the 50,000 item is outrageous. He can now go to the persons responsible for maintenance and tell them that "There is trouble in the order acquisition routine." The basic question is, can he do that, without knowing anything about disk heads, file structures, PL/I variables, and so on? If he can, then I'd say that it is a high quality, maintainable system.

GRUENBERGER: Should you not go one step further? Can he sometimes correct a trouble at his console? That's maintainability in the other sense. His operations, of course, are still in his own terms, and he must be able to satisfy himself that he has patched the trouble, and not just its symptoms.

GORDON: If he has to go to the professional "maintainers," however, it should take them less than 6 months to find and correct the trouble.

WEIZENBAUM: If the system is so clean that the user can perfrom correct diagnosis, then I think it almost follows that it can be maintained properly.

GORDON: I can't agree. The user will claim that he has found trouble in the order acquisition routine. The programmers will then recall that that routine interfaces not only with the inventory program, but also with accounts receivable and payout, and they will start to scheme how to patch it to take care of the diagnosis without affecting the other programs. This end-user criterion is a nice one, but I think it evades the case where the system was badly put together in the first place.

WEIZENBAUM: And I believe that you can't meet the criterion I've stated and have that condition. If there is cross-talk of the sort you describe, then I think my criterion is defeated.

GORDON: You may be right, but I'd still like to see my requirement added explicitly.

WEIZENBAUM: I have in mind an enormously complicated system: the MAXIMA system maintained by Dr. Joel Moses on the KL-10 at M.I.T. It is full of fantastic symbolic mathematics; it is probably the world's greatest applied mathematician at this time. The programming was done in LISP, of all things. Nevertheless, when trouble ma~~i~~nifests itself, it does so in mathematical terms. The mathematician must know what he is doing, of course, but he's operating substantively as a mathematician, not as a computer hacker. He can dig into the system and say "This is where it is," and then someone else can fix it in computing terms.

GREENWALD: I'm going back to the inventory system, since I know more about such things. I believe that the ability to diagnose trouble to the point where you can say "It's in the order-entry program" would take a degree of skill that most people don't have. All the pieces of inventory control are interwoven (and interface with other systems), so that by the time the manager notices that his inventory status report is all screwed up, he can't possibly deduce that the trouble started in the order-entry program.

WEIZENBAUM: Well, I'm not saying it should be easy for him, but only that he should be able to do it, and entirely in terms of inventory rather than in computing terms. If the system is comprehensible, then he can do it, in the sense that it can be done and he is the appropriate one to do it.

GREENWALD: Then I guess that the old manual systems were not comprehensible.

WHITE: There are levels of comprehensibility. I don't believe that a system has to be comprehensible to the man way out at the end, or even someone a few steps removed from the end. It does have to be comprehensible to some group, made up of those who are charged with maintaining it.

WEIZENBAUM: I'm trying to make a distinction between, say, the computer technologists at American Airlines and the people responsible for keeping the reservations system going. If the latter group can do it, then I say it's a comprehensible system. The system has to be transparent to them, in their own terms.

TANAKA: The amount of skill and discipline needed to get the program into shape for them to do that in the first place is at least equalled by the continuing effort to keep the program in that state. As the maintainers fool around fixing things, the great temptation is to have the distinction get more and more blurred.

GRUENBERGER: I think we're overlooking one stage here. At the lowest level, Joe wants the user of the system (say, inventory) to be able to observe trouble; as for example, an order for 60,000 light bulbs when our company uses only 20 a month. Let's say he can detect the trouble and ask the maintainers to fix it. Now, will he also be able to tell that they found the trouble (as opposed to simply changing the amount for light bulbs) and did indeed fix it, and moreover fix it for lead pencils? In other words, if the system has sufficient feedback so that the user can continously verify its accuracy, then you have a really comprehensible system.

BEMER: My programs have limit checks (high and low) on every variable. When something goes out of bounds, I know it.

GRUENBERGER: But you don't know about the ones that are wrong and still pass your limit checks.

BEMER: No, but I have a lot better chance of correctness than most programs.

WHITE: I don't think you can build systems good enough for what you people are requiring. They rest on compilers, which can and do contain logical bugs, and those compilers rest on assemblers, which can contain bugs, and both of them function within an operating system which is known to have bugs. You are asking for people to erect a structure on solid foundations, only we haven't built those yet.

GREENWALD: One trouble with this discussion is that we have taken all our examples from disciplines (like inventory control) that are fairly well known and understood. We will be in real trouble with the new things (like EFTS) that we don't understand.

WEIZENBAUM: Then we shouldn't do it until we understand it. It's very simple: don't rush in.

WHITE: But who is the "we" in that observation? Someone will do it, and no group can prevent it.

ARMER: That's where Joe's analogy with doctors breaks down; there are lots of people willing and eager to rush in before they understand the problems.

GORDON: What's more, if that were a general policy (don't do it till you understand it) we'd never get anything done in our business.

GRUENBERGER: Everyone in computing has observed that the first time you can even begin to understand anything is after you've programmed it the first time. In the case of something like EFTS, that first program is very likely going to be the one put into operation.

WEIZENBAUM: The point is that the responsibility is ours--it can't be passed to the users.

McCRACKEN: I think you're being too harsh. Were the Wright brothers responsible for all the implications of flying?

WEIZENBAUM: The Wright brothers were risking only themselves. As we discussed this morning, it would be possible to train someone to fly a 747 in six months or so. What you want and get, though, is a crew that can still function when things go wrong with all the automatic devices and servos. Even if all the navigation aids in the world were to go out, the planes in the air could still navigate and land safely.

GORDON: One reason for that is the tremendous redundancy (duplexed and multiplexed devices, and backup systems) built into commercial aircraft. I've heard that few commercial flights take off with everything working properly; there is always one system out. But they can still operate safely because they have engineered huge safety factors into every facet of flight. We don't do that with our computing systems. If one component goes out, a large part of the system may go down.

POLAND: Not so. The major airlines all operate at least two computers (duplexed, but not operating in parallel), either one of which could perform the desired task.

GRUENBERGER: And both connected to the same power source?

GORDON: You're saying that the airplane industry, having learned their lesson from the safe design of the aircraft, is now willing to spend money to buy safety in other areas.

POLAND: It's not only the airlines; pick any industry you want.

WEIZENBAUM: I'm not arguing for that kind of protection. The real protection in flight is the intimate knowledge the crew has about the theory of flying, which enables them to understand the aircraft system and, when necessary, override it. They could operate even if all the backup systems went out, because of their understanding of what is going on. And it is this lack of understanding that is failing in our industry.

McCRACKEN: I think your analogy is weak. A fishing boat captin, using LORAN, can go out 200 miles and find the place he was at a month earlier within 50 feet. There is no way he can do that with a sextant.

TANAKA: Further, the analogy called for a failure in one component. What if the primary system (say, all electrical power on the airplane) went out? That's the kind of computer system failure we're talking about.

WEIZENBAUM: I'm not making my point clear, and perhaps flying isn't the proper analogy. The point is that the responsibility for what is going on has not been abdicated to the technology. The responsibility for comprehending what is going on remains in human hands. In our computing systems, when things go wrong, the responsibility has been lost; nobody knows what to do.

WHITE: There are systems that are working, and nobody knows why they are working, and probably no one ever did know.

WEIZENBAUM: Precisely. No one knows how to repair them, and the door to future development is closed.

BEMER: I'm running a photo composition system, and some software in the operating system was changed four months ago (written in an unsupported language). I have to limp along with the thing; it keeps grabbing core up to 410K. It gives all the wrong signals, but it gives correct answers in the end if you can nurse it through. I'll have to get off that system. A lot of major systems with such problems are going to die eventually.

GORDON: We've all lived with systems where you can patch just so far, and then you give up and rewrite the whole thing because the original was incomprehensible, and its author is gone.

McCRACKEN: But we do know today how to write better programs than that. We can't put a precise measure on the quality, but we can recognize higher quality workmanship. At the very least, we can label a program as "terrible."

WHITE: It depends on your definition of "terrible." If it's maintainability, I'll go along with you. But to most people, it's only the speed of execution.

McCRACKEN: I'm thinking of the worst program I've ever seen: a COBOL program in which every paragraph contained an altered GOTO. It is beyond human comprehension to find out how that program works (but it does work, as long as the formats aren't changed). The company using it was going out of their minds, but they couldn't afford the time (they claimed) to rewrite it. That, I submit, is a terrible program.

WHITE: But different people at different times have different metrics. A program may be excellent for most of the criteria we've listed, and still be unacceptable. It could be, for example, easy to maintain, easy to read and understand, and it may take 10% longer to run, and in that particular case, that may be something we can't live with.

GORDON: But that's the exception, not the rule, and most often we haven't faced up to the rule. The bulk of programming does not meet the primary requirement, which is maintainability.

McCRACKEN: In most practical applications, comprehensibility advances all other metrics.

WHITE: Except that the user won't believe that. And that includes 90% of the people I run into.

GREENWALD: We talked this morning in terms of software produced by vendors or software houses. In my experience, that software is almost always better than software produced by the users.

GRUENBERGER: The first answer to that is, "It better be." But, realistically, vendor's software tends to be general purpose; it will be used by many people; its authors must be conscious, as they write, of many strangers effectively looking over their shoulders. User software, on the other hand, is usually special purpose and used only within a small group, and thus tends toward sloppiness.

GREENWALD: But I conclude from that that pressures exerted on the vendors are not going to yield a significant improvement in their software.

GRUENBERGER: I see a reason for optimism in the emergence, in the last 8 months or so, of a bunch of really good books. These include Fred Brooks' The Mythical Man-Mouth, Joe's book Computer Power and Human Reason, Dan's new book on Structured COBOL, and Kernighan and

Plauger's Software Tools (plus some others). It seems to me that the appearance of so many good books suddenly is an indication of maturity in our field. A lot of thoughtful people are now considering such things as: what did we do wrong?; what are the basic principles?; how should we do things right? We now do have literature available on how things should be done. It is my personal opinion that the book Software Tools will have a profound influence on our industry, all for the better. But all the books have an underlying theme of quality, comprehensibility, and maintainability.

WEIZENBAUM: I hope you're right. I recall Arthus Koestler's comment: "The libraries of the world are filled with knowledge of how to live the better life; unfortunately, the distance between the library and the bedroom is astronomical."

GREENWALD: Is the Kernighan and Plauger book being used anywhere as a text?

GRUENBERGER: Hardly; it just came out. But it will be, you bet. It's such a lovely piece of work; if you haven't seen it, don't wait. The books builds a set of building blocks, working up to things like a text editor. What grabbed me was that they used that text editor to produce the book itself.

BEMER: For the last 6 years, the first thing always chopped out of our software budget was software tools. This year it wasn't, so I can see some encouraging signs, too.

TANAKA: Earlier, we drew a comparison to the auto industry, and I'd like to return to that. Suppose IBM decided to lead the way in producing the kind of software we've been dreaming of? IBM's position is much different from that of GM in the auto industry; like it or not, they set the pace in hardware. What if they set the pace also in quality software? Bemer says that it actually wouldn't cost any more, and might even cost less.

GREENWALD: They've done it already, but not from altruistic motives; they were worried about maintenance.

GORDON: Sure, it costs less in the long run, but the marketing decisions are always made from the short run. It doesn't appear to cost less. It actually does, when you consider the fewer reruns and screwups, but those things don't show when you propose it, benchwork it, and cost it out. When you measure one run, in which nothing goes wrong, it runs slower, and that's when the decisions are made.

BEMER: It doesn't even happen that way. When you design a program correctly, using piece parts and all the techniques we know work well; when you copy the techniques that are proven in manufacturing, the whole thing costs less. I'm talking about dash-number programming.

GREENWALD: But the parameters--the things that are going to be subject to change--are being guessed. If you guess wrong, you find out two years later than you're not modularized in the right way. To add one new capability, you have to change ten modules.

GORDON: But it doesn't matter; modularization has an immediate cost. It's true that for release 2 and 3 you'll save a bundle because of the modules, but for release 1 it runs slower because of the linkages between the modules. You'll get it back, to be sure, but it's release 1 that you have to demonstrate, cost, and benchmark.

BEMER: But the history of the last 20 years ought to make that lesson clear.

GORDON: It is clear, to you and me.

WHITE: But it's not clear to the customer--he didn't go through that history himself. Let me give you a typical case in point. We have customer A who are strong users of Mark IV; they believe what we say. They have used it for 5 years. Their total maintenance cost (changes, updates, and fixing bugs) is 15% of their computing budget. They pay in terms of small amounts of run time, due to all the checking features of Mark IV. Now consider customer B. They have had the system for 2 years, but they have a different point of view. They took one of their files, of 100,000 records, and split it into two sub-files and made some runs on the sub-files. The run time, using Mark IV, was a minute and 22 seconds. They redid the source program in assembly language, and the CPU time was, say, 34 seconds. And then they conclude "We can't afford to use your system; it runs 3 times slower." And in company B, maintenance runs 60% of their budget.

GORDON: And they'll keep it at 60%.

WHITE: They sure will. All they can count is CPU seconds.

GORDON: A clean case of sub-optimization. They should be educated to forcus on larger things.

McCRACKEN: I'll make a suggestion for IBM, to enable them to make a quantum leap toward professionalism in this area: refuse to run benchmarks.

WHITE: We've tried that, and we've had to walk away from a sale.

McCRACKEN: But you're too small; it has to be initiated by someone as large as IBM. This sub-optimization--looking only at CPU times and memory sizes--is even now uneconomical, and bound to become really stupid in the future.

BEMER: IBM is in an ideal position to do that at this time.

AMER: Perhaps the time is ideal for another change. We have known for some time of things that could be done in hardware that would make security a little easier. But there is such concern over having old programs run on the new hardware, that these changes are not made.

GREENWALD: Actually, they are made, but then by-passed. The software people manage to circumvent those features, so as to be able to run old programs unchanged.

GORDON: You can't overlook the incredible investment in applications programs.

GREENWALD: Well, it's also hard to predict the future. We came out with an operating system in 1967 that was aimed primarily at the scientific time-sharing user. We are now running commercial transaction processing on it. That shift automatically makes that software a real kludge. It's 9 years old, and it has evolved, twisted, and turned. Its parameters were all wrong, because we couldn't foresee the future. I think that's true of more programs than people are willing to admit, and particularly in the applications area. (We noted earlier that vendor software tends to be better, on the whole.) I agree that better software out of IBM would have a large impact.

GRUENBERGER: We seem to bounce up and down like a yo-yo. Are things going to get better, or are they going to get worse?

POLAND: Both, of course, There will continue to be microsecond counters and bit chasers and other such experts (the typical output of universities) who are chasing that level of excellence. The vendors will have them, but the applications areas will have more of them.

GREENWALD: If the SILT report is correct, they will increase, just by the sheer numbers of new programmers.

WHITE: Sort of a Gresham's Law effect.

POLAND: On the other hand, we will see concurrently facilities that will permit those people to do their thing (and take the consequences) but will also permit the construction of computing systems wherein bit-chasing is not permitted to happen. These will be end-user-oriented facilities. I foresee a real dichotomy, around the year 2000, much worse than what we have now. Some of us can see it now; I think the split will be clear to all, long before 2000.

GREENWALD: But there are signs of progress. Consider that in the late 50's, people refused to use Fortran because of its inefficiency. Now nearly everyone uses higher level languages as a matter of course.

McCRACKEN: Not quite. The microprocessor people are at that stage right now.

GREENWALD: Yes, and we commented this morning that perhaps it won't take them so long to wake up. For that matter, Intel has a PL/I subset for their chip processor.

McCRACKEN: Did you also discuss the effect of ever-cheaper hardware costs?

GREENWALD: Yes, in the sense that the reduced costs apply to everyone's hardware, and therefore if you still have benchmarks, you'll still have these problems.

McCRACKEN: But in 30 years it should be obvious to everyone that software costs far exceed hardware costs.

WHITE: But that's obvious now.

McCRACKEN: It's obvious to me, but it doesn't seem to be to many applications managers; they're still running benchmarks. You could go to them and say "For an extra dollar I can give you another million bytes of core and speed the thing up by a factor of ten--would you still want to do that?"

WHITE: Which is exactly what we've done, and they're still doing it wrong.

TANAKA: We may be somewhat unfair. Not everyone outside this room is a dummy, and many of them may have made sound decisions based on many factors. We are describing a trend, and it should accelerate, but I think it's true that the cliche of software being more expensive than hardware is more and more being promulgated.

GREENWALD: And even now, at least among vendors, we are no longer chasing CPU cycles. We now chase I/O bandwidth cycles.

McCRACKEN: I meant it somewhat differently. If I said to someone "For the same money, I can give you 10 times as much core and make it 10 times faster"--could you do anything with it to alleviate the software problems?

GREENWALD: Not if Parkinson's Law (the work expands to fill the available time) holds.

BEMER: When we run a program under time-sharing, we now bring everything in; we would do better to heed the law that says that 20% of the stored material accounts for most of the activity. We should bring in the rest only on an exception basis, and then no matter how you tried to exhaust the memory space, you'd still be more efficient.

GREENWALD: Our data disputes that; it's better to bring it all in. We convinced ourselves that demand paging is not efficient in our environment. "Working sets" don't work for us.

GORDON: The fact that software costs exceed hardware costs has been with us for years. Tom Watson Jr. commented on that many years ago at a SHARE meeting, and he concluded that the solution was very simple: raise the cost of the hardware.

McCRACKEN: It's my feeling that if you could specify just what you want in hardware, it wouldn't help relieve the software problems much, if at all.

WHITE: You might solve one set of problems, while creating a new set.

GORDON: The nub of the problem is this: hardware represents the sum of the things we know how to do. We leave the rest to software which, by definition, represents the things we'd like to do but don't understand. Remember the 701, which was a spartan machine. It had no error correction, no floating point, no index registers, and a saturated op-code. All of that good stuff was in software subroutines. As we learned how to do those things, they got cast into the hardware. Our good software has always been the leading edge of what will next be put into the hardware.

WHITE: It's what I just said. The hardware people gave us a way to access data in the form of disks. The software people promptly created data base management systems.

BEMER: When the 704 came along with floating point hardware, there weren't too many programs around using subroutines, so it was relatively easy to take advantage of the new feature.

GREENWALD: I will predict that, in 25 years, hardware boxes will be the run-time facilities of some higher level language, somewhat akin to what Burroughs has done, but with more facilities.

WHITE: Most of what is done today in systems programming will be done in the hardware.

GORDON: Will the user be able to specify the hardware to do that?

GREENWALD: No. We're beginning to do it today.

WHITE: Will you need to specify the hardware? Will you really care?

McCRACKEN: We've taken care of cost. Now, suppose we could offer SORT as a machine function--would you want that?

GORDON: We tried just that in the IBM 703. It turned out that such a special purpose machine (it was planned as a tape sorting machine) had nearly all the necessary elements of a general purpose machine. So why bother? Build the general purpose machine, and do with it what you will, like tape sorting.

McCRACKEN: Would you want MIS in hardware?

GORDON: You'd probably want most of the sub-functions of MIS in hardware.

McCRACKEN: But if you were given all the facilities, would you know how to use them? Would you know how to test that they worked correctly?

GREENWALD: For the things I'm talking about, yes. For a language like PL/I, I can specify what it is that hardware could do to help implement it.

WHITE: In building a data base management system, I can envision having hardware do such things as "Give me a record with this key value."

McCRACKEN: OK, I can see such things coming. But take Irwin's PL/I example. Carrying it to an extreme, we would get a machine having PL/I as its machine language. That would be a disaster! We don't know how to design languages well enough.

GORDON: Precisely the point I made earlier; that's why language translators are still software. When we do know how to design languages, then it will move into the hardware.

POLAND: There are machines today whose internal language is APL--and so what? It doesn't do a dammed thing to help getting an application on the system. It has not helped to solve the software problem; the fact that APL is the machine's native language is an irrelevancy.

WHITE: Most applications programmers who work with IBM hardware spend more time debugging their interface with the operating system than they do debugging the applications themselves. Now, if the functions of the operating system were put into the hardware, that would help relieve part of the software problem.

McCRACKEN: I doubt it. It would still take me six months to find out how JCL works.

WHITE: I don't think that that's necessarily true. I agree with Barry; once a function is understood, it can be put into the hardware.

GORDON: We've seen it happen many times. Things that were explored in software (like emulation, or paging) until they were understood then moved into the hardware. Ultimately, I can see JCL being cast into hardware. Even programming structures (like IFTHENELSE) could be put into hardware. Again, it's too early to do it, because we don't thoroughly understand these things yet. They aren't standardized and they haven't settled down. When they do, and we put them into the hardware, we can devote more attention to the applications.

McCRACKEN: If PL/I were in hardware, it would be just as bad a language.

GORDON: Which may be why it is not in hardware.

WEIZENBAUM: There is a LISP machine being built right now. I think it will be great.

GREENWALD: As these things get cast into hardware, I think the reliability of the system will increase. At the very least, it will enable us to isolate the source of troubles.

POLAND: It is a socially accepted practice today for anyone getting a software package from any vendor to mess with it. If he gets a hardware package from a vendor, however, it is not socially acceptable to mess with it.

WHITE: It may not be contractually acceptable to mess with software--ours, for example.

McCRACKEN: In 30 years, what we are calling hardware will then be PROMs, and people do mess with those, even today, and get themselves in deep trouble.

WHITE: I see one encouraging note. We have refused to release source code for our software products for many years, but just in the last few years we are getting people who say "Gee, that's great. The fact that you don't supply source code is one of the reasons for us to buy it, because nobody is going to be messing with it but you, which gives us standarization by definition."

GORDON: It takes a while, but people are learning, aren't they?

ARMER: We have now found one good omen for the future.

POLAND: Even if it doesn't work. It might actually be better to have it wrong and not have people messing with it.

GREENWALD: I'd like to go back. Suppose we here were a consulting firm and we were hired to quantify the quality of the software that various firms had produced in response to a bid. We could perhaps label each piece as "good" or "bad" but how could we rate it closer than that?

ARMER: To put it another way, how could you teach someone to do that?

GREENWALD: If I can't quantify it, I know I can't teach it.

WHITE: You could pick one element (say, maintainability) and arrange to test that and rate it.

GORDON: I'll tell you how to quantify it: gradually. Start with "good" or "bad," and you have a scale of 2. Dan cited a program full of altered GOTO's--that would rate zero. Then maybe you could move up to a scale of 4. The fact that you can't jump to a scale of 100 right away shouldn't keep you from starting a rating scale.

BEMER: You could look at the code and count the occurrences of *+ and *-. There are many such mechanical schemes for weighing software.

GREENWALD: I suppose you could count the number of COMMENTS and their meaning.

McCRACKEN: If the code is readable without any COMMENTS, it could still be good.

GREENWALD: That's perhaps true, but I have yet to see any such code.

GORDON: I might try reading the code and count how many times I have to go to its author and ask him what he was doing. In other words, is it readable to a reasonably competent programmer?

McCRACKEN: That is, do you at least think you know what it does.

GREENWALD: You would also need to know something about the problem being solved.

WHITE: There are two levels of comprehension here. There should be COMMENTs that tell what the algorithm is (which should be there, but not counted toward the quality of the program) and other COMMENTs telling how the programmer implemented the algorithm, and it is the latter set that measures the quality of the program. If there are many of the latter, it probably means that the programming language is difficult to understand.

GORDON: I doubt that it can be that simple. I would look at some of the structured programming constructs: each routine on one page; single entry, single exit; the number of things I have to keep straight in my mind at one time--things like that. Remember the classic difficulty we used to have? We'd get kicked off the machine due to some error and then find ourselves asking "How did I get here?" One of the beauties of structured programs is that you don't have that question any more; you know how you got to any given point. With all these ideas, I'd say that we were already beyond a scale of 2; we could probably rate a program on a scale of 10 without too much disagreement. We're not that helpless anymore.

WHITE: But the problem is that even if you could get agreement (on standards of quality) among people like us, you still can't get very widespread agreement, and particularly in the market place. We built Mark IV to be the kind of system we've been talking about. You can enter a module only from the top, and you will exit only at the bottom and you can't do anything in between. Well, the users forced us to add branching operators. At first we allowed only forward branches, but almost immediately we were pressured to allow backward branches , too, to enable some operations to proceed a little faster. Every vendor is pushed this way, under the threat of not selling.

ARMER: But you told us earlier that sometimes such battles can be won (when you said you don't release source code).

WEIZENBAUM: Can we reach some conclusion as to where we might head in 30 years? What would someone who today considers himself far advanced think of what we are conjecturing? Take, for example, Carl Hewitt and the program he calls GOAL. In that program, as the name implies, one states the desired goal and some clues about it, and GOAL goes about getting there somehow (perhaps by backtracking). Now, with hardware getting very inexpensive, and the possibility emerging of designing hardware for specific tasks, these people might argue that there is no economic barrier to doing business like that. We here have assumed that things will evolve slowly in a straight line. I am more inclined to think that there will be significant changes, and they don't have to be labelled "artificial intelligence" in the sense that that term is used today. But when we look 30 years into the future, we have to consider the possibility of machines that "know a lot more" than machines "know" now.

BEMER: Suppose you wanted to create an operating system comparable to the ones we are now familiar with. What would you specify to a program like GOAL or PLANNER? How complicated would PLANNER have to be to have something like an operating system as a goal?

WEIZENBAUM: Your question may be of the following nature: "I will build this thing called radar, which will be good for locating ships and airplanes, but I can't figure out where to mount it among all those sails." It may be that the future that people like Carl Hewitt (and others) foresee, simply won't include questions like the one you raise. Work along these lines is going on among the AI crowd (those who haven't been converted to cognitive psychology). Perhaps people won't be building operating systems. Part of what I'm talking about is visible with the LISP machine I referred to. It runs in LISP; no compiler, no interpretation, just LISP. You talk to it in LISP, and you can enter a program that says "pursue this goal, using this data base," and it takes off, quite fast. We should consider the implications of such advances, and decide whether their effect will be good or bad.

GORDON:. You are reinforcing Clarence's point about the dichotomy that we will have. While all these advances take place, our field continues to grow, and the backlog of things that people feel must be preserved also grows. I am dealing with a customer now who complains that one of his programs that ran on his 360 won't run on his 370. It turns out that the program involves a 1401 emulator which is running a 650 simulator--and he wants this program to run once a year (and claims he can't be bothered rewriting it). He wants that chain to extend into the future. What do we do with crazy demands like that?

BEMER: Someone should collect old machines, and make them available on a sort of ARPA network, to run old programs on their original host machine. Alternatively, perhaps with our LSI technology, we could make a 650 microcomputer for those people.

WEIZENBAUM: Does that man understand that 650 program?

GORDON: Of course not; there is no documentation.

WEIZENBAUM: Then he shouldn't be using it.

GORDON: But he does know its input/output behavior; it does its job, and he wants it to go on.

WEIZENBAUM: But if he knows that much, either he can describe its function (and it could be readily recoded) or he will believe any answers it gives.

WHITE: He knows that the answers it gives him work for him.

GREENWALD: "I know when I tickle her, she behaves in a certain way, and I don't care how she works inside."

WEIZENBAUM: My point is, if one doesn't understand the program one is using, then one shouldn't be using it.

GORDON: I agree, but I'm not privileged to tell him that.

WEIZENBAUM: If you don't understand the program, then your offer to transliterate the 650 code would be an irresponsible act.

GORDON: Well, my management regards it as an irresponsible act to tell a customer to go to hell.

POLAND: There is no point to debating issues like that one; as someone pointed out, there is always at least one person who _will_ do it, for money.

WEIZENBAUM: And if you go along with that kind of morality, you can rob banks.

GORDON: I'll have to object to that use of "morality." We are not doing it on the grounds that if we don't, someone else will. We feel a responsibility to a customer who was led down this idiotic path by us and is now dependent on it.

WEIZENBAUM: Suppose that program is used in an insurance company. That means that someone is going to get screwed on rates, perhaps, and no one will even know it's happening.

GRUENBERGER: But Barry is pointing out that the company is backing up a chain of events (for which they are responsible) that goes back all the way to the 1401 days when no one (including all of us) even thought there _could_ be moral questions in computing procedures.

McCRACKEN: Just what is the moral issue involved here?

WEIZENBAUM: The fact that this 650 program is incomprehensible to everyone using it. Only its I/O behavior is known.

GORDON: But the only thing different here is the age of the program. Incomprehensibility is the standard of the industry.

WEIZENBAUM: Yes, and that's what I'm decrying. Decisions are being made (e.g., insurance rates) on the basis of programs that no one understands.

WHITE: Fred has an SR-52 programmable calculator. He knows nothing about the workings of its chip (much less the algorithms it uses to calculate logarithms to 13 digits), but he knows a great deal about its I/O behavior--and that's all he needs to know to make effective use of the machine.

GREENWALD: We have a man who uses a black box, let's say, that outputs insurance rates on demand. He's a satisfied user. We seem to be decrying his lack of knowledge of the algorithm involved. I see nothing wrong in the lack.

WEIZENBAUM: If he can characterize the function that relates the input and the output, then we should write a new program for him.

GORDON: He can do that, but he doesn't want to spend the money for that; he argues that this one should work.

WEIZENBAUM: Why don't you write a direct simulator for the 650 on his 370?

GORDON: Who is to pay for that?

WEIZENBAUM: The same guy who is paying you to find the bugs.

GORDON: Wrong. If I write code for him, I have to bill him. Tracking down a bug is part of our services.

WEIZENBAUM: Then the economics is all wrong.

GREENWALD: But there are thousands of such cases.

GORDON: That's right; we're talking about the real world, not the world of academicians. This real world acts as a brake to some extent on developments.

POLAND: It acts as a brake on some kinds of development, but as an accelerator on other kinds. They are exploring ways of getting things done in a large applications area, and this will be a growing trend. I can see a growing use of personal computing facilities, either stand-alone or hooked up to larger machines, which work in a mystical manner, but a manner that is completely open as far as the user is concerned.

WEIZENBAUM: Should our function be to predict the future, or should we try to categorize how things are, how they are going, and how they could be better? If you tell me about the "real world" and how academia is different (i.e., a dream world) and that certain things must be, then I'm very unhappy. Are we senior citizens, or old fogies, or elder statesmen? Shouldn't we focus on how things ought to be done, rather than how they will inevitably be done?

GORDON: Joe, I mean no offense to those of you in the academic world. But your are a small, and relatively unusual segment of our industry. The rest of us cannot follow quite as quickly as you may be able to lead.

GRUENBERGER: If we were a consulting group with dictatorial powers, what could we agree on? Could we agree, for example, that all programs should be structured?

POLAND: No, I wouldn't agree.

McCRACKEN: The industry keeps expanding and hence the number of programmers keeps increasing. There must be some natural upper bounds, like the IQ of programmers. We're rushing into new applications steadily. The new users of the micro machines insist on their right to make every mistake all over again; they listen, but they don't believe it. People are proposing process control applications running with an Altair, and they're not even scared by it. I think they would listen to us, if we could get to them, but there are a lot more of them every day.

WHITE: They see such a difference between the machines they're working on and the ones we talk about, and they can't see the history that got us to where we are.

GRUENBERGER: Five years ago you could have set an upper bound on the number of pocket calculators that could be sold in this country, and you would have been wrong by a factor of 10,000. There doesn't seem to be an upper bound today. I doubt that there's an upper bound on micro computers, either.

McCRACKEN: The hardware costs are coming down so fast, that people are ignoring the software costs. They see the $495 cost for a CPU (for which they can own a computer) and when you ask them about software they say "No problem; I'll do that on a weekend." A year later, they may find that they've spent 12 times as much on software as on all their hardware (which also went up somewhat), but in the meantime they've committed their whole company's future--and that's spreading like crabgrass.

POLAND: A pleasant thought is that, with any luck, computing services will become available at the end-user level that will drive out that kind of nonsense. They may come about either via minis with 370-type software (done right), or with applications software, prepackaged, that can be used by a non-computer-specialist. In the year 2000 there will still be a large group of people, just as noisy as they are now, who will be trying to do something that you can't do easily and in a straightforward manner with the facilities that are then available. We ought to be working on facilities for all these groups, so that computers could be sold much as we now sell $18 calculators; that is, usable and understandable by non-experts.

GORDON: Our problems today do not come from the actual computation; that could be done by hand calculators or by small personal computers. Today's problems come from making data bases accessible to many people simultaneously, leading to the problems of managing, as in traffic management. You then have to consider data integrity, and security.

BEMER: To support Clarence's point, please note that I've ordered a 5100.

POLAND: Yes, and if five of your buddies also get 5100's and then you get together and say "Now let's share some data"--at that time you'll be hoist on your own petard. As soon as you try to swap cassettes, you're in trouble.

GORDON: That's my point; the problems of the future lie with data management.

BEMER: That's why I want content-addressable storage, because I don't think other methods are feasible.

GREENWALD: One big problem of the future will be the management of verbally-entered data.

GORDON: It will be a complex problem, but much on the same level as that of decimal entry to a binary machine.

WEIZENBAUM: I have a terminal whose control element is a microprocessor, and that microprocessor is buggy. If you hit the keys in certain sequences, you get strange actions. If a lot of people use those terminals, they are going to produce buggy programs. So even simple problems are getting screwed up.

GORDON: I never claimed that it was impossible to screw up simple things.

WEIZENBAUM: It is not only not impossible; it is the most probable thing. It is a problem of modern society; the quality of work has gone 'way down. I have here a marvelous watch that should be cleaned periodically, but I don't dare try; the preventive maintenance would kill it. Little simple jobs are not simple in the sense that people will do them correctly; they won't. And there is no reason to expect that they will do them correctly in the future.

POLAND: There is one reason: sheer sales economics.

WEIZENBAUM: I don't agree. Small problems may be simple, in the sense that people should know how to do them correctly, but the fact remains that they won't be. Now let me go to the other point: verbal communication with machines. Except in very restricted domains, that is a long way off; it is an astronomical problem. The problems we face in the future range from those of incompetent technicians (who don't know they're incompetent), to the problems of shared data bases, data management, and communications. If we were to adjourn right now, we would end on a very gloomy note.

GREENWALD: On the flight to New York, I was listening to the conversations between our pilot and the ATC. I contrasted all our technical gadgets for flying with what I was hearing, and it's scary. At one point, the ATC said "22 right" when he meant "22 left." Remarks like "Can you see United 12?" shake you up. But for all that (in light of our discussions here), I'd still rather have humans doing the job than trusting to a computer program.

ARMER: There was some concern expressed about the new users of microprocessors making lots of mistakes. Is that a real social problem? There can be damage done, but will it be serious? Won't the unfit simply not survive? They won't be trying to control nuclear reactors. For most applications, the scale will be small enough, due to the size of the equipment involved, to produce only local bad effects.

McCRACKEN: It depends. We have cases where computer errors have resulted in people getting killed. That may be repeated. It seems to me that there is more riding today on computer decisions than there ever has been on other technological divices.

WEIZENBAUM: We can deplore the sad state of workmanship among watch repairman, and wash machine repairmen, and auto mechanics, but we here represent, in a sense, the computing profession, and we have an obligation to bring craftsmanship back within our industry. We should do what we can to influence people to be able to say "I don't understand this sufficiently to produce it and put my name on it." Part of that responsibility, of course, is mine as a teacher; I should foster that sense of responsibility among my students.

BEMER: I have adopted Herb Grosch's suggestion and I sign every program I write.

ARMER: What do you do with a program like OS--put Carl Reynold's name at the top?

WHITE: You could have every module of the system have the programmer's name on it as part of its COMMENTs.

McCRACKEN: But who is responsible when something is misused? Suppose someone misuses Joe's ELIZA program (as they have done)--is Joe responsible?

WEIZENBAUM: Actually, I considered not publishing ELIZA for just that reason; perhaps I shouldn't have. But at least when it was published, it was in a debunking mode, and I made it clear how it should be used.

GRUENBERGER: Yes, and a book just appeared that reprinted just one page of your article--totally out of context--and I hold that that's irresponsible publishing, because no one will go back to read your article in its entirety.

GORDON: There seems to be a feeling here that some segments of our industry are demonstrating the morality of the buffalo; namely, let the chips fall where they may.

McCRACKEN: The designers of the original microprocessor chip had no idea that anyone was going to build general purpose computers around it, so they could not have anticipated any of the troubles we have referred to.

WHITE: That's also true of the first computer, and we see history repeating itself.

GREENWALD: But the chip is several orders of magnitude cheaper than that was, and so its effects will be not only different, but more extensive.

McCRACKEN: The same applies to the LISP processor; it will be faster and less expensive, too. It will have effects that cannot be predicted.

GRUENBERGER: Let me go back to a point of disagreement. I said earlier that I thought we could agree that eventually all programs would have to be structured, and we didn't agree. I'd like to hear why not.

POLAND: The word is "programs." We will not be doing things by writing programs in the sense that we do it today. We will use non-procedural processes. For such processes (e.g., non-procedural conversations with a computing system) the whole concept of structured programming is irrelevant. The things we call programs will be produced, on the whole, by vendors and those will probably be structured. In the same vein, I believe that chips will be produced by design automation, and not by any process that puts pen to paper.

GRUENBERGER: Let me reword it, to find something we might agree on. "Sets of instructions for computers, if they are spaghetti-like, full of GOTOs, and a mishmash, shall be outlawed."

GREENWALD: But there will be languages (assembly, for example) that do not allow the constructs of structured programming, so it can't be.

WEIZENBAUM: Let me try it. We have verbal and written communication and some standards of literacy, so that there is a generally accepted level below which we can say "That won't do." Could we not have a minimum level of literacy in computer programs? This is independent of the fact that the programs may work. Maybe it should be called elegance, and the level could be low, but shouldn't there be <u>some</u> minimum level?

GREENWALD: To my way of thinking, you're now differentiating between structured programming and structured design. There is no reason why a program that is implemented in assembly language couldn't be written in structured language.

GORDON: There is no reason why hardware, within the next 5 years, cannot accept structured codes directly.

GRUENBERGER: Perhaps an example of Joe's literacy level might be the common student error of writing

```
LOAD ACCUMULATOR
LOAD ACCUMULATOR
```

in succession--that would be below the level of program literacy. We could find much better examples, I'm sure.

GORDON: But who decides and enforces this level? When the year-end report has to get out, is someone supposed to say "We can't run this because it's aesthetically displeasing?"

GRUENBERGER: Well, you do just that with the written report that goes with it; you don't let illiteracy creep in there.

WEIZENBAUM: In general, in business, a man being considered for promotion who is asked to give a talk to his peers or to customers, who turns out to be functionally illiterate, will probably be denied the promotion.

*: Not at IBM.

WEIZENBAUM: I'm not talking about an occasional lapse in grammar or misuse of words, but functional illiteracy. Now that computer languages have become so much a part of our means of general communication, I think we should work toward establishing mores of conduct in those languages as we have in English.

GRUENBERGER: The guy who writes two LDAs in a row won't be working on that year-end report because he won't get that far.

GREENWALD: Nonsense; you won't even know it.

GORDON: I agree. No one communicates between people with programs. You write one and you give it to the machine to read.

WEIZENBAUM: A program written by and for one man might have that property. But programs are read by other than their authors. It is becoming common to have teams read each other's programs. I suggest that programs will become a medium of communication between people.

GREENWALD: If you want to promote literacy, you should require that every program have a structured walk-through.

POLAND: Whenever you try to write a treaty or a contract or a law in two languages, you have ambiguity and misunderstanding.

Similarly, when we say the same thing in two languages in our field (such as a compiler and a reference manual for it) we have that problem.

WEIZENBAUM: Any legal document must have a certain standard of elegance, or correctitude, in order to be accepted.

GREENWALD: We had a rule at RAND that every line of code had to be signed off by two people. In a structured walk-through, essentially every line of code is signed by a team of people.

POLAND: In construction work on public buildings, every blueprint is signed by at least two people. But if I design a wood frame to grow tomatoes on, is it reasonable to demand that someone read and approve my design?

WEIZENBAUM: It's a question of the resulting interfaces. By your own argument, you could demand freedom to sink a tank in your back yard to store gasoline. Your tomato frame affects only you, but your tank may interface with your neighbors, and your township will certainly insist on their right to approve your plan. The analogy with programs is quite clear.

When someone buys a microprocessor to use to turn lights and coffee makers on and off, he affects no one else; that's a tomato frame. But when a microprocessor is sold to go into some other device, the situation is more like the gasoline tank: then there should be certain minimum standards including that the inspector is able to read the code. That certainly isn't the way it is, but perhaps that's the way it ought to be.

GORDON: That's a great idea, but it's going to be hard to sell. At one time programs were reviewed systematically because the machines were expensive and people were relatively cheap. Now it's reversed; people are expensive and the machines are so cheap that it's usually easier to run the code and see if it works. If the universities will successfully promote your idea of program literacy, I'm all for it. Maybe they'll come into industry and spread the word that that's the way it's done.

GRUENBERGER: As one of those in the universities, let me say that I frequently hesitate when it comes to giving an "A" in a computing course. I am thinking at that time of the effect this person might have in industry with my blessing.

GREENWALD: I think that computer management is illiterate in computing. I tried to push the idea in our shop of ego-less programming (everyone reads every one else's code) and got good support from our recent college graduates. They had all read Weinberg's book. When I read that book myself I was surprised to find that we had practiced ego-less programming for years at RAND. One thing that it does for you, besides getting better code, is instill a spirit of "We are a team trying to get this project done" rather than "I am a coder and this is my own little domain." It encourages people to say "Hey, will you check this program for me?--there's something wrong with it." I think the trouble lies with programming management, due to either being not knowledgable in computing, and/or gutless, and/or "That's a good idea, but I have to get this program running today."

WHITE: I really believe that will cure itself with time.

GREENWALD: But these people are going higher and higher in management.

POLAND: But I think that the solutions we are considering are overly theoretical and counter to "common sense," and the chances of getting them implemented are low. I think there are alternatives, such as the constraints of structured programming, or such simple things as making the compiler so that it will not compile a variable address of a GOTO. Those things we can do. But it's just a pious hope to think we can spread the idea of having all programs reviewed.

GREENWALD: I don't think we can do the former. For example, in our shop, we have a subset of a language for use in our internal systems development. We could offer that sub-set to our customers, but inside of a year they'd be demanding the full set.

BEMER: "He who has control of the compilers and assemblers is really the dictator of the installation."

GORDON: One of programming management's most important functions is to inhibit the natural creativity of programmers.

GRUENBERGER: Barry, where did you get the idea that code was reviewed in the early days? I never heard of it.

GORDON: There were times; in fact, consider the days when people wrote code before they got their machine. We called it "desk checking," when people were cheap compared to machines. We were doing something right but perhaps for the wrong reason.

GREENWALD: Up to 1960 at RAND, anyone on my team had had his code reviewed by someone else.

GRUENBERGER: I suspect that those policies were not dependent on the scarceness of the machine, or relative costs. When you didn't have a machine, you checked code because there was little else to do, and when you had a machine you used it because it was there. In those days (when computing was the greatest game ever invented), we seldom stopped to analyze whether we did anything sensibly--the only goal was to get programs to fly. It is only in the last 5 years that sober minds have actively sought the best way to do things. And this is an optimistic note for the future.

BEMER: I agree that the proper policy cannot arbitrarily be dictated by what is available and what happens to be cheap. Some things will always be done better with the exercise of human judgement, and some things should always have been done by, for example, error checking routines in the compilers.

GRUENBERGER: No matter how cheap diagnostic programs get, I will always want the ultimate diagnosis of my tummy ache to be done by a living doctor.

GREENWALD: One place where we're being hurt is in having management ignore programmer backup. People can leave a project, or die, or quit--and that fact should always be considered, so that no one becomes a key person to the project. But this principle is being totally ignored. So is ego-less programming. We still tolerate having programmers hide in their little corners, not letting anyone see what they're doing. These are things that we talk about, and may even agree on, but they are difficult to impose on any group.

GRUENBERGER: But those places where their life depends on doing it right, do it right.

GREENWALD: But what is right is usually a function of the environment; there are few absolutes. Some shops can operate for the long run, while others, for many good reasons, have to operate for the short run. If a business will fold unless a given calculation is done, then the correct answers in the shortest elapsed time certainly takes precedence over elegant code. Now, when you add in to that the moral atmosphere we seem to have in business and society, it becomes increasingly difficult to define what is "right."

McCRACKEN: At least I can do what appears to me as right, dammit.

GRUENBERGER: That's the message that Joe has been pushing for years: that there is tremendous power in a good example.

FOR YOUR ADDED ENTERTAINMENT

ESPRIT de COMPUTING '76

✻ OFFERS ✻

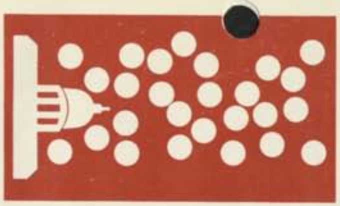
1. **WILSON ST. EAST DINNER PLAYHOUSE:** Wednesday evenings banquet promises to be another unique feature of Esprit de Computing '76. You can select a superb entree from the menu and highlight your dinner with wine . . .then sit back and be entertained by a professional production of SLEUTH, a brilliant suspence thriller. (Included with Full-Conference and Spouse Registrations.)

2. **TOUR OF THE CITY OF MADISON:** A visit in, around and all about Madison. The downtown dominated by the imposing State Capitol building, rising from its tree-lined square. Through the adjacent University of Wisconsin campus with two stops at special points of interest. Into several of the city's beautiful parks, many curving along three surrounding lakes. A drive into historic and beautiful residential areas and through the arboretum, the University's outdoor natural botanical laboratory. (Wed. Apr. 28-10 a.m.; Cost: $4.00)

3. **TOUR OF HOUSE ON THE ROCK and CAVE OF THE MOUNDS:** The House on The Rock, located 40 miles west of Madison is recommended by the AAA as a place of exceptional interest. The House on The Rock is where the unusual is commonplace. You'll find no commercial distraction from the many forms of natural and man-made beauty here. Beginning with a drive along the parklike approach, until the final beat of some mechanical musical marvel, every second of your tour is crammed with sights, sounds and textures artfully calculated to please every sense.

   The Cave of The Mounds is indeed one of Wisconsin's finest natural wonders. Located near Blue Mounds, Wisconsin, the Cave of The Mounds offers natural splendor reminiscent of Mammouth Cave or Carlsbad Caverns.

   This tour will include lunch at the Spring Green, a gourmet restaurant located in the heart of Frank Lloyd Wright country on the beautiful Wisconsin River. (Thurs. Apr. 29-9 a.m.-5 p.m.; Cost: $18.00)



ESPRIT de COMPUTING '76
210 N. Yellowstone Drive
Madison, Wisconsin 53705

Southern Wisconsin Chapter
of
D.P.M.A.
presents

ESPRIT de COMPUTING '76

APRIL 27, 28 & 29, 1976

Sheraton Inn - Madison Gateway
706 John Nolan Drive
Madison, Wisconsin

GENERAL SESSIONS:

Title: COMPUTER SECURITY
Speaker: ROBERT H. COURTNEY, JR.

No two data processing systems need precisely the same array of security measures. Variations in the physical environments, system configurations, data sensitivities, system applications and numerous other factors all contribute to the differences in their security needs. Mr. Courtney's presentation provides a rationale and an orderly, systematic procedure for the selection of security measures on the basis of cost/effectiveness relationships.

The material presented by Mr. Courtney includes discussions of problem definition, fixing responsibility, the valuation of data, threat analysis, and, finally, the selection of security measures. Included are brief discussions of identification schemes, authorization, audit, security measures in hardware and programs, physical security and operating procedures, eavesdropping and wiretapping.

Robert H. Courtney, Jr. is IBM's Manager of Data Security and Privacy. He is responsible for establishing architecture and design criteria for data security in IBM's hardware and software and assuring their incorporation into these products.

He joined IBM in 1960 as Manager, Intelligence Systems Department, in the Federal Systems Division in Washington. He later went to IBM's Kingston, New York, facility as Manager of Displays and Graphics Development. In this capacity he managed the organization which introduced display and graphics devices into the IBM product line.

He is a native of Virginia and a graduate of Virginia Tech. (VPI) in Electrical Engineering. He is a member of numerous federal, state and industry committees addressing the security and privacy issues.

His work is published in the proceedings of numerous conferences and symposia, IBM publications and in the Computer Law Service.

Title: LEGAL ISSUES RE EDP HARDWARE/SOFTWARE CONTRACTING
Speakers: SUSAN H. NYCUM AND TERRY D. MILLER

Contracting the computer products and services requires a blend of skills and talents. Ms. Nycum and Mr. Miller will provide strong assistance to Esprit de Computing '76 attendees by focusing upon legal issues pertaining to EDP hardware and software contracting. Ms. Nycum will discuss the approach to negotiation and the contents of typical computer contracts from the DP Manager's point of view. Mr. Miller will highlight such areas as determining potential contractor responsibility, evaluating procurement bids, and interpreting government guidelines for organizations implementing government contracts.

A graduate of Ohio Wesleyan University and Duquesne University Law School, Ms. Nycum is associated with the San Francisco law firm of Chickering & Gregory, where she specializes in the legal problems of computers. She is also the principal legal consultant to Stanford Research Institute's study of Computer Abuse for the National Science Foundation.

Prior to practicing in California, Ms. Nycum was the director of the Stanford Campus Computer Facility and Manager of User Services and Operations at Carnegie-Mellon University Computer Center. She also served as Research Associate and Law and Computer Fellow at Stanford Law School. Ms. Nycum entered the computer law field in 1965 as a Division Head of the Health Law Center-Aspen Systems Corporation, where she was responsible for research and development and production associated with many of Aspen's products including the development of litigation support systems, creation of computerized state statute data bases, and specialized computer assisted research for clients.

Ms. Nycum is currently a council member of the ABA Section on Science and Technology, Director of the Computer Law Association, and Chairman of the Standing Committee on Legal Issues of the ACM. Susan Nycum is a member of the Bar of California, Pennsylvania, and the United States Supreme Court. She has published numerous computer law articles and has recently co-authored with Robert P. Bigelow the book: **Your Computer and the Law,** which was released in January.

Terry D. Miller is President of Government Sales Consultants, Inc. a firm engaged in assisting federal agencies and ADP vendors with all aspects of procurement of ADP equipment, software and services. Prior to forming GSCI, Mr. Miller was employed as a computer equipment analyst and procurement analyst. He also spent eight years with FCC as a computer equipment analyst, systems analyst, programmer and Chief of the Computer Operations Branch.

Mr. Miller is the author of Federal ADP Procurement, a book designed to assist buyer and seller with understanding the procurement system. He also authors monthly columns on ADP Procurement in Modern Data and Mini-Micro Systems and other publications. The subject of his talk will center upon legal issues pertaining to EDP hardware contracting and will center on a discussion of ADP contracting and contract types. This session should prove to be invaluable to the DP professional responsible for hardware decisions.

Title: STAGES OF EDP GROWTH
Speaker: ROBERT W. BEMER

The past nature of EDP growth has reflected only one dimension–relative size and power. Forthcoming stages will reflect a shrinking in EDP hardware, and perhaps, even software. To enable the audience to gain insight into planning for the future, Mr. Bemer will categorize several other dimensions of EDP growth and change such as software, data base, networking, personalizing, and non-expert usage.

Robert W. Bemer is currently Consulting Engineer, User Environment to the Director, Advanced Systems Engineering, Honeywell Information Systems. His computing career began at the RAND Corporation in 1949. After two years at Lockheed Aircraft Co., he organized the computing departments at Marquardt Aircraft and the Lockheed Missiles and Space Company. He joined the staff of the IBM Corp. in 1955 and held several programming managerial positions. In 1962, he joined UNIVAC as Director of Systems Programming. During 1965, he spent a year at Bull-GE as Consultant to the General Manager, and was then assigned as Consultant in Phoenix. He has been involved in the standardization of computer languages since 1960, and is presently Chairman of the International Standards Organization Subcommittee on Programming Languages. He was a primary developer of ASCII and has authored over 50 papers. He is a Fellow of the British Computer Society and a member of ACM and DPMA.

Title: SELF-FULFILLING PROPHECIES
Speaker: DR. MICHAEL A. JOHNSON

'Increased productivity' has become a slogan among today's managers. One approach which has been advocated to improve productivity has been the development of a positive mental attitude within employees. Recent research, however, suggest that an employee's positive mental attitude can only be sustained if the supervisor also develops and maintains a positive work environment where results are achieved and organizational and personal needs are satisfied. Dr. Johnson will describe the four factors–climate, feedback, constructive assistance with inputs, and objective evaluation of outputs–which are essential for a manager to instill positive job expectations.

Michael A. Johnson is a Professor of Organizational Management at the University of Minnesota and a Visiting Professor in the University of Michigan, Graduate School of Business. He served as the Director of Management Planning and Development for the Minnesota Department of Highways from 1970 to 1975.

Professor Johnson completed his studies in Psychology at the University of Minnesota and received his Juris Doctor from the William Mitchell College of Law in St. Paul, Minnesota.

Professor Johnson has conducted numerous seminars in Management and Communication throughout the country for such organizations as IBM, the State of Minnesota, and the U. S. Civil Service Commission. He is the co-originator of the Accountability Management Planning System (AMPS), a dynamic new management system that is being implemented throughout the United States.

LUNCHEON ADDRESSES:

Title: POSSIBLE FUTURES . . . HARDWARE/SOFTWARE
Speaker: CAPT. GRACE HOPPER

In an environment of accelerating change, it is essential that everyone connected with computers be aware of the systems that will develop in both the immediate and less-immediate future. It is the task of management to plan, to select personnel, to decide and to act, today, in the light of future development both defined and possible. Capt. Hopper will discuss both the defined and possible, which should be of real benefit to the DP professional making decisions today that will affect tomorrow's environment.

From 1949 to 1971, Dr. Hopper was associated with the UNIVAC Division of the Sperry-Rand Corp. in such capacities as Senior Mathematician, Senior Programmer, Director of Automatic Programming and Staff Scientist in Systems Programmings.

Dr. Hopper has published over 50 papers and articles on automatic programming, and has been the recipient of numerous awards in her field. She was selected as the Data Processing Management Association's first Computer Sciences "Man-of-the-Year" in 1969. In 1971, Sperry-Rand's UNIVAC Division initiated the Grace Murray Hopper Award for young computer personnel to be awarded annually by the Association for Computing Machinery. She is a Fellow of the Institute of Electrical and Electronics Engineers, the American Association for the Advancement of Science, the Association of Computer Programmers and Analysts, and the CODASYL Executive Committee.

Title: DP MANAGEMENT RESPONSIBILITIES RE: PRIVACY AND COMPUTER ABUSE
Speaker: DONN B. PARKER

Privacy legislation is establishing constraints on the information processing function. Mr. Parker will evaluate this legislation in relation to its impact on data processing, and delineate the DP Manager's responsibilities regarding this issue. In addition, Donn Parker will stress that optimizing the DP organization's safeguards solely to meet legislative requirements may actually result in the sub-optimization of the over-all safety of the installation. He will describe the role computers play in crime and present a profile of perpetrators of computer abuse acts. He also will share his predictions about computer related crime in the future.

Donn B. Parker is Senior Information Processing Specialist at the Stanford Research Institute, where he recently completed a National Science Foundation sponsored study on computer abuse. In addition to his six years with the SRI, he was employed twelve years with General Dynamics in San Diego and eight years with Control Data Corp. in Palo Alto, CA. He has a M.S. in Math from the University of California, Berkley. In June, at NCC '76 in New York, Mr. Parker will present two papers entitled: "Computer Abuse Assessment," and "Computer Abuse Perpetrators and Vulnerabilities of Computer Systems." His forthcoming book on "Crime by Computer" also will be published in June. Actively affiliated with ACM, he has served ACM as a Council Member-at-Large, and National Secretary. Currently, he is serving as Chairman of the AFIPS Professional Standards and Practices Committee.

SEMINAR SESSIONS:

Title: MANAGEMENT DEVELOPMENT
Speaker: DR. DAVID P. NORTON

The allocation of EDP resources and the alignment of these resources with organizational goals over the long-term has been one of the most complex of organizational processes. The lack of a consistent structure to relate requirements and resources has compounded this dilemma. To address this problem, Dr. Norton will conduct this management seminar as follows:

Long-Range Planning: This session will discuss several alternative approaches to the long-range planning process and develop a structure for linking organization requirements, technological change and the DP organization's resources to a set of planning guidelines and performance benchmarks.

Organizational Alternatives: This session will discuss the major forces affecting the issues of centralization and decentralization (e.g., the mini-computer revolution) and provide a structure for sorting these factors and for dealing with constraints in the phasing of organizational change.

Management Control: Performance measurement is the foundation of any mature approach to management. Such measurement must take place at the highest levels of the DP organization as well as the lowest. This session will discuss a "management by objectives" approach to goal setting and performance measurement which encompasses and links each level of the DP organization.

Dr. Norton is a management consultant specializing in the management of EDP in large organizations and is President of D.P. Management Corporation. Dr. Norton has been a Systems Consultant with the RCA Corp. and a staff Consultant with Index Systems. Dr. Norton received his Ph.D. from Harvard Business School, in the area of management information systems and has written many articles, including co-authoring "The Stages of EDP Growth," HARVARD BUSINESS REVIEW, 1974.

Title: BEHAVIORAL STUDIES
Speaker: DR. JOHN G. GEIER

Recent psychological research tends to support the following statements: 1) you cannot motivate other people; 2) individuals do things for their reasons, not for ours; 3) all people are motivated. To apply the impact of these statements to the management field, the manager needs answers to the following questions: 1) How do I go about getting things done? 2) In what type of situation do I tend to lead most effectively? 3) Am I more comfortable when dealing with a technical or human emotion problem? 4) What is my individual style of interacting with people? 5) While leading, how well do I approach the motivation/satisfies of people? 6) How well I tend to react when faced with stron opposition? 7) How do I conduct myself as a member of a group? 8) What additional assistance do I need to develop more effective skills?

The instrument, "The Personal Profile," will be administered in the session in order that each participant may receive an analysis which will include the answers to these questions. From this base, the participants will be able to determine the specific individual strengths of others and how to build upon these strengths for greater harmony and organizational productivity.

Dr. John G. Geier, Ph.D., is presently Director of Behavioral Science, Health Ecology Division, University of Minnesota. His social psychological research and application to work in task oriented groups is nationally recognized. He has conducted numerous training programs for business and industry. Participants include General Motors, IBM, AT&T, Sears Roebuck, and many other business and professional organizations. He directs clinics for top executives at various areas in the country.

He is author of articles and manuals and books related to manpower and career planning, managerial motivation, stress and health in controlling behavior. Use of the "instrumented approach" in his seminars has resulted in rave comments as "the best I have ever attended". Others have said, "the approach has positively changed our way of working and developing team concepts."

Title: PROJECT MANAGEMENT
Speakers: JEROME F. ROLEFSON and ALLEN B. RUSSELL

Project management has proven to be an effective tool to help handle the problems of completing systems and programming projects on schedule, within budget and with the desired results. This seminar will provide an in depth look at project management - what it is, does and requires. It will also outline procedures, forms and organizational considerations necessary to utilize project management in your company or department. The seminar will conclude with an in-depth look at the features, operations and cost of a working project management system.

Jerome F. Rolefson is Assistant Professor of Management, Department of Business and Management, U. Wisconsin-Ext. Milwaukee.

Professor Rolefson received his B.S. degree in Electrical Engineering and his M.B.A. in Management, both from the University of Wisconsin. He holds the Certificate of Data Processing. He is a member of DPMA and the Association for Systems Management.

Allen B. Russell has been the manager of Data Processing at Madison since August 1969. He was responsible for converting the City's data processing goals into a formal long-range plan and publishing standards which govern all data processing functions.

Mr. Russell began his data processing career with the U. S. Army in 1955. He was director of Data Processing for the Ordnance Corps in Europe from 1961 through June, 1963.

Mr. Russell is currently the President of the Southern Wisconsin Chapter of DPMA and is a member of the Data Processing Advisory Committee - Area Board of Vocational Technical & Adult Education District 4.

Title: MINI-COMPUTERS
Speakers: DON GROSS, MARIJANE McDONOUGH and JIM KROK

Mini-computers have invaded the data processing industry with tremendous assets in the constant battle to improve the price/performance ratio. A very strong trend is underway, as organizations of all types and sizes learn that the mini-computer can take on a substantial part of the data processing burden. The trend is worth examining . . . which is what the members of Mini-Computer Systems, Inc. will help Esprit de Computing '76 attendees do. The three seminar sessions will address mini-computer selection criteria, operating systems and file design, and user systems design-application concerns. Everyone associated with data processing and also, management not directly involved in data processing should find this seminar beneficial.

Donald L. Gross is President and a Director of Mini-Computer Systems, Inc. Previously, he was Systems Sales Manager of Olivetti Corp., where he was employed for nine years. Mr. Gross has been active in activities including service as President of the Data General Corp. users group.

Marijane McDonough is V-Pres. MCS and Director of Systems. Ms. McDonough has been responsible for the systems analysis sector of the company's operations and has also been active in customer training. Prior to joining the corporation in 1971, she was a systems engineer at IBM.

James C. Krok is Manager of Systems Software of Mini-Computer Systems, Inc. Mr. Krok joined the company in 1972 and has been instrumental in development of the proprietary "Micos" mini-computer system. He has been in the DP field for 15 years, including service with Standard Oil (New Jersey) and began working with minis in 1964.

Title: DATA BASE
Speaker: LEO J. COHEN

The data base system project represents a new departure in the development of computing systems for the DP industry. This is because the data base system project has no identifiable middle and, usually, no end. This means that other terms must be found for identifying project progress, and results in a descriptive structure for the project itself.

In his overview, Mr. Cohen will discuss: data base systems, as opposed to file-oriented systems, a description of the seven major steps to be accomplished in the data base system project, and the general technical outlines of a data base management system.

The second part of this presentation will deal with data base systems that are available for support. These packages divide into two categories known as "designer" and "end user." The differentiation will be discussed and an analysis of the packages in terms of best DP use.

The third section of the presentation is concerned with a comparative analysis of the packages, which necessarily includes a discussion of the problem of data base administration. This is a subject whose complexity is strongly dependent on the particular package and the environment for its application to data base systems within the organization.

Leo Cohen, President, Performance Development Corporation is a noted author in operating system design, data base systems and systems performance measurement. He has been active in the computing industry for over twenty years and has hardware experience covering the equipment of every major vendor. Mr. Cohen has designed a wide variety of software systems including compilers, operating systems, simulators, and data base management systems.

Title: DATA COMMUNICATIONS
Speaker: DR. DIXON R. DOLL

Data communications services, applications, and networking techniques are reaching levels of maturity and cost-effectiveness that impact all types of data processing installations. The seminar reviews current offerings in the services area such as digital networks, value-added cariers, satellite carriers, and conventional offerings whose new pricing structures are constantly being revised. Also, the roles of intelligent terminals and recently announced line protocols such as SDLC will be candidly evaluated. Finally, the data processing manager will be encouraged to adopt an aggressive policy of acquiring self-sufficiency in design tools for resolving the numerous network optimization tradeoffs which must constantly be evaluated. A discussion of such tools will conclude the presentation.

Dixon R. Doll is the President of DMW Telecommunications Corporation, a system engineering firm serving the computer and telecommunications industries. Dr. Doll is also a founder and technical Director of the International Communications Corporation's ICC Institute in Miami.

Dr. Doll has worked with IBM, Raytheon, and Graphic Scanning Corp. on a broad array of projects. These included the development of programs for computer-aided network optimization, and the development of technical specifications for an on-line message switching system. Dr. Doll is a principal architect of the communications network configurator, a family of computer programs used to design and analyze end-user computer-communication networks.

Title: STRUCTURED PROGRAMMING
Speaker: CLARK W. CATELAIN

The continual increase in the size and complexity of computer programming projects has resulted in even greater increases in the cost of building and maintaining a system of programs. Concurrently, program reliability has decreased. This presentation will survey currently popular techniques for designing and implementing programs and will identify methods for managing projects which utilize these techniques.

Topics will include structured programming, proof of correctness, abstraction, modularity, structured flow diagrams, HIPO, walkthroughs and consequences for COBOL.

Clark W. Catelain is durrently the Program Product Manager, Western Application Development Center, Burroughs Corporation. Mr. Catelain is responsible for managing the development of software products for Burroughs. He has been involved in the design, implementation and management of various data base systems, COBOL standards work relating to structured programming and in adapting new programming techniques.

Mr. Catelain received a B.S. in Comp. Sci. from Purdue U. and did graduate work at Wayne State. U.

ESPRIT de COMPUTING '76

A UNIQUE EDUCATIONAL OPPORTUNITY

☑ Tired of reading confusing technical journals?

☑ Looking for an in-depth discussion of data base . . . structured programming . . . data communications . . . or mini-computers?

☑ Want to improve your management capabilities through a seminar on **behavioral studies** . . . management development . . . or project management?

☑ If you answered "YES" to any of the above . . .

ESPRIT de COMPUTING '76

offers YOU a unique educational opportunity

Esprit de Computing '76, sponsored by the Southern Wisconsin Chapter of DPMA, promises to provide an innovative, professional growth-oriented experience for data processing professionals of all realms. This conference will focus its outstanding educational curriculum around a slate of speakers **unparalled on any single conference agenda.** General sessions will highlight and provide an overview on: Computer Security Issues, Legal Issues Relating to Hardware and Software Contracting, Examining the Stages of EDP Growth, DP Management Responsibilities Regarding Privacy, Self-Fullfilling Prophecies . . . Job Expectations, and Possible Futures . . . Hardware and Software.

But, that is only the beginning! The most unique aspect of Esprit de Computing '76 is SEVEN IN-DEPTH EDUCATIONAL SEMINARS. During two days of the conference, each participant may select an intensive three-part seminar. Each seminar is taught by a leader in the respective field and, to insure the optimum educational opportunity, each seminar will be limited to 50 participants on a first come basis. The seminar topics and their corresponding instructors and sub-topics are outlined as follows.

SEMINAR TOPIC	INSTRUCTOR	SESSION ONE	SESSION TWO	SESSION THREE
STRUCTURED PROGRAMMING	CLARK CATELAIN, PRODUCT MANAGER, BURROUGHS CORP.	DESIGNING WITH RELIABILITY AS A GOAL	IMPLEMENTING RELIABLE PROGRAMS	RELATED MANAGEMENT TECHNIQUES
DATA BASE	LEO COHEN, PRESIDENT, PERFORMANCE	OVERVIEW	APPLICATION CONSIDERATIONS	A COMPARATIVE ANALYSIS OF DATA BASE SOFTWARE
DATA COMMUNICATIONS	DIXON DOLL, PRESIDENT, TELECOMMUNICATIONS INC.	OVERVIEW	APPLICATIONS CONCERNS	A COMPARATIVE ANALYSIS OF DATA COMMUNICATIONS SOFTWARE
BEHAVIORAL STUDIES	DR. JOHN GEIER UNIV. OF MINNESOTA	EVALUATING YOUR LEADERSHIP STYLE	INTERPRETING YOUR WORK BEHAVIORAL CHARACTERISTICS	IMPROVING YOUR INTER-PERSONAL RELATIONSHIPS
MINI-COMPUTERS	DONALD GROSS, PRESIDENT, MINI-COMPUTER SYSTEMS, INC.	SELECTION CRITERIA	OPERATING SYSTEMS & FILE DESIGN	USER SYSTEMS DESIGN
MANAGEMENT DEVELOPMENT	DR. DAVID NORTON, PRESIDENT, D.P. MGMT. CORP.	LONG-RANGE PLANNING	EDP ORGANIZATIONAL ALTERNATIVES	MANAGEMENT CONTROL OF THE D.P. FUNCTION
PROJECT MANAGEMENT	JEROME ROLEFSON UW-MILWAUKEE and AL RUSSELL CITY OF MADISON	WHAT IT IS, DOES, AND REQUIRES	SETTING UP PROJECTS IN YOUR ORGANIZATION	PROJECT MANAGEMENT APPLIED

Esprit de Computing '76 does offer a unique educational opportunity; an opportunity tailored to individual needs and interest. You won't want to miss this one, so register **now** to insure your place in the sessions of your choice.

CALENDAR AT A GLANCE

TUESDAY--APRIL 27, 1976:

1:00 OPENING REMARKS--Richard F. Gehrt CDP, DPMA Region 5 Vice-President. Kathleen Wagner Micke CDP, DPMA Region 5 Conference General Chairperson. Allen B. Russell, President, DPMA Southern Wisconsin Chapter.

1:15 WELCOME--Honorable Paul R. Soglin, Mayor, City of Madison.

1:30 KEYNOTE PRESENTATION--'Computer Security: Disaster Avoidance and Recovery.' Robert H. Courtney, Manager, Data Security, IBM Corporation.

3:00 Coffee/Cola Break.

3:30 PANEL DISCUSSION--'Legal Issues Re: EDP Software/Hardware Contracting.' Attorney Susan H. Nycum, Chickering and Gregory. Terry D. Miller, President, Government Sales Consultants, Inc.

5:00 Conference Evaluation.

WEDNESDAY--APRIL 28, 1976

8:30 OPENING SESSION--'Stages of EDP Growth.' Robert W. Bemer, Honeywell Information Systems.

10:00 Coffee/Rolls.

10:30 SEMINAR SESSIONS--PART I:

A. Management Development.

B. Mini-Computers.

C. Data Base.

D. Data Communications.

E. Structured Programming.

12:00 LUNCHEON ADDRESS--'DP Management Responsibilities Re: Privacy and Computer Abuse.' Donn B. Parker, Information Specialist, Stanford Research Institute.

1:30 SEMINAR SESSIONS--PART II: (A, B, C, D, E.)

3:00 Coffee/Cola Break.

3:30 SEMINAR SESSIONS--PART III: (A, B, C, D, E.)

5:00 Conference Evaluation.

6:30 DINNER PLAYHOUSE--WILSON STREET EAST 'SLEUTH'--A Suspense Thriller.

THURSDAY--APRIL 29, 1976:

8:30 OPENING SESSION--'Job Expectations . . . Self-Fulfilling Prophecies.' Dr. Michael A. Johnson, University of Minnesota.

10:00 Coffee/Rolls.

10:30 SEMINAR SESSIONS--PART I:

A. Behavioral Studies.

B. Project Management.

C. Data Base.

D. Data Communications.

E. Structured Programming.

12:00 LUNCHEON ADDRESS--'Future Possibilities . . . Software/Hardware.' Capt. Grace M. Hopper, U.S. Dept. of Navy.

1:30 SEMINAR SESSIONS--PART II: (A, B, C, D, E.)

3:00 Coffee/Cola Break.

3:30 SEMINAR SESSIONS--PART III: (A, B, C, D, E.)

5:00 Conference Evaluation.

ESPRIT de COMPUTING '76 -- WEDNESDAY, APRIL 28, 1976

TIME	EVENT	CODE	SPEAKER	TOPIC	LOCATION
8:30 a.m.	Opening Session	--	Robert W. Bemer Honeywell Information Systems	Stages of EDP Growth	North Ballroom
10:00 a.m.	Coffee/Rolls	--			Mezzanine
10:30 a.m.	Seminar Sessions: Part I				
	Behavioral Studies	AW	Dr. Michael A. Johnson University of Minnesota	Evaluating Your Leadership Style	Room A
	Data Base	C8	Leo J. Cohen Performance Development Corp.	Overview	North Ballroom
	Management Development	A8	Dr. David Norton, Kenneth Rau D. P. Management Corporation	Long-Range Planning	Room B
	Mini-Computers	B8	Don Gross, Jim Krok, Marijane McDonough Mini-Computer Systems, Inc.	Selection Criteria	Room 106
	Project Management	BW	Jerome Rolefson, Al Russell UW-Milwaukee, City of Madison	What It Is, Does and Requires	Room C
	Structured Programming	E8	Clark W. Catelain Burroughs Corporation	Designing With Reliability As A Goal	Room D
12:00 Noon	Luncheon Address	--	Donn B. Parker Stanford Research Institute	DP Management Responsibilities Re: Privacy and Computer Abuse	South Ballroom
1:30 p.m.	Seminar Sessions: Part II				
	Behavioral Studies	AW	Dr. Michael A. Johnson University of Minnesota	Interpreting Your Work Behavioral Characteristics	Room A
	Data Base	C9	Leo J. Cohen Performance Development	Application Considerations	North Ballroom
	Management Development	A8	Dr. David Norton, Kenneth Rau D. P. Management Corporation	EDP Organizational Alternatives	Room B
	Mini-Computers	B8	Don Gross, Jim Krok, Marijane McDonough Mini-Computer Systems, Inc.	Operating Systems & File Design	Room 106
	Project Management	BW	Jerome Rolefson, Al Russell UW-Milwaukee, City of Madison	Setting Up Projects In Your Organization	Room C
	Structured Programming	E8	Clark W. Catelain Burroughs Corporation	Implementing Reliable Programs	Room D
3:00 p.m.	Coffee/Cola Break	--			Mezzanine
3:30 p.m.	Seminar Sessions: Part III				
	Behavioral Studies	AW	Dr. Michael A. Johnson University of Minnesota	Improving Your Inter-Personal Relationships	Room A
	Data Base	C8	Leo J. Cohen Performance Development Corp.	A Comparative Analysis of Data Base Software	North Ballroom
	Management Development	A8	Dr. David Norton, Kenneth Rau D. P. Management Dorporation	Management Control of the D. P. Function	Room B
	Mini-Computers	B8	Don Gross, Jim Krok, Marijane McDonough Mini-Computer Systems, Inc.	User Systems Design	Room 106
	Project Management	BW	Jerome Rolefson, Al Russell UW-Milwaukee, City of Madison	Project Management Applied	Room C
	Structured Programming	E8	Clark W. Catelain Burroughs Corporation	Related Management Techniques	Room D

ESPRIT de COMPUTING '76 -- THURSDAY, APRIL 29, 1976

TIME	EVENT	CODE	SPEAKER	TOPIC	LOCATION
8:30 a.m.	Opening Session	--	Dr. Michael A. Johnson University of Minnesota	Job Expectations. . . Self-Fulfilling Prophecies	North Ballroom
10:00 a.m.	Coffee/Rolls	--			Mezzanine
10:30 a.m.	Seminar Sessions: Part I				
	Behavioral Studies	A9	Dr. John G. Geier University of Minnesota	Evaluating Your Leadership Style	Room A
OK	Data Base	C9	Stephen L. Robinson Performance Development Corp.	Overview	North Ballroom
	Data Communications	D9	Dr. Dixon R. Doll DMW Telecommunications Corp.	Overview	Room 106
	Management Development	AT	Dr. David Norton, Kenneth Rau D. P. Management horporation	Long-Range Planning	Room B
	Project Management	B9	Jerome Rolefson, Al Russell UW-Milwaukee, City of Madison	What It Is, Does and Requires	Room C
	Structured Programming	E9	Clark W. Catelain Burroughs Dorporation	Designing with Reliability As A Goal	Room D
12:00 Noon	Luncheon-Address	--	Capt. Grace M. Hopper U. S. Dept. of Navy	Future Possibilities. . . Software/Hardware	South Ballroom

ESPRIT de COMPUTING '76 -- TUESDAY, APRIL 27, 1976

TIME	EVENT	SPEAKER	TOPIC	LOCATION
1:00 p.m.	Opening Remarks			North Ballroom
		Richard F. Gehrt CDP Region 5 Vice-President		
		Kathleen Wagner Micke CDP Conference General Chairperson		
		Allen B. Russell Southern Wisconsin Chptr. Pres.		
1:15 p.m.	Welcome	Honorable Paul R. Soglin Mayor, City of Madison		North Ballroom
1:30 p.m.	Keynote Presentation	Robert H. Courtney IBM Corporation	Computer Security: Disaster Avoidance and Recovery	North Ballroom
3:00 p.m.	Coffee/Cola Break			Mezzaine
3:30 p.m.	Panel Discussion			North Ballroom
		Richard W. McCoy, CDP UW-Madison	Moderator	
		Atty. Susan H. Nycum Chickering and Gregory	Legal Issues Re: EDP Software/Hardware Contracting-Part I	
		Terry D. Miller Government Sales Consultants	Legal Issues Re: EDP Software/Hardware Contracting-Part II	

ALAN R. KAPLAN / Contributing Editor

COBOL Means Business... ...and so does DATA GENERAL

A major COBOL implementation is always newsworthy, particularly when it amounts to nothing less than a full ANSI implementation at the highest level. As it happens, such an announcement was made this month by none other than mini manufacturer Data General, and barely one year after announcing its first end-user commercial system, the Eclipse-based C-300. In that short span, the Southboro, (MA) onetime "iron-maker" has managed to marry some of the most powerful hardware ever offered in the $100,000 range with as ambitious a package of operating software as anything provided by the established edp "biggies."

Unlike its mini competitors, which, when they were able to offer COBOL at all, traditionally elected to implement one or another of the several COBOL subsets designed specifically for smaller systems ("Required" COBOL, "Basic" COBOL, "Compact" COBOL, "Hypo" COBOL, etc.), Data General's highest-level ANSI-74 implementation means that it can take full advantage of COBOL's *chef raison d'etre:* program transferability. Almost from the moment a COBOL program was first successfully transferred between two different computers (ironically, between an RCA 501 and a UNIVAC II, in December, 1960), ease of conversion has been more important than ease of use. The enormous cost of manual conversion, once estimated by the Navy's Grace Hopper to be on the order of 40.5 person-years per 500 programs (typical conversion), will remain – at least until such time as it becomes possible to translate from one language to another as automatically as COBOL programs can today be transferred across machines.

Of the eleven functional modules specified by the ANSI X3.23-1974 standard, all but two are implemented at the highest level. The rarely-used Report Writer module was withheld because other modules contain procedures for report generation and because of the abundance of packaged (and better) report writers available from independents. CAM, Data General's Communications Access Manager, replaces the ANSI Telecommunications module, because the C-300 on which DG's COBOL is offered is a *multi*programming system. (As Honeywell's Robert W. Bemer pointed out in an address to the Tenth Anniversary Meeting of CODASYL, data communications specifications "should be unified and common to all programming languages that must coexist in the same multiprogramming environment," i.e., under the same operating system. In other words, the ANSI Telecommunications module is an anachronism – a holdover from uniprogramming systems.)

BEYOND COBOL

Impressive as it is, "full-implementation" only hints at What's available via DG's COBOL. The extensions DG offers for its COBOL put it far beyond the ANSI capabilities, particularly with respect to file management. Most of these extensions are through commands to INFOS, an exceptionally flexible data base management system developed concurrently with the design of the C-300 to make the most of that system's powerful hardware.

INFOS runs under the C-300's Mapped Real-time Disk Operating System (MRDOS) with a minimum of 128K bytes of core or semiconductor memory. Besides providing the conventional sequential, random and indexed sequential access methods supported by most COBOLs, INFOS extends its indexed sequential access method (ISAM) to produce a data base access method (DBAM) that offers such additional features as dynamic data base inversion (lets multiple indices access a single record) and multilevel indexing (allows hierarchical key specification).

REMOTE DEBUGGING. *DG's COBOL has an interactive debugger so users can debug programs from the console terminal, using English commands and free-format input.*

Moreover, the INFOS extension is interfaced *directly* to DG's COBOL. That is, INFOS is brought in not through calls, but through verbs, e.g., "Define Subindex," "Retrieve Key." This COBOL-INFOS combination gives the C-300 a data base management capability that matches or exceeds that of machines many times its price. Compared with existing manufacturer-supported minicomputer systems, there is no comparison. Few have any kind of COBOL; those that do rarely include the Sort, Interprogram Communication and/or Segmentation modules;and fewer still have an ISAM.

In addition to the COBOL modules enhanced by INFOS (e.g., Table Handling and Indexed I/O), which under the various ANSI levels routinely provide varying degrees of indexing and referencing, DG offers many other extensions. An example is DG's Debug module. Unlike the standard ANSI batch debugger, DG's is interactive, allowing users to enable and disable breakpoints, and to examine and modify data items dynamically, using such simple commands as "Compute" and "Display."

Microprocessors can now be economically used where in the past even digital logic was prohibitive. The chips are going into the design of appliances, consumer entertainment products, including games, and automobiles. But whole new applications opportunities are opening in business and industrial equipment and even scientific apparatus that could have used, but could not justify, minicomputer control.

Much has been said about the high cost of software, but the microprocessor-based system can be programmed economically by people who have not had formal training and experience as programmers. The "do-it-yourself" approach obviates the need for hiring specialized consultants, even for reasonably complicated systems. The overall design can be handled by one person without recourse to outside logic circuit or programming experts.

None of the companies offering training systems is claiming instant learning: just as in the mastery of any complex subject, the student's progress is proportional to his effort and ability. However, some of the instruction texts are organized so that the student can work at his own pace and periodically check his comprehension.

Some firms are offering formal classroom and lab instruction taught by experienced instructors. These classes encourage student progress and assist him over difficult material. Classes are generally recommended for all who can afford to take the time to attend.

All training is based on the characteristics and instruction set of the processor chip used within the training aid. The device selected is typically a popular, low-priced unit and those now favored are the Intel 8080 and Motorola 6800, both eight-bit, single-chip processors.

The prime objective is the demonstration of the principles, concepts and application of one device and, by extension, all microprocessors. The student is taught to reconsider the problem in a new way, organize it in a sequence of steps, and translate those steps into a flow chart. The tradeoffs between hardware and software options must be carefully evaluated before writing the actual program.

Typical training aids are self-contained units with the processor, memory and I/O circuits mounted on plug-in circuit boards and a power supply. Switches and indicators on a front panel are used to address and display the memory contents as well as the status of the processor.

Access to the program memory is normally by front panel switches in machine language in accordance with the instruction set of the processor being used. The address and content of the program memory may be displayed to facilitate program loading. Most training aids, however, have little or no provision for interfacing with external I/O devices so these functions are simulated with the panel switches and displays.

The Micro-Designer from E&L Instruments goes one step further by providing effective means for interfacing actual relays, external control devices and displays. Thus the trainer is also a hardware and software development tool. One of the prominent firms offering training systems, E&L Instruments furnishes programmed texts called "Bugbooks" that cover the same subjects as the microprocessor maker's manuals, but are also organized as laboratory workbooks. (See MODERN DATA, Dec., 1975, page 49.) Considerable attention is given to practical benchtop experimentation. ■

STANDARDS

Proposed ISO Standard "Computer Output Microfiche"

by Don M. Avedon
Technical Director, NMA

In order to bring our readers up-to-date on international standards for COM, we are presenting below a draft international standard, DIS 5126, which has been prepared by Working Group 1 of Technical Committee TC46/SC1 of the International Organization for Standardization (ISO). This working group is concerned with all aspects of microfiche standardization, and is composed of the following members: Denmark, France, South Africa, Belgium, Germany, Sweden, Switzerland, United Kingdom, Canada, and the United States, which serves as secretariat. The chairman of the group is Don M. Avedon.

It should be noted that this proposed standard covers effective reductions of 24X and 48X, but deliberately omits 42X. It should also be pointed out that, unlike United States COM standards, this document is limited to microfiche and to alphanumeric systems and does not cover roll film or graphics. On the other hand, it does cover both quality and format requirements, which are in separate standards in the United States.

The draft international standard is currently out for letter ballot of all ISO member countries. Since it is not in conflict with United States standards and is compatible with American products, the United States intends to cast an affirmative vote for this standard.

1. SCOPE AND FIELD OF APPLICATION

This international standard specifies the characteristics of transparent A6 size computer output microfiche at reductions of 1:24 and 1:48. Formats with frame sizes are provided for A4 and 279 mm X 355 mm (11 inches X 14 inches) equivalent page sizes at each reduction. This international standard does not cover computer output microfilm in 16-mm or 35-mm roll formats, which will be the subject of a separate standard, nor does it cover microfiche of source documents, which is the subject of ISO 2707 and ISO 2708. Depending on requirements, the microfiche may be a negative or positive.

2. REFERENCES

ISO 543, Cinematography–Motion-picture safety film–Definition, testing, and marking.

ISO 1073, Alphanumeric character sets for optical recognition.[1]

ISO/R 1831, Printing specifications for optical character recognition.

ISO 2707, Transparent A6 size microfiche of uniform division–Image arrangements 1 and 2.

ISO 2708, Transparent A6 size microfiche of variable division–Image arrangements A and B.

3. PHYSICAL CHARACTERISTICS

3.1 Sheet Size

The external dimensions of the distribution microfiche shall be

$$105\,{}^{0}_{-0.75}\text{ mm} \times 148\,{}^{0}_{-1}\text{ mm}$$

(See Figures 1, 2, 3, and 4, and annex A.)

The tolerances specified for the distribution microfiche apply immediately after processing. The measurements shall be made when the film has come to equilibrium at 23 ± 2°C and 50 ± 5 percent relative humidity. Size variations due to raw stock slitting and processing have been considered in determining the tolerances. Additional size changes may occur during aging, especially for films on cellulose ester supports. See annex A, clause A.3.

[1] At present at the stage of draft (revision of ISO/R 1073-1969).

Table 1. Microfiche Formats*

Arrangement	Equivalent document size	Reduction†	Columns	Rows	Number of frames
2‡	A4	1:24	14	7	98
3§	279 mm × 355 mm (11 inches × 14 inches)	1:24	9	7	63
4‖	A4	1:48	28	15	420
5#	279 mm × 355 mm (11 inches × 14 inches)	1:48	18	15	270

*Arrangement 1 is not applicable to COM. See ISO 2707.
†See 4.3.
‡See Figure 1.
§See Figure 2.
‖See Figure 3.
#See Figure 4.

Temporary size changes due to temperature and humidity changes are described in annex A, clause A.4.

3.2 Formats

Table 1 specifies the arrangements that shall be used for computer output microfiche.

3.3 Thickness

The thickness of the film used for the microfiche shall be not greater than 0.22 mm.

3.4 Heading Area

The heading area above the image area of each microfiche shall be reserved for identification references to be legible without magnification.

3.5 Heading Area Backing

An opaque or semi-opaque backing[2] for the heading area is optional. If a heading area backing is used, it shall not increase the thickness of the fiche by more than 0.01 mm.

3.6 Squareness

Each side edge of the microfiche shall be perpendicular to the bottom (reference) edge within ± 0.13 mm (± 0.005 inch) for each 25 mm of height of the microfiche. Squareness shall be tested by placing the microfiche between two rectangles of dimensions 105 mm X 148 mm and 104.75 mm X 147.50 mm respectively, to represent maximum and minimum limits.

3.7 Identification of Sensitized Side

To facilitate microfiche-to-microfiche copying, a notch or a corner cut may be used to identify the sensitized layer of the microfiche. When a notch is used, it shall be made in the shorter side of the sheet, near the appropriate corner. The notch may be of any shape, but it shall not penetrate more than 1.6 mm inward from the edge of the microfiche.

When a corner cut is used, it shall be made in the appropriate corner of the heading area only. The cut shall extend a nominal 6 mm along the longer side of the microfiche and a nominal 9 mm along the shorter side of the microfiche.

The sensitized side shall be identified by one of the following methods:

> *Method A.* When a sheet of raw film or a microfiche is held with the long sides in a horizontal position with the heading area at the top and the notch is in the lower right-hand corner, or the corner cut is in the upper left-hand corner, the sensitized side will be toward the observer.
>
> *Method B.* When a sheet of raw film or a microfiche is held with the long sides in a vertical position and the notch or corner cut is in the upper right-hand corner, the sensitized side will be toward the observer.

3.8 Corner Rounding

The corners of the microfiche may be rounded, with the exception of those corners which have been subjected to a corner cut (see 3.7). When corners are rounded, the process shall not remove more than 3 mm of either of the two edges forming the corner.

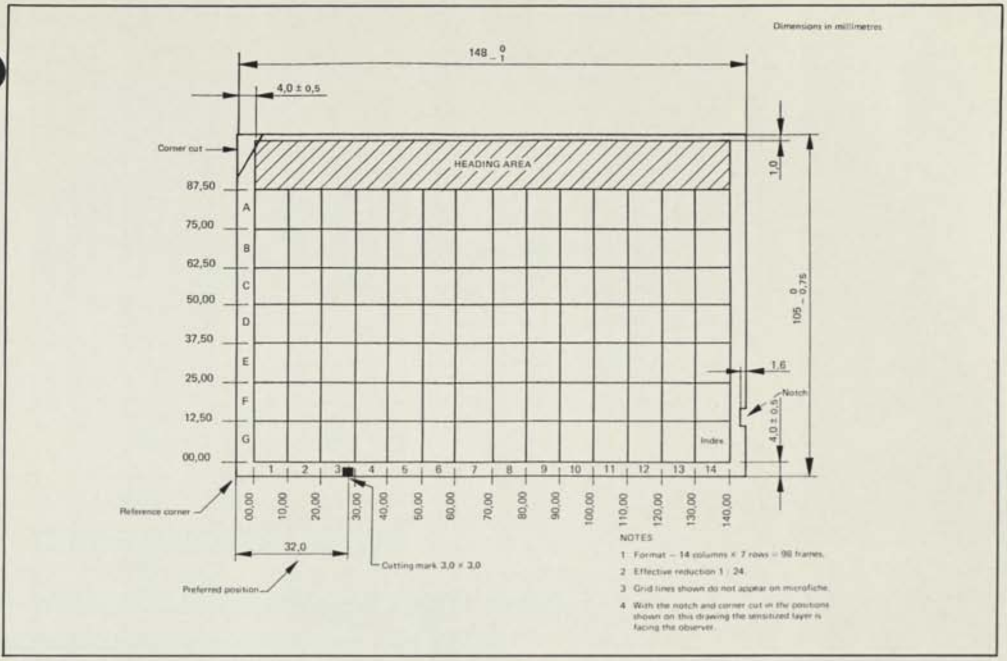
3.9 Measurements Involving Cut-Off Corners

Where segments of an edge have been removed by corner rounding or corner cuts, a straight line extending the remainder of the edge in the relevant direction shall constitute the basis for measuring dimensions and spacing.

3.10 Safety Film

The film used shall comply with ISO 543.

[2] The use of such backing restricts duplication.



NOTES

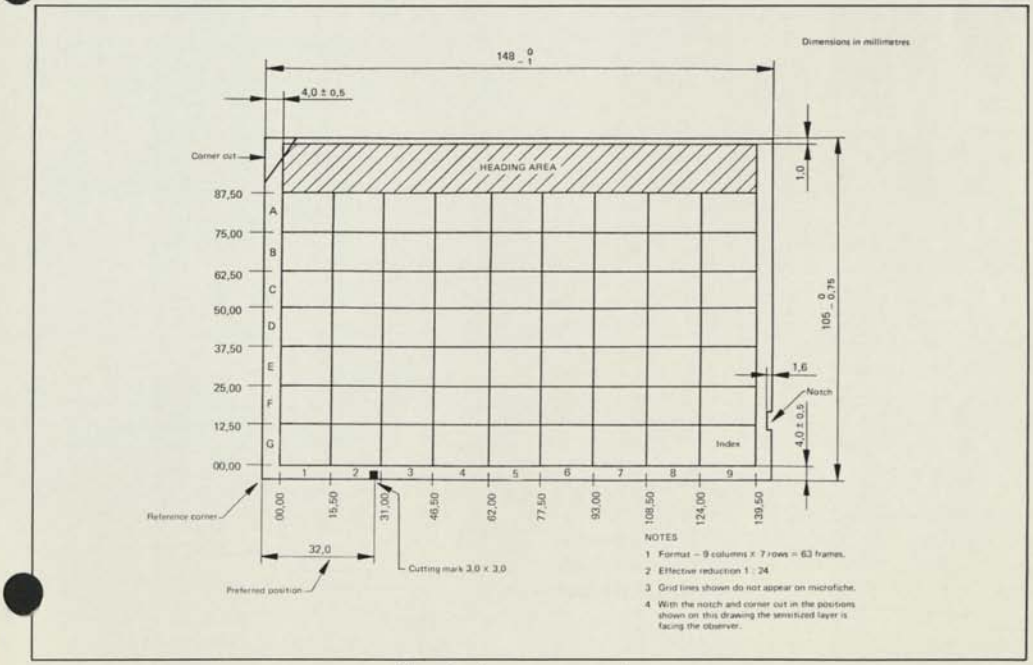
1 Format – 14 columns × 7 rows = 98 frames.

2 Effective reduction 1 : 24.

3 Grid lines shown do not appear on microfiche.

4 With the notch and corner cut in the positions shown on this drawing the sensitized layer is facing the observer.

Figure 1. Image arrangement 2.



NOTES

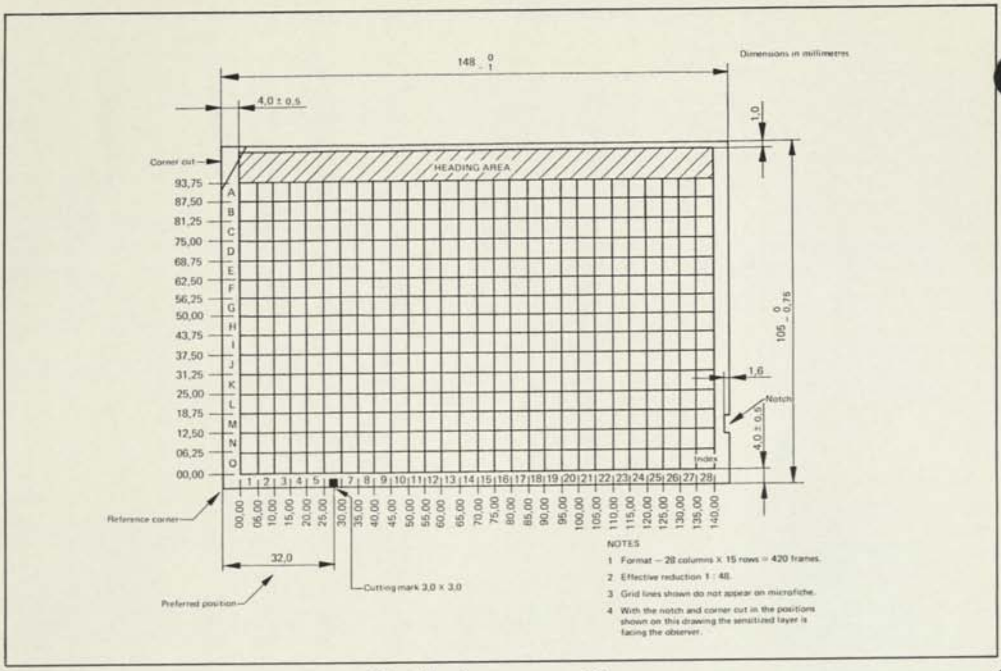
1 Format – 9 columns × 7 rows = 63 frames.

2 Effective reduction 1 : 24

3 Grid lines shown do not appear on microfiche.

4 With the notch and corner cut in the positions shown on this drawing the sensitized layer is facing the observer.

Figure 2. Image arrangement 3.



NOTES

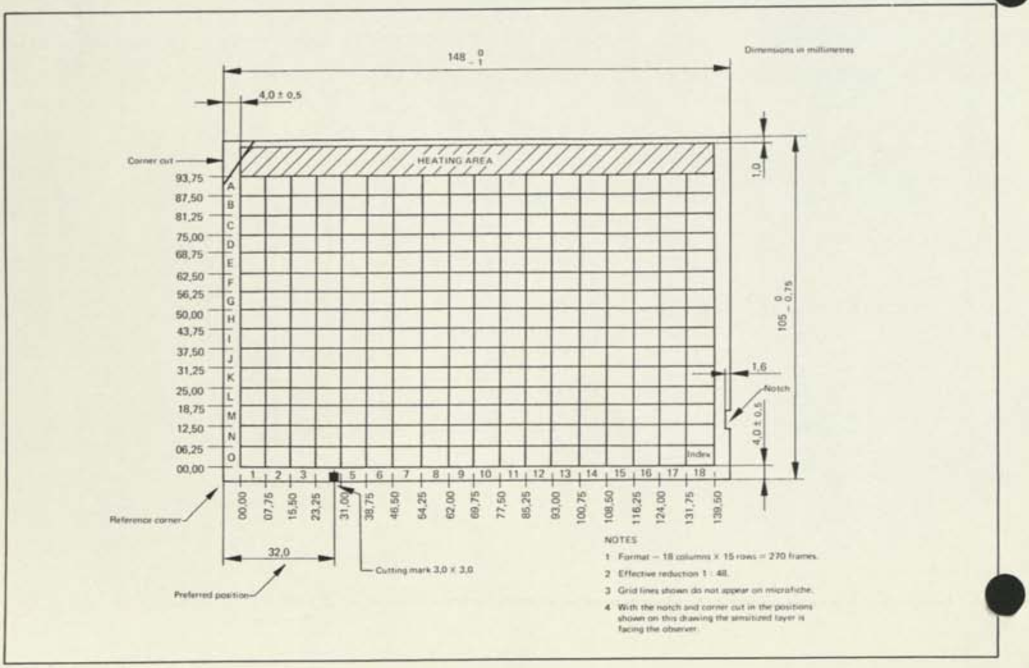
1 Format — 28 columns × 15 rows = 420 frames.

2 Effective reduction 1 : 48.

3 Grid lines shown do not appear on microfiche.

4 With the notch and corner cut in the positions shown on this drawing the sensitized layer is facing the observer.

Figure 3. Image arrangement 4.



NOTES

1 Format — 18 columns × 15 rows = 270 frames.

2 Effective reduction 1 : 48.

3 Grid lines shown do not appear on microfiche.

4 With the notch and corner cut in the positions shown on this drawing the sensitized layer is facing the observer.

Figure 4. Image arrangement 5.

4. FRAME SIZE AND FORMAT

4.1 General Layout

Frame size and placement of images shall be in accordance with Figures 1, 2, 3, or 4, whichever is appropriate.

4.2 Microimage Placement and Orientation

Microimages shall be positioned within one of the grid patterns shown in Figures 1, 2, 3, and 4, whichever is appropriate. All measurements use the bottom edge and the bottom left-hand corner of the fiche as reference. When the fiche is held so that the heading is right-reading and upright, microimages shall always be right-reading and upright.

4.3 Effective Reduction

Image arrangements 2 and 3 shall have an effective reduction of 1:23 to 1:25.5. Image arrangements 4 and 5 shall have an effective reduction of 1:47 to 1:50.

4.4 Area and Placement of the Heading

The minimum areas reserved for the heading are indicated in Figures 1, 2, 3, and 4 by shading. If additional heading space is required, the area allocated to the next entire row or rows of images shall be used. When more than one row is used for the heading, the frame identification, as specified in 4.8, shall remain unchanged. The heading area constitutes the top of the microfiche. The minimum area reserved for the heading shall be used only for heading and identification purposes on all microfiche, and not for microimages.

4.5 Identification and Heading Arrangement

The document or identification number of the fiche shall be in the left-most part of the heading area. All characters in the heading shall be upright and right-reading. All entries shall be readable without magnification.

4.6 Pagination[3]

When the microfiche is held so that the heading is upright and right-reading, the first microimage shall be placed in the top left corner of the grid area. Succeeding frames may appear either in sequence downward from left to right from column to column (vertical pagination), or in sequence from left to right and downward from row to row (horizontal pagination).

4.7 Trailer Microfiche Identification

When trailer microfiche are used, each microfiche in the set, including the first one, shall be identified sequentially, and the last microfiche in the set shall be identified as the last one.

4.8 Frame Identification

Where coordinate identification is used for location of images, alphabetic characters shall be used to identify rows. Starting at the top row below the heading area, the first row shall be A, the second B, and so on. Columns shall be identified by numerals starting at the left. The first column shall be 1, the second 2, and so on.

The indication of coordinates on the microfiche is optional. If coordinates are shown on the microfiche, they shall be located in the margins (see Figures 1, 2, 3, and 4) or in the lower portion of the heading area.[4]

5. AUTOMATION REQUIREMENTS: CUTTING MARK

Each microfiche may carry a cutting mark to provide for automatic cutting of processed roll film into microfiche. This cutting mark shall be 3 mm X 3 mm square, and the center of the square shall be located 32.0 ± 0.2 mm from the left edge of the microfiche, on the bottom edge.

6. INDEX FRAME

If an index to the microfiche is to be provided, the last microimage of the index shall be placed at the bottom right corner of the grid area. Preceding index frames shall appear in reverse sequence subtracting from the allotted format.

7. INFORMATION DENSITY (CHARACTER PACKING)

Standard dimensions of the computer output microfilm (COM) microforms are based on effective reductions. The character packing density of an equivalent paper document is specified as 60 characters per square inch, corresponding to a character pitch of 2.54 mm (0.1 inch) and a line spacing of 4.23 mm (0.6 inch).

8. ALPHANUMERIC CHARACTER PARAMETERS

The alphanumeric character shall meet the legibility requirements specified in 9.1 with the objective of insuring human readability as well as OCR compatibility. A font and some dimensions which are designed to meet this objective are given in part

[3] See annex B.

[4] When coordinates are placed in the bottom margin, they may interfere with automatic cutters sensing the cutting mark.

II of ISO 1073, and annex D of this international standard.

9. QUALITY REQUIREMENTS

9.1 Legibility of First-Generation Microfiche

9.1.1 Requirements

A square array of 12 lines, of at least 20 characters and symbols presented in a random sequence, and including all characters and symbols capable of being generated by the COM, shall be recorded in the center and each corner of the full frame size. Each of the five arrays should utilize different random number sequences. (See annex C for five different random number groups using 63 characters and symbols.) The test sample should contain a block of characters representative of each style of font used. Where information will be used in more than one orientation, i.e., most of the text reading left to right, but some text reading upwards, as in graphs, then blocks representative of each orientation should appear in the test sample.

The test samples should contain information compacted horizontally and vertically, representative of the maximum information congestion anticipated for use. Alphanumeric COMs generally have from 70 to 132 characters per line and a line spacing of 4.23 mm (6 lines per inch), while graphic COMs have more variation. Each character or symbol so generated shall be identifiable without error when viewed on a print or reader screen.

9.1.2 Test Method

A printer or reader magnification of not less than 12X shall be used so that the smallest size uppercase character height shall be a maximum of 1.6 mm (0.063 inch). The space between successive lines of characters in the array shall be no greater than 7/8 the height of the capital letter E. Alphanumeric COMs with a character height between 2.28 and 2.54 mm (0.09 and 0.1 inch) would use a maximum reader or print magnification to determine system image quality of 16X for nominal 1:24 reduction and 32X for nominal 1:48 reduction COM images. Viewing shall be in an ambient illumination of approximately 540 lx. Good quality-control practice dictates that this test be performed on a routine basis.

9.2 Legibility of Reproduction Copies

The subsequent generation which serves as the user copy shall meet the same legibility standard as that described for the first generation.

9.3 Curl and Bow

A fully processed microfiche cut to distribution size shall be placed convex side down on a flat surface for at least 6 hours in an atmosphere in which the temperature is 21°C and the relative humidity 50 percent, after which no part of the microfiche shall be more than 6.5 mm above the surface.

Editorial Note. Due to space limitations, Annex A—Dimensional Characteristics of Microfiche, Annex B—Commentary on Pagination Modes, Annex C—Legibility Arrays, and Annex D—Alphanumeric Character Parameters have been omitted. However, they may be obtained by writing to Don M. Avedon, Technical Director, National Micrographics Association, 8728 Colesville Road, Silver Spring, MD 20910.

Making Microfiche Irresistible

by Robert W. Bemer
Honeywell Information Systems

Robert W. Bemer is a senior consulting engineer at Honeywell Information Systems, Phoenix, AZ. He was formerly a staff consultant and editor of the Honeywell Computer Journal. Before joining Honeywell he was with General Electric as manager of systems and software engineering integration, and had previously held positions with the Univac Division of Sperry Rand, IBM Corporation, Lockheed Missiles and Space Division, Marquardt Aircraft and the RAND Corporation.

Mr. Bemer holds an A.B. degree in mathematics from Albion College and a certificate in aeronautical engineering from Curtiss-Wright Tech. He is a fellow of the British Computer Society, and has written numerous articles for computer publications.

ABSTRACT

Designing hardcopy and microfiche formats to exist interchangeably and coequally brings multiple benefits. An unusual feature of this new method is that the fiche copy as viewed in the reader is larger, not smaller, than the original copy! A single fiche may contain either 392 or 420 pages, each with 3/4 of the capacity of the usual A4 page. Thus the information density is tripled, without impairing (and in fact enhancing) readability.

Although I became one of the pioneers in 1971 by issuing a journal with an integral microfiche copy (in a pocket inside the back cover of the Honeywell Computer Journal – no longer in production), there still remains the feeling of being a novice. For a bullfighting parallel – although a novice, I am nevertheless a microficianado!

I sense that microfiche is retarded from full acceptance (except by students) because too much of it is 1-to-1 copy of either a typewritten page (perhaps of insufficient quality) or a printed page designed without consideration for fiche. Readability must be enhanced before acceptance will be complete, particularly for those of us with bifocal lenses.

The computer world is also bending to the forces of increased costs for paper and mail. Indeed, a recent article by Rogoff [1] argues that it is cheaper to send letters in encoded form, by packet switching, than in typed form by mail. I have for some time followed the practice of encoding by entering my letters into a computer and photocomposing the text. Gilbert Jones of IBM warns us that the day of the paper file cabinet is nearly over.

MOTIVATION AND SERENDIPITY

I think that I have stumbled onto an advantageous way of using microfiche copy. It might not have worked, had not several different processes become cost-effective at about the same time. As it is, there seems to be a promise of (as the title says) making the use of microfiche irresistible.

The method came from a study of reducing the consumption of 11" by 14" printer paper – not for the usual COM applications where multiple copies are disseminated for use and archival storage; Honeywell Information Systems was already doing that extensively. Our problem was with programmers and their huge dumps and other listings – thrown away after a glance or two. Would it be practical to replace them by COM masters that would never be copied, just discarded after diagnosis? The answer was *yes*, with a better than 2:1 cost advantage over printer paper.

A Self-Descriptive Page: This page shows the method by example. Anyone can verify it by filming the page at 24X and reading the fiche in a COM reader. The upper (or lower) pairs of "little pages" will fill the screen. (Columns are slightly shortened because J. Micrographics doesn't use A4 paper). The type is the actual Helvetica Medium font that we use.

As our study showed that 95% of the cost would be in fiche production, and only 5% for the fiche readers, it became evident that the programmer work station is likely to consist of an online terminal and an offline fiche reader for the 11" x 14" paper images. But with readers amortized so effectively and cheaply by the diagnostic function, it became irresistible to consider putting programming manuals and other documentation on fiche also.

We have always had special problems with manuals. Software is revised and corrected quite often, but the economics of hardcopy production and paper-oriented methods work against complete reissuance of manuals conforming to a specific software release.

Once we envision a programmer work station consisting of a terminal and microfiche reader, it is seen desirable to have all program manuals and useful documentation available in microfiche form at that station, for reasons of efficiency and storage. It has been argued that once we get to this point it may be more economical and practical, providing video terminals are used, to store manuals in the computer itself for display of portions as needed. However, there are advantages for using fiche in conjunction with terminals:

- Fiche produced photographically can contain diagrams, tables, photographs, and graphic-quality distinctions in text (i.e., variation in type size, weight (bold), slope (italic)) that enhance the extraction of meaning. Admittedly some such distinctions are possible in a more limited way on softcopy terminals.
- The manual may be in constant display while using the terminal in a working (not study) mode. For example, a procedure outlined in the fiche manual may be followed step by step. This argues favorably for using nothing but dual-carriage readers.

A question now arises on the mechanics of producing a manual on fiche. Certainly no one wishes to see a reproduction of a typewritten page. We are fortunate that our manuals have, for many years, been produced by a computer text editor system. The drawbacks have been:

N

- They are attuned to 8.5" x 11" paper, which is too big to handle when working at a terminal.
- Output was by IBM or DATEL typewriter-style terminals having constant (pica) spacing that is more difficult to read than graphic quality type. Such readability problems are worsened when fiche is used.

MECHANICS OF TEXT AND IMAGE PRODUCTION

A reasonable and economic goal is to use the same fiche reader to view both the manual and the COM output of printer pages without the added expense of a second lens for different magnification, and without the irksome task of changing the lens setting. In short, everything should be read at a constant 48, 42, or 36X.

One's first inclination is to produce the manuals via the COM units that produce the printer page images, but these drawbacks are encountered:

- The COM units now in operation cannot produce a full spectrum of characters and symbols, bold and italics, tables, etc. Even an upper and lower case alphabet is difficult to come by.
- Only pica (constant) spacing is available now.
- Fonts, in only one size, are not of full graphic quality.
- Although our new graphic COM equipment may solve all of these problems, photographs are still difficult, and a method is still needed for those locations not having this equipment.

Future equipment to meet all these requirements can be expensive, so the present situation may be fortunate. It forces us to traditional camera methods producing 24X fiche from 98 hardcopy images of A4 size.

The happy result is that we are now committed to produce fiche at 24X that will be read at 36X or 48X. Thus the image that will be viewed is 1.5 to 2 times as large as the copy that was photographed! This is a fine feature, for one of the hindrances to widespread fiche usage has always been poor or unacceptable legibility.

N + 1

Choosing the Page Size

Because we wish to read manuals (with text, diagrams, and pictures) using the same reader and magnification with which we read COM output, the design problem was this:

- What size of printed area, photographed at 24X by step-and-repeat camera, will permit a left- and right-hand page pair to occupy approximately the same screen area as the printed area of a 11" x 14" printer page (COM version) produced at 48X?
- The aspect ratio (width to height) should be equally suitable for the interior area of a standard 6" by 9" book, for the fiche reader, and for A4 (or 8.5" x 11") paper.
- Four pages should be photographed as a single frame for the standard 98-image fiche.

To get sufficient print quality we used the photocomposition methods developed for the Honeywell Computer Journal. Here our computer text editor was modified to produce input to the Page 1 System, which then carried out the hyphenation, justification, and photocomposition.

To derive the printed area (column width by column length plus page number) we examine the physical grid of the 24X fiche. It is 10 mm wide by 12.5 mm high. At 24X the corresponding full-size grid is 240 mm wide by 300 mm high (A4 paper is 210 by 297). Putting 4 pages in this area means that each would be in a grid 120 mm wide by 150 mm high. Alternately, each *pair* of pages has a grid 240 mm wide by 150 mm high. Figure 1 shows the actual dimensions chosen for the printed area.

Although these dimensions work, so might other choices. The actual choice was also influenced by hardcopy considerations. The 4.5" x 7" interior of a 6" x 9" book page has an aspect ratio of 1.55 (a basic consideration because a 37% photographic enlargement of our original copy will provide for a hard copy book of this size).

The page size must also serve for hard copy directly as it comes from the photocomposing unit. It turns out that the 83.5 mm x 128 mm size fits very nicely on A6 paper. We print these, add Bristol board covers, and punch two holes for individual rings. These permit the manual to fold flat at any place, for convenience at a terminal (see Figure 2).

N + 28

Choosing Font and Size

The fiche that is produced by this method will now have, in the usual row of 14 images, 28 images in the top half and 28 images in the bottom half, for a total of 392 pages per fiche. (If the heading area of the 48X format is suitable, we can get 420 pages per fiche.) What then is the most text that could or should be placed upon such pages? This is a function of point size. Our 83.5 mm-wide column has a 236-point capacity, quite standard for double-column printing. In fact, an optional A4 layout is possible by not breaking the columns vertically. However, this method is only suitable for hard copy.

I have found that 7-point type with a 2-point lead is very satisfactory. At the full 48X magnification this appears as 14-point with a 4-point lead, much like a child's primer. At 36X, or 75%, it is nearly 11-point, with a 3-point lead, and still very readable. In 6 x 9 book form this becomes 9.6-point -- again very suitable for reading from hardcopy text. I chose Helvetica Medium as the font, for its fairly thick strokes give good readability on microfiche. No serif fonts are acceptable, to my mind. Of course, the possibility of 8-point has not been ruled out, but we note that 7-point type is still very readable in our hardcopy manuals for terminal usage. This is due to both the graphic quality and the fact that they contain reference material, not novels. Perhaps you have reached this point without undue annoyance, even though it is in 7-point for the express purpose of illustrating the method.

Choosing the Image

The remaining option was whether the fiche should be a positive or a negative. Our COM customarily comes out negative for computer output, and this is the easiest photographic process. Our expert likes the negative form because foreign material, which fiche unavoidably attracts, does not show up so much. Despite all this, Some programmers prefer the positive form, and we let them specify their choice. When you can get, for a dime, a fiche containing 2 or 3 complete manuals -- I feel we can afford to give the programmer a new copy whenever too much foreign material collects!

Figure 3 shows the entire timesharing manual for the HIS Level 66 computer system (index at bottom). A black matte template is used for shooting. I feel that it enhances the layout by clean demarcation of images.

N + 29

(This Page is part of the text of this Article.)

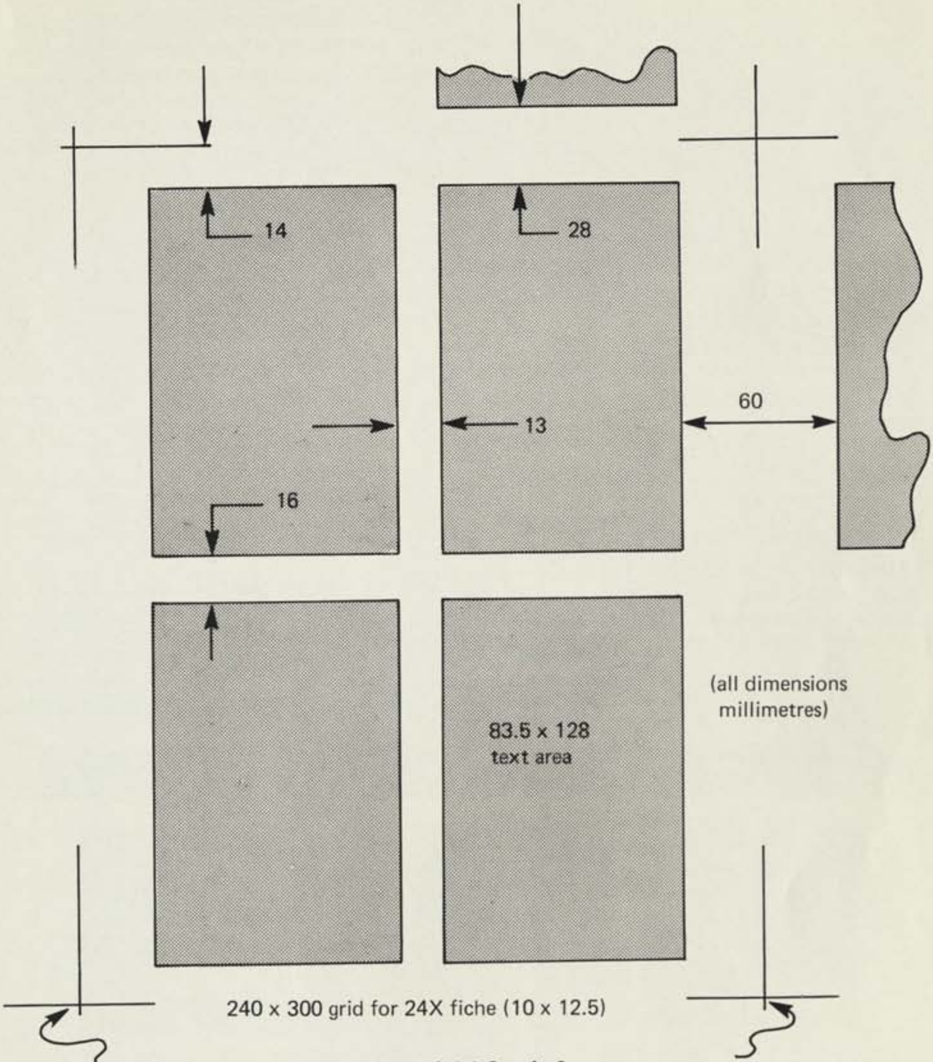


Figure 1. Layout of Little Pages for Camera.

ESSENCE OF THE METHOD

We now have a useful, effective and economical method, by the simple expedient of assaulting two accepted conventions:

- That the fiche process is subordinate to hardcopy production (sort of an afterthought, if you will), and that either COM or step-process filming must take the copy as it finds it.
- That the film will always be read at a magnifiation equivalent to, or less than, the reduction in production.

These conventions are now abrogated. Copy and layout are designed with both fiche and hardcopy considerations in mind. They are compatible, and in many cases the hardcopy production is 1:1 offset from the same master used for the fiche. Now the fiche process is not subordinate; it is coequal. Be-

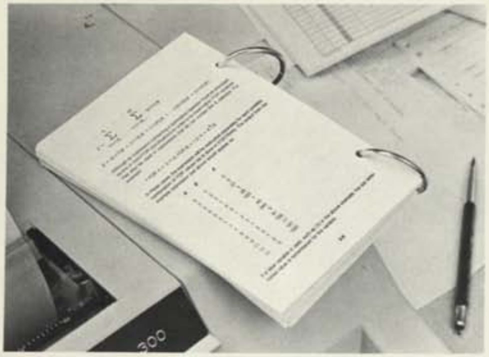


Figure 2. Hardcopy Manual.

cause this is so, we can consider reading at a higher magnification than the original reduction. All we had to do was design the layout so that the required portion *did* fall within the screen dimensions when read at the higher magnification.

WHAT ARE THE BENEFITS?

1. Removal of the usual 8-foot shelf of manuals that the programmer keeps to consume office and filing cabinet space. Not to mention the pile of listings on his table or floor.
2. If much desired, the 7-point original copy can be used by direct offset to make a hardcopy reference manual that costs only 1/3 as much in paper as present manuals do.
3. The user can choose between forms, or have both. For a trip, one can carry everything supporting a large computer system and its software. At home, he may be the programmer in charge, and thus need hardcopy for annotation.
4. Fortuitously, two pages can be photocomposed on a single galley (12" limit for the Videocomp). With column width guaranteed, four pages per galley can be produced by going, as the printers say, "two-up." As a galley costs us less than $2.40 each, flat rate, the individual page cost is about $0.60! This gives an equivvalent rate of $0.80 per A4 page. Try to touch that with your typist or present composition methods! In fact, it seems to be a general principle that the money saved in photocomposing at a smaller point size more than offsets any incremental camera costs in producing stripped mechanicals at target size.
5. The low cost of fiche permits us to reshoot the manual in the form of the previous version, except annotated by circled numbers for each changed section. The changed sections are shown in revised form elsewhere on the fiche. As soon as the user comes to a changed section,

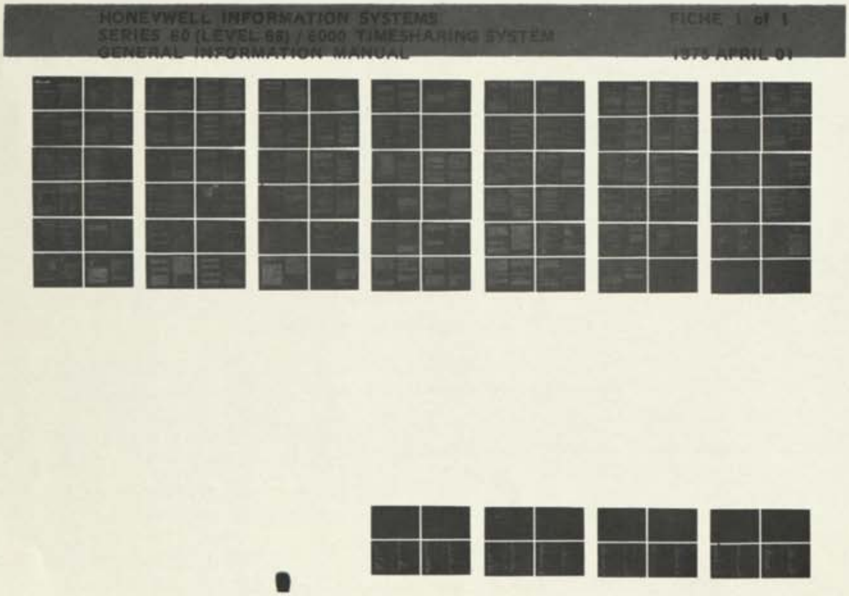

HONEYWELL INFORMATION SYSTEMS
FICHE 1 of 1
SERIES 60 (LEVEL 66) / 6000 TIMESHARING SYSTEM
GENERAL INFORMATION MANUAL
1975 APRIL 01

Figure 3. Fiche Version of Extensive Computer Manual.

he moves the carriage to display the corrected and revised section. This is an improvement uniquely possible to fiche. To know simultaneously what it used to be, and what it is now, conveys real intelligence. A change bar does not.

REFERENCES

1. Mortimer Rogoff, "The case for electronic mail," Modern Data, 1974 December, 34-37.

NATIONAL MICROGRAPHICS ASSOCIATION

8728 COLESVILLE ROAD
SILVER SPRING, MARYLAND 20910
301-587-8444

January 5, 1976

Mr. Robert W. Bemer
Honeywell Information Systems
PO Box 6000
C 61
Phoenix, Arizona 85005

Dear Mr. Bemer:

Enclosed please find two copies of the January/February 1976 issue of the Journal of Micrographics in which the article "Making Microfiche Irresistible" appears, and some tear sheets of the article itself.

Thank you for this valuable contribution to the Journal. We believe this article will be of great interest and benefit to many people in the micrographics industry.

Enclosed is information on reprints of Journal articles. Please let us know if you would like a few additional copies of the Journal if they are available.

Sincerely,

Don Avedon

Don M. Avedon
Editor

DMA:stw
Enclosures

'76 The spirit of micrographics
NMA
APRIL 27-30, 1976 CHICAGO, ILLINOIS

1975

DATA BASE DIRECTIONS
The Next Steps

Proceedings of the Workshop of the National Bureau of Standards and the Association for Computing Machinery, held at Fort Lauderdale, Florida, October 29-31, 1975

John L. Berg, Editor

Systems and Software Division
Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D.C. 20234

Richard G. Canning, General Chairperson

Working Panel Chairpersons:

Donald L. Adams, Robert W. Bemer, George Dodd, R. Michael Gall, and Charles D. Trigg

Sponsored by:

National Bureau of Standards

Association for Computing Machinery

U.S. DEPARTMENT OF COMMERCE, Elliot L. Richardson, *Secretary*
Edward O. Vetter, *Under Secretary*
Dr. Betsy Ancker-Johnson, *Assistant Secretary for Science and Technology*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Acting Director*

Issued September 1976

Library of Congress Cataloging in Publication Data
Main entry under title:
Data base directions, the next steps.
(National Bureau of Standards special publication ; 451)
Supt. of Docs. no.: C13.10:451
1. Data base Management--Congresses. I. Berg, John L. II. United States. National Bureau of Standards. III. Association for Computing Machinery. IV. Title. V. Series: United States. National Bureau of Standards. Special publication ; 451
QC100.U57 no. 451 [QA76.9.D3] 602'.1s
[001.6'442] 76-608219

National Bureau of Standards Special Publication 451

Nat. Bur. Stand. (U.S.), Spec. Publ. 451, 175 pages (Sept. 1976)
CODEN: XNBSAV

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1976

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402
(Order by SD Catalog No. C13.10:451). Stock No. 003-003-01662-4 Price: $2.40
(Add 25 percent additional for other than U.S. mailing).

4. STANDARDS: A DATABASE IMPERATIVE

STANDARDIZATION

4. STANDARDS: A DATA BASE IMPERATIVE

Working Panel Report on Standardization

Chairman: Robert W. Bemer

Biographical Sketch

Robert W. Bemer is a Senior Consulting Engineer with Honeywell Information System, Inc. His extensive list of accomplishments include:

Director of Programming Standards at IBM in 1962,
Developed original scope and program of work for ASA X3
and the ISO TC97 standards body,
Chairman, TC97/SC5, Common Programming Languages.

In addition, Mr. Bemer was editor of the Honeywell Computer Journal and the publication, "Computers and Crisis." Earlier in his career, while at IBM, he developed COMTRAN, a predecessor of COBOL, and XTRAN, a predecessor of ALGOL. He is now chairman of the ANSI SPARC Study Group on Text Processing.

Participants*

Thomas Bergin
R. E. Blasius
Milt Bryce
Jeffery Ehrlich
Chester Smith
Lee Talbert
Alan Taylor, Recorder
Ewart Willey

4.1 Terms of Reference.

Because the working group was requested to project the status of DataBase System** standards in the next five years, the membership was formed of selected active experts who are familiar with past and present standardization efforts in the computer field. Moreover, the membership was deliberately selected to include international views and experience.

The forecasting requirement in the terms of reference required the group to consider the perceived need for successful and safe database usage. All agreed that there was every indication that the current increase in database usage would continue, and that this would be beneficial to commerce and government in all countries. Provided, however, that some way existed to ensure that the users of such databases could have confidence in the validity of information produced without having personally to undertake the impossible task of understanding all of the complexities involved in the creation and operation of the database, as well as the use of the data stored there.

* Complete addresses and affiliations are in Appendix C
** A neologism; see section 4.4.1.

Standards were seen as a method of providing users with such confidence. Accordingly, the working group focused upon the realistic and attainable standards that current technology could be expected to provide, in this time period, to promote and protect safe database usage. The need to anticipate still unknown technological developments (a need implicit in all standardization processes) was regarded as part and parcel of this task.

4.2 Basic Premises.

- o Database standards embrace more than "management"

Database standardization activities are expected to cover all aspects of database usage, rather than just the narrow emphasis upon database management that has until now taken up most of the activity in the U.S. and other standardization groups. The already-developed CODASYL work on Data Description and Data Manipulation Languages offers a more-than-acceptable technical basis for standards. Because technical standards of some sort are prerequisite for any protective standards for database use, the working group believes that the perceived urgent needs for such protection will be based upon the CODASYL and related work.

- o Database standards are an international concern and responsibility

The identity of problems across international borders, a basic corollary of the easily-perceived identity of computer benefits that have similarly passed from nation to nation, makes it both likely and advantageous that the standardization work should be coordinated from an international, rather than simply national level. The volunteer effort that has fueled national effort in the past will not be able to cope fully with the apparently inevitable trend to internationalize database standards. The urgency and economy of obtaining internationally-agreed standards should, and do, more than justify the small amount of new funding required for their development.

- o The monetary and social aspect of database standards is large

It is difficult to calculate actual benefits of international protective standards, which can provide both safe operation of current databases and a safe, economic transition to the use of new hardware and software developments as they arrive, but we know them to be very great. Unprotected database usage has no real way of either assuring the integrity of the operation or protecting large investments in databases from being reduced or destroyed by technical obsolescence. Nor can we achieve the benefits from reducing training requirements, providing easy interchangeability, and using newer technologies that permit users to choose between central and distributive philosophies for database operations.

Program

Second National Symposium On The Management Of Data Elements In Information Processing

1975 October 23-24

National Bureau of Standards
Gaithersburg, Maryland

Sponsored By

National Bureau of Standards
U.S. Department of Commerce
and
American National Standards Institute
Committee X3L8

About the Sponsors

The National Bureau of Standards has responsibilities under the authorities of Public Law 89-306 (the Brooks Bill) and Executive Order 11717 for providing scientific and technological advisory services to Federal agencies relating to automatic data processing and for developing and maintaining ADP standards that provide for the economic and effective operation and utilization of Federal data systems. This responsibility includes the leadership of an Executive Branch program for standardizing data elements and representations. Currently several Federal Information Processing Standards (FIPS) are devoted to standard representations for data elements. Other standards are being developed for use in transportation, personnel and communications data systems.

Within the American National Standards Institute work has been undertaken to develop standards for representing data elements. These efforts are being conducted by the X3L8 Committee. Other national voluntary standards are being developed for data elements and representations in the fields of traffic-vehicle and banking data systems. Internationally standards are being developed for representations of data elements under the sponsorship of the International Organization for Standardization.

SECOND NATIONAL SYMPOSIUM ON THE MANAGEMENT OF DATA ELEMENTS IN INFORMATION PROCESSING

This symposium will include presentations on timely data management topics such as the role of the data manager, communications needs for data standards, data element directories, standard codes for character and control, use of check characters, data elements in bibliographic data bases, product coding, coding for clinical medicine, human factors, data resource management, data base management systems, and other related data standardization and data management efforts.

— Program —

Thursday, 1975 October 23

9:30 a.m. Introduction

Program Chairman
David V. Savidge
Special Assistant to the President
Comnet

A Focus on the Role of the Data Manager

Ruth M. Davis, Ph. D.
Director
Institute for Computer Sciences and Technology
National Bureau of Standards

10:45 - 11:15 a.m. Coffee

11:15 a.m. Introduction of First Session Presentations

Session Chairman:
W. Scott Haynie
Director, Operations Engineering Support
Western Union

International Standards for Data Transmission

V. N. Vaughan, Jr.
Chairman CCITT Study Group Sp A
A. T. & T. Company

What general characteristics should be standardized to permit international data transmission over telecommunications networks? A CCITT special study group has faced and will continue to face the age-old dilemma known to all who work on standards. That is, to set a standard too early can stifle further technological development but if the effort is made too late it is almost certain to fail, because a variety of "de facto" standards will be well entrenched. The problem is one of timing. That Sp A has largely overcome this dilemma and established a family of "standards" which are widely used is probably due to its collective sense of pragmatism in preference to perfection.

An Information Documentation Language:
Keystone to *Effective* Information Interchange

William M. Taggart, Jr., Ph. D.
Associate Professor of Management
Florida International University

The challenge of *effective* information interchange confronts the information processing community. Sooner or later it must be successfully approached to insure the development of *information* processing systems for a society that is increasingly dependent upon these capabilities for intra- and interorganizational communication that *is understood*. An Information Documentation Language standard appears to offer one approach for coming to grips with this challenge.

An Information Management View of Data Management

Marvin G. Wallis
Management Analyst
NASA

All too often data management seems to be primarily concerned with the management of data elements and often ignores the total flow of information. This is an attempt to relate information management as a broad view of information flow and information management of which data management is a part.

A Data Element Directory For A State Motor Vehicles Agency

John Roberts
Data Standards and Controls Bureau
New York State Department of Motor Vehicles

The New York State Department of Motor Vehicles operates one of the most comprehensive data processing installations in the United States of America. The socially and economically significant data in its computer files is a resource that demands protection and management. The data has to be identified, fully defined and its methods of representation documented for internal and external users.

Status of Army Materiel Command's Progression From Reports Control to Data Element Management

Edith F. Young
Headquarters
U.S. Army Materiel Command (AMCMS-IR)

Last year at the previous Data Element Sym-

"Business Factors Alias 'Codes': A Table Approach to System Design and Non-significant Coding"

Gary B. Johnson
Data Base Manager, Semiconductor Products Division
Motorola, Incorporated

It is our policy at Motorola that all codes must be managed external to programs in a master table system. This procedure is controlled by our System Directory and has considerably reduced our program maintenance.

An Adaptive File Management System
Udo W. Pooch
Texas A&M University

A program module is described defining an interface between an online - information system and the Input/Output Control System of the computer system. Programs belonging to this module are grouped by the function they perform: buffering, item relocation, compression, and dynamic priority assignment. The interface is adaptive in nature by physically reorganizing the File Structure based on usage statistics. Records are physically assigned to priority areas to reduce system I/O. The results of the reorganization is to construct working set files, a subset of the original file structure, having a substantial portion of all file activity.

Development of a Data Dictionary/Directory Using a Data Base Management System
Esther K. C. Lee
EDP Analyst
and
Dr. Edward Y.S. Lee
MTS
Jet Propulsion Laboratory
California Institute of Technology

The paper will discuss the research and development work on a data dictionary/directory (D/D) for a large construction data system. However, the structure of this D/D is such that it can be easily maintained by using a Data Base Management System and the techniques are being readily transferred to other large and complicated data systems allowing the data administrator to track the development of a growing collection of data elements.

A Methodology for Development of Standard Data Elements Within Multiple Public Agencies
L. D. (Dave) England
Assistant Chief, Data Systems Bureau
Texas State Department of Public Welfare

The authors have presented a case study of an extensive standards development project undertaken among nine Texas State agencies during parts of the years 1973 and 1974. It was financed with Federal/State matched funds through the auspices of the U.S. Department of Health, Education, and Welfare, the Texas State Department of Public Welfare, and the Texas Governor's Office of Information Services. The degree of transferability of the developed methodology and technology will be examined for potential implementation in other states.

The Standards Implications of the Developing Inter-Relationships Between On-Line Bibliographic Retrieval, Data Manipulation and Micrographics Display
Robert M. Landau
Science Information Association

A brief history of the rapid development of the On-Line Scientific And Technical Information Retrieval System (OLSATIRS) with emphasis on compatibility, convertability and standards problems is described. A similar description is given for the developments in the areas of data and micrographics. Procedural, nomenclature, interchange and economic problems in these three fields are then considered. Comparisons, contrasts, and inter-relationships between the three fields are provided. Problem areas and opportunities are suggested.

5:15 p.m. Adjournment

Friday, 1975 October 24

9:30 a.m. Introduction of Third Session Presentations

Session Chairman:
Sheila Smythe
Senior Vice President
Blue Cross - Blue Shield

Analysis of the Data Element and Microelement Structure of a Variety of Bibliographic Data Bases

Professor Martha E. Williams
Director of Information Retrieval Research Laboratory
Coordinated Science Laboratory, University of Illinois

In conjunction with a National Science Foundation sponsored grant to develop a "Data Base Mapping Model and Search Scheme" a large number of bibliographic data bases have been analyzed and broken down to the microelement level in order to determine a common set of macroelements contained (though in different forms with different tags and labels and having different representations) in the data bases. Mapping is based on commonality of data elements and subject content. Maps will show relationships between data bases.

Data Element Characterization for Clinical Medicine

E. R. Gabrieli, M.D., F.C.A.P.
Director, Clinical Information Center
State University of New York at Buffalo
and E. J. Meyer Memorial Hospital

A multidisciplinary national task force should be created, authorized, and funded, to compile a medical lexicon which can be the foundation of a medical information system. The current medical information crisis should be relieved by judicious use of the technology. The three steps, viz. construction of the medical lexicon, design of a code scheme, and planning the medical software for a cognitive memory will be accomplished very slowly, spontaneously. A highly visible, national, coordinative leadership in this crucial area may be the key to a successful man-machine system in medicine.

Unique Product Identification — One Number from Maker to User

Henry G. Littrell, III
Director of Technical Services
Distribution Codes, Inc.

A standardized product coding system across industry lines to improve production and expedite product movement.

10:45 - 11:15 Coffee

11:15 a.m. An Integrated Dictionary for Systems and Data Components

Curg Shields
M. Bryce and Associates, Inc.

This paper discusses two products developed and marketed by M. Bryce and Associates, Inc. "PRIDE", PRofitable Information by DEsign, is a planned approach to Information Systems design, development and implementation currently installed in over 430 systems organizations. "PRIDE"-Logik, **L**ogical **O**rganizing and **G**athering of **I**nformation **K**nowledge is an automated systems and data dictionary for use with "PRIDE". Both are proprietary products, copyrighted and trademarked by M. Bryce and Associates, Inc. Included in this paper will be a general discussion of these products and user reactions to an integrated systems and data dictionary.

Check Characters and The "Self-Checking String" — What, Where, Why, When and How

J. R. Nelson
The Upjohn Company
and
E. Hellerman
Bureau of the Census

"Self-checking strings" are character strings, used as data-base keys, which must pass a self-validation check before the keyed information can be transferred. Basic to self-checking strings is the concept of "check characters." The identification of a predominant class of systems used to compute and validate a self-checking string is made. Unified and detailed procedures are given for 1) defining a self-checking string, 2) computing a valid self-checking string, and 3) validating a string.

A Proposed Standard Routine For Generating Check Characters

Paul-Andre DesJardins
Hospital Saint-Michel-Archange
Canada

Some methods of generating check characters have become "de facto" standards. Unfortunately, they have many inefficiencies built-in which only the infancy of computer information processing could excuse. More over the lack of a true industry-wide standard could mean a check mate to anyone involved in data interchange.

So let us define some new standards and implement them in a single ANSI COBOL routine which generates the desired check character according to a given parameter.

A full listing of a proposed routine is presented.

On-Line Tactical Data Inputting: Research in Operator Training and Performance

Irving Alderman, Research Psychologist
Organizations & Systems Research Lab
U.S. Army Research Institute for the Behavioral and Social Sciences

This paper selectively reviews research in the development of job aids to improve training and performance of operators in the on-line translation and entry of messages. Current research in this area at ARI will be discussed.

1:00 - 2:00 p.m. Lunch

2:00 p.m. Introduction of Fourth Session Presentations

Session Chairman:
William Kenworthey, Jr.
Data Standards Coordinator
National Security Agency

Data Standardization

Harry S. White, Jr.
Chairman of ANSI X3L8 Data Standards Committee
National Bureau of Standards

Discussion on new data standardization and data management efforts (including data element directories and security guidelines) and other current federal, national and international data standards activities.

A Challenging Aspect of Word Processing

Victor G. Kehler
Systems Management and Programming
Directorate of Administration
Department of the Air Force — Headquarters

Presentation deals with the need for faster and less costly method for processing information within the Air Force. This paper traces the development, current status, problems and future of using computer assistance with word processing systems, computer output microform devices and electronic phototypesetters to process information.

System Design Considerations for the AMC Data Element Dictionary Directory System

Fred Puente
Automated Logistics Mgmt. Systems Agency (AMXAL-MBD)
U.S. Army Materiel Command

In early 1967, the Army Materiel Command implemented a Data Element Dictionary System in support of the development of a large, complex standard logistics system. The logistics system was designed to manage Army logistics at the wholesale level. The AMC Data Element Dictionary/Directory that evolved has supported system development through the functional analysis, systems analysis, design, programming, documentation, testing and implementation phases at six major Army logistics commodity commands located throughout the United States.

The Role of the Internal Auditor in Data Management

Richard H. Fahnline
Civil Service Commission

Internal auditing must address itself to meaningful audit of ADP. Brief definition of internal auditing. The need for evaluation of automated systems is great. Internal auditors and ADP professionals should work together to establish controls over ADP. Preliminary list of audit points for data elements. Cooperative development of standards for data elements, applications, and evaluation.

Techniques in Developing Standard Procedures for Data Editing

George W. Covill
Automation Industries
Vitro Laboratories Division

PLAYSCRIPT Procedures are a simple method of presenting information to employees who are engaged in the pre and post-processing activities of a data processing system. It can also be applied to explain and detail other administrative activities in support of data processing.

Closing Remarks

David V. Savidge
Program Chairman

4:00 p.m. Adjournment

General Information

Registration. A registration fee of $50 is being charged to help defray the total cost of conducting the Symposium. The registration fee will include a copy of the proceedings, lunches, coffee breaks, transportation to and from the headquarters hotel, and general meeting costs.

Advance registration is requested. Please complete the enclosed registration form, and mail with your fee (checks made payable to Second Data Element Management Symposium) to Mrs. Hazel E. McEwen, Institute for Computer Sciences and Technology, National Bureau of Standards, Washington, D.C. 20234.

The registration desk will be open at the Red Auditorium, National Bureau of Standards from 8:30 A.M. each day of the Symposium.

Meeting Room. All sessions of the Symposium will be held in the Red Auditorium in the Administration Building at the National Bureau of Standards in Gaithersburg, Maryland.

Housing. The Holiday Inn of Bethesda, 8120 Wisconsin Avenue, Bethesda, Maryland, has been designated the Symposium headquarters hotel. To insure reservations at the special Symposium rate of $23 single and $29 twin, return the enclosed hotel reservation card *directly* to the hotel *no later than October 8, 1975.*

Transportation. Bus transportation, included in the registration fee, will be provided each day between the Holiday Inn of Bethesda and NBS in Gaithersburg. Transportation will also be provided to the two Washington airports at the conclusion of the meeting on Friday. For those arriving by air, transportation to the Holiday Inn is available by regular airport limousine service. People driving their own cars will find ample parking space on the NBS grounds — follow the signs to visitors parking.

Proceedings. A copy of the proceedings of the Symposium will be mailed to each Registrant after the Symposium.

NOTES

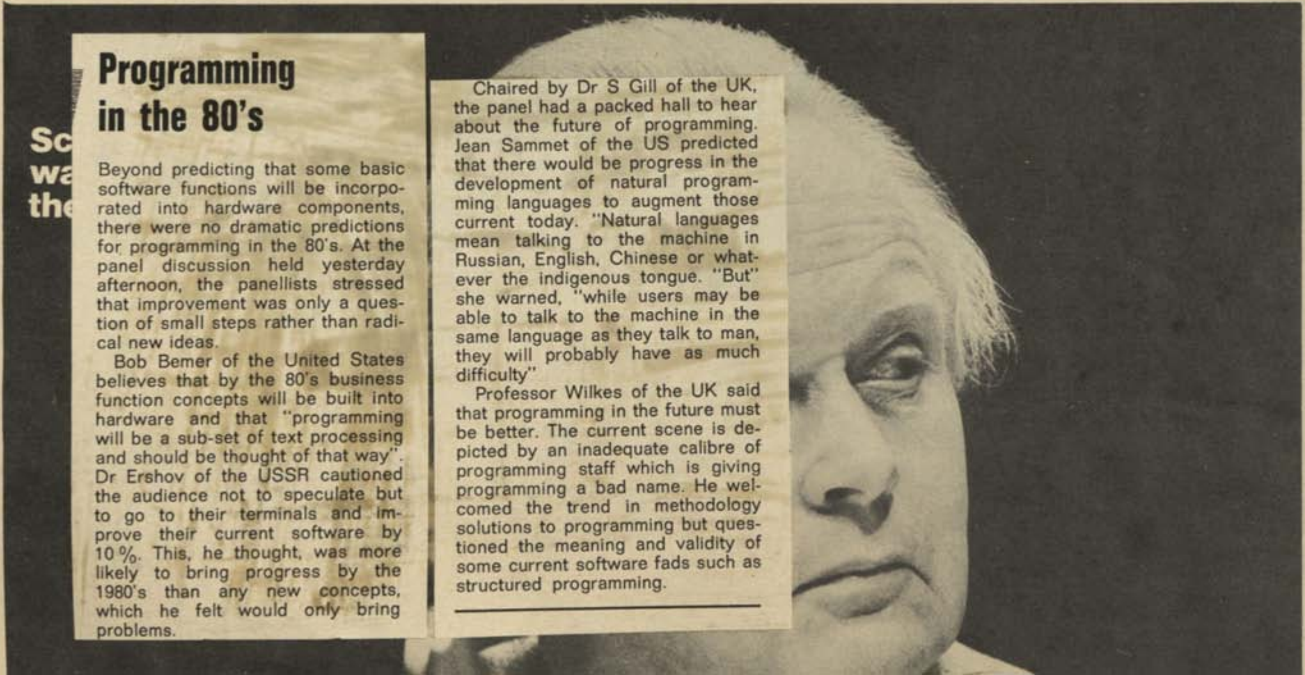
1974

data
CONGRESS
news

IFIP CONGRESS 74
MEDINFO 74

IFIP

No. 2 Wednesday August 7, 1974



Sc
wa
th

Programming in the 80's

Beyond predicting that some basic software functions will be incorporated into hardware components, there were no dramatic predictions for programming in the 80's. At the panel discussion held yesterday afternoon, the panellists stressed that improvement was only a question of small steps rather than radical new ideas.

Bob Bemer of the United States believes that by the 80's business function concepts will be built into hardware and that "programming will be a sub-set of text processing and should be thought of that way". Dr Ershov of the USSR cautioned the audience not to speculate but to go to their terminals and improve their current software by 10 %. This, he thought, was more likely to bring progress by the 1980's than any new concepts, which he felt would only bring problems.

Chaired by Dr S Gill of the UK, the panel had a packed hall to hear about the future of programming. Jean Sammet of the US predicted that there would be progress in the development of natural programming languages to augment those current today. "Natural languages mean talking to the machine in Russian, English, Chinese or whatever the indigenous tongue. "But" she warned, "while users may be able to talk to the machine in the same language as they talk to man, they will probably have as much difficulty"

Professor Wilkes of the UK said that programming in the future must be better. The current scene is depicted by an inadequate calibre of programming staff which is giving programming a bad name. He welcomed the trend in methodology solutions to programming but questioned the meaning and validity of some current software fads such as structured programming.

AMERICA GOES METRIC

MARCH / APRIL 1974

AMERICAN METRIC JOURNAL

THE BIMONTHLY PUBLICATION OF METRIC TRAINING and EDUCATIONAL PROGRESS

Rockwell International Goes Metric

Cut metric conversion costs with this instant computer

Brilliant Summit MCC offers 36 conversion programs, 5 calculator functions & memory!

The big switch to metrics affects us all, but if instant accuracy is vital to your business or profession, you need a lot more than a conversion table and sharp pencil. You need a computer which can convert any metric data—including square and cubic measurements—to U.S. terms, or from U.S. to metric. You also need an electronic calculator to apply those conversions. What you *really* need is both computer and calculator combined in a single unit—Summit's amazing 21-key, hand-held MCC!

More features:

- 36 conversion programs, including Fahrenheit-Celsius
- Addition, subtraction, multiplication and division, plus direct percentages
- Simple algebraic logic; you enter a problem exactly as you would write it
- Memory storage and retrieval
- Full floating decimal system
- Big, bright, 8-digit display
- Unwanted zero suppression
- Overflow, underflow, low battery indicators
- Finest quality components
- Slim-line: 4¾"x 2⅝"x 1", 5 oz. (12cm x 6.7cm x 2.5cm, 145g)
- Rechargeable nickel cadmium batteries
- Includes AC adapter/charger
- Deluxe padded carrying pouch
- 100% American-made
- Surprisingly low price

Optional: Imperial to metric—metric to Imperial conversions.

Suggested retail: only

$109.95

See your dealer, or order direct from

Summit International Corporation

P.O. Box 15736 180 West 2950 South
Salt Lake City, Utah 84115
(801) 486-7255

Some Ways to make Primary Journals More Effective

R. W. Bemer, 1974 April 29

The assumption is made that *all* content will be in mechanical form, stored in computer databases, and composed mechanically. From this the following benefits can be derived:

1. Concordances and indexes may be produced by computer. They provide:
   - a. Control of usage, spelling, cliches, legal aspects, and readability.
   - b. Information for the author about where, and how many times, each topic or term is used. This can be valuable for restructuring and/or tightening the text.
   - c. A spectrum for selecting keywords, categories, etc. for storage in databanks for future retrieval.

2. The ability to quickly generate a composed revision permits:
   - a. The author to see (not just galleys for technical correction, but) the entire article in approximation of its final form. By cutting and pasting (automatic with the computer text), the restructuring is not complicated, and is in fact encouraged wherever useful.
   - b. The editor to compose suggested revisions for author agreement. This may be for text, for text structure, for suborganization, for improved diagrams, etc. All of these are to maximize the chance of the message being understood.
   - c. The reviewers to add their editorial and technical suggestions for timely incorporation.
   - d. The editor to suggest text or illustrations that may enable the transfer of secondary aspects (as of an application) to usage or knowledge in other fields (e.g., similarity of techniques, human interfaces, environment).

3. References may be checked against standard form (See ISO TC46) and the proposed revisions displayed for approval. References are enhanced by identifying, in addition to the source item itself, specific reviews and abstracts of them. This can be done with a computer database.

4. Alternate composing methods are available, particularly for separating an article into dual form - hard copy and fiche copy (or magnetic media form). For example, a selection from, or the topical headings of, the nonhardcopy version can be included in the hardcopy portion. In some cases, particularly for references that are difficult for the usual reader to obtain, auxiliary information may be supplied on the microfiche copy.

Two opinions are expressed here:

- ■ The business of a primary journal is to communicate, not to glorify the author. Communication is expensive - in paper, mailing, etc. Therefore it should be tightened, and the author cannot be expected to do this (although he should certainly be willing to acquiesce to it). It becomes an additional editorial responsibility. But some of this can be done mechanically.

- ■ The role of the structure editor will become more important. He must ensure the presence of the necessary keys and links to a maximum of other disciplines. Categories and classes are very important to successful retrieval from databanks that are large and unspecialized.

1973

Letters

Backspace bungle

A great number of terminals were shown in action at the National Computer Conference. Unfortunately, most of the crt terminals had a major logical flaw, one that the designers and exhibitors apparently do not understand.

The flaw is in using the backspace character to move backward in the entry string to a desired point, effectively erasing those characters so that new characters may be input from that point.

The backspace is 0/8 in the ISO Code (ASCII for the provincial), encoded 00001000. It is defined as moving the printing position backward on the line. It is *not* defined as *erasing* the character in the position. On the contrary, it is generally used for diacritical and other marks to be overprinted (i.e. underline, overline, umlaut, accent acute, accent grave, tilde) to form composite symbols.

It is my contention that the backspace character must be used in a standard manner, as defined by the international standard. A file that prints to a typewriter-like terminal must print the same on a crt terminal or a photocomposition device.

If the crt terminal designers wish to operate in this combined backspace-erase mode, then they should use the soft copy controls presently being standardized in X3. Examples are the control characters for cursor movement and clearing the screen or line from the cursor position. Write to Bob Brown, Secretary of X3, at CBEMA, 1828 L Street NW, Washington, DC 20036, to get this information. Designers! do not, repeat, do not use backspace as you are. Change it, and use another separate and distinct key. Please. You are jeopardizing your position in the huge photocomposition and publishing market!

R. W. BEMER
Honeywell Information Systems Inc.
Phoenix, Arizona

Tilting at windmills

Should you have a "Don Quixote" award, Mr. Ferguson should win in an "IBM walk." ("System/3 Doesn't Belong to IBM," June, p. 62.)

While he preaches "typical IBM user —IBM sold him a system and that's the end of it"—he forgot to mention:

1. The company has less expensive products—1130/50, 1401H and 360/20 subs . . . that outperform the 3—for less money. (Burroughs 1700s run circles around the 3.)
2. Ask any IBM sales rep what his commission is for "selling" a 3.
3. "S/3's a user market." The applications customizer and field-developed programs prove the dependence on IBM.
4. I tried to buy compilers—IBM said they're not for sale!

I wish he were right.

GEORGE AHMUTY
Allis & George, Inc.
Westport, Connecticut

Mr. Ferguson replies: If, indeed, I am to receive the DATAMATION Don Quixote award, then I must be permitted to quote from my creator, Miguel de Cervantes: "There's not the least thing can be said or done, but people will talk and find fault."

The faults that Mr. Ahmuty raises are interesting ones, although I don't know particularly what they have to do with the article in question.

First, I'd like to suggest that he do a little more homework regarding "less expensive products . . . that outperform the 3." He specifically mentions the 360/20 as a better cost performer than the System/3, when the figures quoted in the article refute that stand without question.

Also, I fail to see the relevance of an IBM sales rep's commission for selling a System/3, although I wouldn't mind having it. Even though I am enthusiastic about its future potential, IBM's "Application Customizer" has been somewhat less than a rousing success. However, IBM does sell a lot of software (although most of it is not in the form of FDPs) just as they sell a lot of hardware. So?

Finally, I don't understand Mr. Ahmuty's fourth point at all. Let me state as a fact that IBM will sell their compilers (like any other software house) but not for reproduction, whether you own a System/3 or not.

And to return to Senor Cervantes, "You're leaping over the hedge before you come to the stile."

Testing . . . one . . . two

The article "Suspense Won't Kill Us" by Paul Armer (Editor's Readout, June, p. 53) could have better been titled "One Giant Leap Backwards." That is the idea that Mr. Armer appears to be trying to get across. To follow his logic, ACM and various colleges and universities across the nation should discontinue their curriculum in data processing science. They certainly cannot certify to the competence of graduates of their actual or suggested curriculum.

Mr. Armer rightly states that the present examination for the Certificate of Data Processing is not a perfect tool to attest to one's competence in data processing. Many years ago we certified equipment operators on the basis of multiple choice examinations. Today, after employing the techniques of systems engineering and performance testing in these same areas, we can look back and see how ridiculous those early methods were. In those days we could have discontinued our test. However, we felt they were a better measurement of one's abilities than no test at all. These imperfect tests gave us the statistical and practical background to devise the present performance-oriented test. They have proven beyond any doubt that they were, in fact, profitable.

Some years ago, the Data Processing Management Association saw a need for some method of certification. A body of knowledge had been building up in the field of data processing. We had exams that could give an idea of one's capability of assimilating knowledge in the field, but nothing that would show how much actual knowledge one had assimilated. The membership of DPMA worked together to devise some method to measure this assimilation. The result was the examination for a Certificate in Data Processing. Over the intervening years they have attempted in good faith to improve the capability of the examination to measure how much knowedge one has assimilated. At the same time, they have worked diligently to keep the examination updated with the rapidly changing field of data processing.

As with all things of this nature, the field has grown tremendously and has developed many specialized sub-fields. No longer can DPMA alone provide the base for the expertise necessary to continue updating the exam and developing a better tool of measurement. They have called upon the entire professional data processing community to assist. When we first heard about the idea of the Computer Foundation, many of us thought that DPMA was demanding too much power. Now that we have had time to analyze it, and to hear the criticism, I think it is clear why DPMA so wisely demanded some control. They knew that there would be a group of negative-thinking individuals who would attempt to destroy the program rather than build on it. This control which they asked for would assist in insuring the continuation of the program and its eventual improvement.

Now I think it is time for all of us to discontinue criticizing and to roll up our sleeves and get to work. Rather than killing the only tool of measurement we have, let's apply our efforts toward its improvement.

CORNELIUS M. HEAD
Indianapolis, Indiana

If we don't start to organize our societies, we may end up with as many as we have practitioners. My feelings on this situation are as follows:

1. Any organizational effort by the Computer Foundation, etc. . . . must be based on need—the need of a profession seeking identity and standards, not merely the need to spread the economies of a functioning dp organization.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 406, COMPUTER PERFORMANCE EVALUATION:
REPORT OF THE 1973 NBS/ACM WORKSHOP, MARCH 27–30, 1973, SAN DIEGO, CALIF. (Issued August 1975)

Standards in Performance Evaluation and Measurement

R. W. Bemer

Honeywell Information Systems, Phoenix, Arizona 85005

Giving "evaluation" equal billing with "measurement" opens the door to discussion of performance that is good or bad, as opposed to fast or slow. Through this opening come considerations of security and confidentiality, validation of software and hardware means for performing arithmetic operations and evaluating mathematical functions (to varying degrees of precision and accuracy), code independency, auditing and warranty, optional optimization in compilation of running programs in high-level languages, and retention of statistics of every aspect of operation—for later analysis and reduction of duplicate work.

Key words: Accuracy; audit; certification; code-independent; documentation; optimization; precision; run statistics; security; terminology; validation; warranty.

1. Justification

The United States Government has imposed certain requirements upon the manufacture of automobiles, i.e., to be constructed so as to withstand collision at X kph without sustaining more than $Y in damage, or the like. The Government has stated that requiring such action is within its right to protect the safety of its citizens.

Perhaps the reason that analogy of automobiles to computers is so facile is that computers are also a major restructurer of society. The newer computer uses have a greater than ever proportion of integration into human activities (even into the automobile). It seems certain that the computer has a direct effect upon not only the safety of our citizens, but also upon other rights. It might thus be reasonable to demand that software and hardware should also be built to certain standards to protect these rights.

Giving "evaluation" equal billing with "measurement" in the discussion of performance of computer systems is a major step, for it permits us to subsume good and bad performance as well as fast and slow performance. It enables us to view the need for confidentiality and security concurrently with performance measurement. There is probably much commonality in the requirements for both.

2. Nomenclature

The present intense efforts on performance evaluation and measurement indicate a movement toward professionalism in the computing field. Yet inspection of successful professions shows the basic need for standard nomenclature, and this is lacking in our field. In particular, the American National Standard Vocabulary is to be renamed as a dictionary; this is quite proper, for it is only a list of defined usage in alphabetical order of the terms. It has no structure, whereas the IFIP/ICC Vocabulary did. Imagine a dictionary for the botanist!

And did you ever see such a sloppy term as "overhead"?

We might start with the primitive of:

Work—Answer-producing
—Answer-validating
Not Work—Scheduling
—Monitoring
—Allocating Resources
—Reporting
—etc.

Another partitioning includes people as well—in a time sequence of software preparation, testing and validation, production runs, and modification. All of these need to have subactivities named and defined more rigorously than at present. The jargon of JCL is incomprehensible to those that use other systems, and vice versa in many cases.

We need standard terminology for the operating system functions—resource management, data management, core compaction, incomplete allocation attempts, waiting, swapping, saving for restart or protection against crash, user validation, etc., etc., so that the smaller functions and program kernels can be assigned to their proper place in the classification structure.

These are the working functions, which would go on whether or not the performance was measured. Similarly, we need good definitions of the monitoring and measuring functions.

3. Reporting

A distinction should be made between the two types of reporting—online for operator intervention and change, and offline (later) for accounting and analysis. Both provide opportunities for performance improvement. The most improvement is likely to be available through providing the operator with sufficient tools, once the operating system has been shaken down somewhat. (I would prefer to see operators of higher caliber than programmers, at least for complex systems, with this reflected in the promotion scale.)

ANSI X3 is very unlikely to achieve a standard for operating systems. There could be some standardization in the subset of reporting activities and their appearance to operators. This might seem unnecessary in the present situation, where programmers change installations with a basic knowledge of some standard programming language, whereas operators scarcely ever do so. But wait until management finds out that some operators have skills, and a feel for tuning a system, that make them far more valuable than any programmer who knows COBOL only.

Accordingly, it is not too early to seek some standards for reporting, by both printed message and analog displays, of resources allocated and used with respect to the individual jobs or batches of jobs. From the crude manometer display on up, more than resource consumption must be reported; contention must also be reported and identified to specific tasks, i.e., resource wastage as well as resource consumption.

4. Software Construction

4.1. Code Independency

All software, whether it be written in high-level or assembly language, should be code-independent from the native character code of the CPU and/or any other code such as the ISO Code (ASCII) and EBCDIC.

The importance of this condition may be judged by the fact that the original 360 software, written without control over such code dependencies, has never been able to be converted to run the 360 as an ASCII-based machine—a feat that the hardware is fully capable of doing.

It may also be judged by an example program in the benchmark tests for the WWMCCS procurement. The source program, although written in COBOL, utilized conditional statements that were operative based upon knowledge of the collating sequence of the EBCDIC (in order to provide these benchmark programs, they were first written for the IBM 360/50, and so tested). The HIS 6000 programmers assumed from the terms of the specifications that ASCII was to be used throughout, and at first could not get correct answers. When a subroutine was inserted to mimic the EBCDIC sequence, there was an 8 percent penalty in running time.

The class of statements that can operate improperly due to code dependency is definable. Source programs may be searched mechanically (by program) for such occurrences, and offending statements at least printed out for manual inspection, if not automatic.

Alternatively, input data to program testing should be given in up to three codes—ASCII, EBCDIC, and the native CPU code if it differs. Such testing should all fall under the Quality Assurance function.

As to public warranty, all software should be certified to auditors, and in advertising, satisfactorily tested for code independency, whenever there is any possibility of portability.

4.2. Frequency of Usage

Software should be so constructed that a frequency count of execution is obtainable, upon demand, for all components. This requires a standard way of identifying such components, and conformance to standards for call and linkage (in hierarchical form, by function).

There should also be provision for count of actual machine instructions during execution of a working program (for the program itself, however, distinct from the operating system, which should have its own count). This provides a "signature" analysis of generated code. In the WWMCCS procurement, a high frequency of single-character moves indicated improper generation of object code. Rewrite resulted in a great improvement in running time.

Frequency of program component execution is quite a different thing from frequency of instruction usage. Both are useful. The latter may be accomplished satisfactorily in a Monte Carlo sense by trapping the instruction in operation at fixed intervals of time. In 600 FORTRAN, this showed that a 4-instruction linkage took up 7 percent of all running time during compilation. Two instructions were cut easily, thus improving 3.5 percent. Over the lifetime of the system, this amounts to several million dollars.

4.3. Computational Accuracy

Results, or answers, are commonly not as accurate as the programmer expects them to be. This is often due to successive operations, truncation, roundoff, basic precision used for both fixed and floating point operations. Use of greater precision should be not only under the control of the programmer, but also as a handle to the operating system. It is conceivable that the programmer should be required to state a value of expected or required accuracy for answers from a computational program segment. The operating system could randomly switch to multiple precision and rerun that segment, with an error message if the difference from the single precision answers exceeds the stated bound.

There should be a standard for floating point computation (in either hardware, firmware, or software) that says: When addition or subtraction of two floating point numbers results in an effective zero because they are of equal magnitude to the precision used, the result shall have a fixed point part of zero, with an exponent part diminished only by the precision of the fixed point part—the exponent shall not be the minimum representable. For old CPU's that do not operate in this manner, all such computations should be interrupted for logging and/or notice to the operator/programmer.

There are many studies in the literature (and the number is accelerating) that show inaccuracies in the common mathematical and business functions that exceed by far the inaccuracies in the normal arithmetic functions. This calls for certification of such functions for specific accuracy within a specific range, with public notice given—for either free or product software, arithmetic, mathematical, or business.

There should be a standard for such programmed function that requires the accuracy, execution time, and storage use to be integral with the function. Then the programmer could call for certain accuracies for general computation, and one of multiple forms for a specific function could be selected to meet (but not overmeet) that requirement.

4.4. The Compilation Process

We take the premise that programs of any significance will be compiled many times prior to successful operation, and many times later for update and modification, and that this process will move to the jurisdiction of other than the originating programmer.

Optimization is often a substantial component of running time, sometimes up to half. Therefore compilers should be constructed so that optimization is selectable.

Virtual storage or not, breaking up a large program into several components for compilation and testing is still good practice.

The compiler should have facility to flag identifiers of fewer than enough characters to make good documentation for other users. Uniqueness is not enough.

Compilers should always produce an updated source program! This should contain at least:

- An imprimatur identifying the compiler used, language features required (or not used), level, and time.
- A statement of the facilities and resources used, running time (either demanded or assigned), etc., for later analysis.
- A concordance of identifiers and statement types used (this may be in hard copy at option).
- A reblocked source program, indented to show nested levels.
- Appended list of mistake messages, if any, or an indicator of successful compilation, as far as the compiler can tell.

5. Documentation

All data on media should be self-descriptive as to format and content, regardless of whether or not it is to be used for interchange. Present labeling standards are insufficient.

It is presently difficult to associate program documentation and run instructions with the program itself, because many programs are kept in punch card form. However, with the full-scale advent of cassettes this condition should be mandatory.

Local documentation, i.e., that associated with the individual operating statements or groups of statements, may be subject to a certain minimum amount of verbiage, else the program may not pass Quality Assurance.

6. Hardware

It is difficult to make many standards for hardware design, for the technology is at a time when virtually anything is possible at a reasonable price, due to microprogramming and chips.

One definite requirement is that all CPU's should have at least two clocks—one continuous and one resettable—both fully available to software.

3. Workshop Discussion

Many participants felt that "standards" could not be set because performance evaluation ideas have not matured adequately. Instead, the term "guideline" was adopted by most people. One of the areas for potential guidelines was accounting data.

Browne: It should be possible to have some guidelines, even if not standards, saying that all systems shall put out the following things on an accounting basis. If it's done right, there should be some minimum guidelines for main-frame vendors and software vendors that solve some of our problems. I think this is a "must." I think we should put some guidelines down suggesting that this is a minimum kind of thing that we ought to be looking for; we'll do better later.

Bell: It seems that the epitome of what we're stranded for is for accounting data, when the systems collect essentially the same data and put it in different formats with slightly different definitions. It's apparently trivially easy to make them coincident. They ought to be coincident so that things can be done in a consistent manner. It's like having tape drives with different size reels.

Browne: There are two points to the problem. They should be receptacles for linear transformation and be consistent.

Bell: I second it.

Boehm: Ok, would somebody state precisely what it is that we're saying ought to be "musts."

Browne: I think we should write some guidelines—we must write some guidelines for minimum content in the accounting system and for a common format for accounting data.

While the need for such guidelines was clear, potential problems were noted by other participants.

Kolence: I'd like to recommend two points that I think are important. One is that along with the type and format of data of be obtained, the capability for the user of such data to obtain other new data is important. In other words, I don't think we could expect our suggestions to serve a fixed set of data that's going to be given for everything. I think it's imperative that we make a resolution open ended to permit other types of data to be collected. In other words, the facilities must be there to collect other data than what we anticipate now. That's point one. Point two is what we were talking about earlier: That integrated instrumentation systems include a minimum set of accounting data and report it well.

Jeffery: You want also to be absolutely sure that what goes into a guideline can use results from a research environment.

NATIONAL BUREAU OF STANDARDS SPECIAL PUBLICATION 406 COMPUTER PERFORMANCE EVALUATION:
REPORT OF THE 1973 NBS/ACM WORKSHOP, MARCH 27–30, 1973, SAN DIEGO, CALIF. (Issued August 1975)

The Weltansicht of Software Physics

Kenneth W. Kolence

Palo Alto, California 94303

This paper is a brief exposition of the idea that a "software physics" exists, and furthermore that it is based on the same concepts as used in the natural sciences. The idea of a software unit is introduced to name the entities embodying the basic observable properties of software physics. These properties are identified as work and time. (Another property, existence, is not referenced in this paper.) The relation of these properties, in a general sense, to the variables of performance monitors and modeling is commented on.

Key words: Computer performance measurement, software physics, software units, software work.

To a large extent, the way one looks upon the basic nature of an object of scientific study determines the form and the power of the subsequent theoretic constructs. Examples of this fact abound in the history of science. It therefore is important to be explicit about one's world view when proposing the development of a field of science. The purpose of this paper is to explicitly consider the world view from which I believe a software physics can be fruitfully constructed.

It is impossible for me to start with the "most important" idea first, since this is one of those situations where several ideas or concepts appear to be of equal importance. In a way, one may also look upon the set of concepts as a set of axioms, some of which may be related or replaced with others to build different logical constructs. But the one concept which appears to justify the use of the terminology "software physics" and which certainly strongly affects the form of the theory arises from the following observation.

There are many fields of human endeavor which call themselves sciences: physics, chemistry, biology, sociology, anthropology, and computer science are but a few. Of these, we observe that those which use the basic "principles" of physics and chemistry make up a conceptually single group, characterized by common terminology and, further, the ability to translate knowledge in one special area to other specialized fields. We tend to call these the "hard sciences" or the "physical sciences." The other group is most singularly characterized by their individual iconoclasm.

Computer science is absolutely unrelated to economic theory, sociology, political science, etc. Indeed, even in that subgrouping of the "soft sciences" which is generally concerned with the study of man and his behavior, little if any interconnection exists in the deep conceptual sense found in the physical sciences. As a means of differentiating between these groupings of sciences, let us divide sciences into "physical sciences" and "singular sciences," where the term singular is used to denote the lack of an underlying conceptual infrastructure between the singular sciences, in opposition to the infrastructure of the physical sciences. (The infrastructure of the physical sciences is well illustrated in Margenau's "The Nature of Physical Reality," McGraw Hill.)

A fundamental choice in one's world-view consists in the (usually implicit) decision to either approach the building of the science from a singular point of view, or within the context of the infrastructure of the physical sciences. An absolutely essential point to understand is that one must make this decision wholeheartedly; either the complete infrastructure is accepted or not. To "borrow" terms and ideas from the physical sciences without acknowledging the full conceptual linkage between all of the fundamental concepts of the physical sciences is to straddle the fence between the physical sciences and a singular science. As the 18th and 19th century Rationalists discovered, the transition probability on that fence is heavily biased toward the singular sciences.

The fundamental choice of software physics is to wholeheartedly accept the full conceptual infrastructure of the physical sciences as the foundation from which to evolve a theory of softwear behavior.

There are good, practical reasons for this choice. If the choice is wrong, we shall be forced into the field of singular sciences relatively rapidly—say, less than a decade. But, we shall know why we are a singular science, and have at least some proof, namely our failure, that we are indeed singular. But, if the choice is right, we will be deeply aided and speeded on our way by the availability of the infrastructure and its intellectual wealth of preciseness and form. Analogy, that most powerful of tools of scientific discovery, is at the same time a most dangerous of seas to venture upon for a scientific quest. It is less dangerous if the full conceptual infrastructure is accepted as one can subject the analogy to at least some critical tests. For singular sciences, the shoals of analogy are uncharted.

Another practical reason for our choice is that our decision permits us to recognize theory when we see it, as opposed to accepting mechanistic descriptions as theory—the bane of the soft sciences. We shall return to this point in more detail later.

If this fundamental choice names software physics, and clearly and cleanly separates it from the field of computer sciences as it is known today, it does so intellectually but, of course, not in terms of the object of study. Yet, the world-view of software physics has yet another important concept which both broadens and simplifies the object of study beyond that of computer sciences: the software unit.

In software physics, the object of our study must be the inherent properties of software, without regard to the arbitrarily selected sizes or packages of code which we name subroutines, tasks, programs, jobs, applications, operating systems, etc. In other words, we must be at least initially concerned with universal properties of software. Each of the aforementioned packages may well have interesting properties in their own right, but the properties of first interest to software physics are those which they all share. A word, a name, is needed to characterize this set of universal properties, and any grouping of code which may be of interest in the context of such properties. The name I have selected is software unit. Thus, whenever in software physics one speaks of a software unit, one is not distinguishing size; rather, one is distinguishing universal properties from properties arising uniquely from the structural, and perhaps functional, choices made during the design process.

The softwear unit plays a role in software physics roughly the same as the center of mass, in its role as a point mass, plays in natural physics. In fact, throughout the natural physics, one deals with equivalent concepts; electrical charge, time, mass, energy, forces, etc. are universal properties associated with matter in some sense, regardless if matter is artificially fashioned into an object or if it is considered in terms of molecules or galaxies. The term software unit is meant to convey the vessel in which similarly universal properties are embodied.

One great advantage of the software unit concept is that the properties of software units are observable to monitors and other forms of instrumentation. In fact, with extremely few exceptions, the observables of computer monitors are only observables of software units. The current basic challenge of software physics is to provide a basic unifying theory relating these observables one to another in meaningful ways. This work has been completed in essence, and is currently being prepared for publication. The work to be published must be experimentally tested before it can be called an accepted theory. At the minimum however, it will represent an example of a theory in software physics. Thus, the essential aspects of the software physics world-view can be summarized by saying that it is believed the basic principles and concepts of the natural sciences will be found to apply to the behavior of the universal properties of software units. It should perhaps be explicitly pointed out that descriptions of software, such as listings, flow-charts, etc., are outside of the current range of interest of software physics.

Software units assume the physical form of electrical and magnetic states within a computing system, and the observables of software physics, such as "CPU busy," are due to the action of software units within the computing system. In simpler words, one measures the effect of a software unit driving a computing system. An interaction thus exists between the workload software unit and the physical configuration of the computing system. This is most obvious when one considers a family of machines, such as the 360 and 370 series. Within a given machine type, say a 360/65, the I/O configuration attached may vary considerably. If one runs an identical program software unit on two or more 360/65's with different I/O configurations, one is apt to observe quite different

I/O measures. Yet the CPU measures as provided by, say PPE, are quite constant. Reversing the conditions and changing main frames up and down the 360 line, one obtains a variation in the CPU measures as well as I/O.

It is perhaps a subtle but important point that the same workload produces different values for the observables. It leads to the question of which observables are independent of configuration and which are at least partially, if not wholly, dependent upon the configuration in which the software unit is physically realized. In my work, the question has an especially simple answer: for a given software unit realized identically on two or more different configurations, the work done by a software unit is independent of configuration, but the times associated with performing that work are dependent on the configuration. Variables, such as power, composed of work and time variables, are dependent on the configuration through time, and independent with respect to work. This, by the way, is an easily testable hypothesis given precise definitions of the terms work and time.

Both work and time are also concepts of the natural physics. In software physics, work and time must be fully equivalent at the conceptual level to these concepts in natural physics if our world-view is to hold. As it turns out, in my studies at least, time has been the more difficult to be precise about. Work however is the key conceptual link between the natural physics and software physics, since it links directly to the concepts of energy and force, and thence on to the remainder of the conceptual infrastructure of the natural sciences. Regardless of the correctness of my own work, I would expect that the concept of work is the key to a demonstratively viable software physics.

Work, in my studies, is said to be done by a software unit whenever a medium is recorded upon, and the amount of work performed is numerically equal to the number of bits acted on. (This means the identity transformation does the same amount of work as a transformation which changes all bits.) In natural physics, work is performed whenever a force acts to change the state of the system under observation. These two definitions are equivalent, with the software unit playing the role of the force, and the media acted upon (e.g., core, registers, magnetic tape or disk, punched cards or paper tape, printer paper, etc.) representing the system under observation. Because of this equivalence, the definition of software work results in the identification of a software unit as a force because of the relationship between these two concepts in the natural sciences. The equivalence also forms a solid link, in my studies at least, between the two physics which will maintain the essential world-view belief that software physics is not a singular science.

Certain implications of the idea of a software physics are meaningful to the practical problems of computer measurement, and others to equally practical problems in the current efforts to analytically model computer systems behavior. The first set of problems are directly addressed by the work currently under preparation. Suffice it to say that most of the questions concerning the meaning and relationships between observables obtained by monitoring are resolved in very simple ways. However, the implications in terms of analytic modeling are not covered in that work, and a few words on the subject are useful here.

Perhaps the most fundamental implication, and one which nicely spotlights the distinction between modeling and theory development, lies in the choice one has as to the variables used in an analytic model. Currently, one normally assumes rather limited "workload distributions," and is completely free to select whatever variables appear appropriate. Because the analytic results one obtains often differ depending on the workload distributions used, and because these distributions are not known to generally occur in practice, the results of most modeling efforts are rather limited in their generality. More to the point, however, the variables selected (e.g., "mean arrival rate," "mean service time," "page fault rate," etc.) are unrelated to fundamental properties of software in general or, in our terms, to software units. Since they are fundamentally "time" variables, they are deeply related to a particular hardware configuration. By itself, this is not bad. What is bad is that they are not related often in a sufficiently analytic fashion to be generally meaningful.

In essence, current modeling efforts are hampered in attaining generality by two factors: no accepted theory exists which identifies the fundamental variables of software behavior, and no general method of characterizing workloads in terms of these variables is available. A theory, or more correctly, a sufficiently powerful theory of software physics should resolve these difficulties. My own work is but a step toward that sufficiently powerful theory, but hopefully it will be of some use in model building by both its world-view and its definitions of software unit work and time.

COMPUTERS AND OUR SOCIETY

R. W. Bemer, Honeywell Information Systems, PHX

NordData - Copenhagen - 1973 August 15-17

INTRODUCTION

In 1950, after my "graveyard shift" at the RAND Corporation, I was still working at 0830 on a 604 board to take an 8-digit square root of an 8-digit number (until then not accomplished mechanically). A round little man approached and asked what I was doing. I told him. He then asked about the calculator, and as I answered each question the next one got more difficult and penetrating, until I was really straining every faculty to answer correspondingly. He did not introduce himself, but I found out later that day that it was John von Neumann.

Naturally the incident remains very clear in my mind. I recall that he did *not* leave me saying "Use the tool well for the social benefit of mankind", or anything else in this vein. There were very few men in the computer world or business then that were considering social ramifications of this sort. Ed Berkeley was, and remains, an exception. To most of us it was just a time of freeing the mind to do far beyond our previous capabilities, at a fantastic rate. We were lured and beguiled; the newness and vast potential drew us, with so much waiting to be done. We took little time for speculation about the eventual effect of computers upon our society, or the extent and scope of the usage to come.

This insensitivity may also have been due to the fact that the first work was almost exclusively concerned with processes upon numbers. Even when I started in 1949, ten years after the first program-controlled calculator was designed, the manipulation of symbols was considered by only a few, and did not even become recognized as a proper computer function until 1956.

Before starting with specifics, let me admit that the title of my talk might have been considered presumptious a decade ago, and perhaps still is by some. Yet I intend to show that there has been a significant change in the type of applications made possible by computers, a change we are ill-prepared for. Any tool that provides leverage or amplification can be misused. I shall give some case histories to demonstrate some ways of misuse and why they continue to be effective. Then I shall outline some measures to reverse the trend and stop much of the misusage.

A CLASSIFICATION OF COMPUTER APPLICATIONS

For purposes of this talk, I propose a simple and perhaps novel classification of computer usage:

CALIFORNIA STATE UNIVERSITY, NORTHRIDGE
Northridge, California 91324

School of Business Administration and Economics

Office of the Dean
(213) 885 - 2456

December 17, 1974

Mr. Robert Bemer
Honeywell Computer Journal
Box 6000
Phoenix, Arizona 85005

Dear Bob:

The enclosed copy of POPULAR COMPUTING contains a condensed report of the session at the Airport Marina Hotel in 1973. I thought you might like to see it.

Cordially,

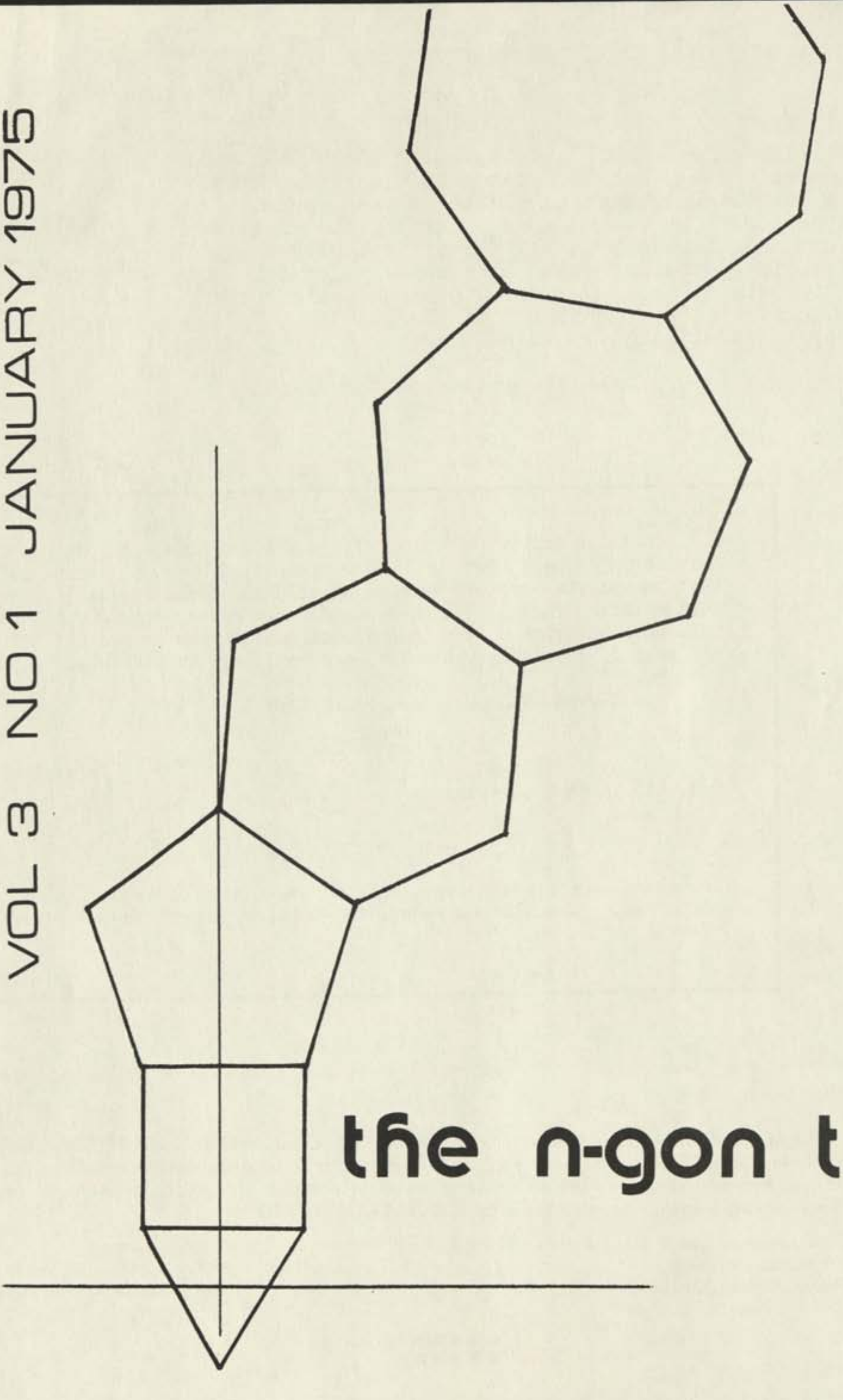
Fred

Fred Gruenberger

FG:kk
Enclosure

Popular Computing 22

VOL 3 NO 1 JANUARY 1975



the n-gon trip

Regular polygons of 3, 4, 5,...,97 sides, each with sides one unit long, are linked together as shown on the cover. The triangle has its center at the origin. For the polygons with an even number of sides, the direction of the chain is straight ahead. For those with an odd number of sides, the direction alternates right and left. Thus, after the 5, 9, 13,... sided polygons, the chain turns slightly to the right; for the 7, 11, 15,... sided polygons, it turns slightly to the left.

PROBLEM **72**

Problem: where will the center of the 97-gon be?

In issue No. 20, a dozen different algorithms were presented for calculating square root. David Ferguson (of Group/3) points out that one of the earliest machine algorithms should be added to the collection. The algorithm (of unknown authorship) dates back to the time when a divide operation on an automatic machine was a frill, and even if available was to be used as little as possible.

The Newton-Raphson scheme is applied to

$$y = 1/x^2 - N = 0$$

and results in the recursion

$$x_{n+1} = .5\, x_n(3 - Nx_n^2).$$

The method converges slowly, but has the virtue of requiring no divisions. When it converges, the required square root is given by Nx.

POPULAR COMPUTING is published monthly at Box 272, Calabasas, California 91302. Subscription rate in the United States is $18 per year, or $15 if remittance accompanies the order. For Canada and Mexico, add $4 per year to the above rates. For all other countries, add $6 per year to the above rates. Back issues $2 each. Copyright 1975 by POPULAR COMPUTING.

symposium 15

Every year since 1958, a one-day discussion session on computing has been held. At the 1973 session, held at the Airport Marina Hotel in Los Angeles, the attendees were:

Paul Armer, Center for Advanced Study in the Behavioral Sciences
Robert Bemer, Honeywell Information Systems
Erich Bloch, IBM
Fred Braddock, Informatics Inc.
Curtis Gerald, California Polytechnic State University,
San Luis Obispo
George Glaser, AFIPS
Irwin Greenwald, Xerox Corporation
Fred Gruenberger, California State University, Northridge
Don Krehbiel, Santa Monica City College
Thomas R. Parkin, Control Data Corporation
Robert Reinstedt, The RAND Corporation

[A copy of the complete transcript of the symposium can be obtained for $10 from the Bureau of Business Services and Research, California State University, Northridge, 91324, the sponsor of the symposium.]

The 15th symposium had the topic "Exploring the Future." A modified Delphi technique was used to try to achieve a consensus on when certain milestones would be passed. For example, the attendees were polled in advance of the meeting for their opinion on the proposition "PL/I will be as dead as ALGOL is (in this country) in 1973; that is, no vendor will boast of offering PL/I as a language." On that particular item, the advance polling indicated a mean of 1988, with a low of 1973 and a high of after the year 2000. At the meeting on December 1, the discussion was aimed at accounting for the wide range of responses, in order to try to reach some agreement. The areas considered included the following:

1. The date when half the computing power of the U.S. would reside in what are now called mini computers.
2. The future of PL/I and APL.
3. The date when the world's chess champion would be a computer program.
4. The date when language translation, from idiomatic language A to idiomatic language B, would be economically feasible by machine.
5. The date when fingerprint recognition would be economically feasible by machine.
6. The date when computing would be a standard school subject in the same sense that algebra is today.
7. The date when more than half the states would require some system of licensing for computists.

As might be expected, no consensus was reached on most of the items. Some excerpts from the discussion are of interest. Bear in mind that the following quotations are taken somewhat out of context.

PARKIN: I look on computers as intelligence amplifiers; as drudgery-grinders; as tools in exactly the same sense as a lathe. Computers do precisely what we tell them to do. They will probably become as pervasive and all-encompassing in our lives as electric energy. I expect that computers will change civilization more than the industrial revolution did. I don't see how we can fault computers for the ills of our government. The 500-odd men in Congress worry about (1) getting reelected, (2) lining their pockets, and–maybe–(3) the country's problems. It's our fault if we don't set up the mechanisms for getting better people in government. You can't blame computers because people chose not to correlate data.

ARMER: One of the weaknesses of the Delphi technique is the difficulty of wording the questions so that they're unambiguous without at the same time revealing what the designer considers the "proper" answer. This is a fine example. The problem of fingerprint recognition has two distinct and widely differing meanings:

(1) Here is my fingerprint. Does it indeed match the
one in your file labelled "Paul Armer"?
(2) Here is a fingerprint. Whose is it?

⇨

GRUENBERGER: I tried to word the item about PL/I very carefully. ALGOL is alive in Europe, but it is totally dead here. It's still available from many vendors, but they don't brag about it. You can't buy a machine on the basis of its ALGOL capability.

GREENWALD: My answer to that question (after the year 2000) assumed that there would be dialects of PL/I that would continue.

GRUENBERGER: Prof. Gerald and I were involved in the current procurement of new computers for the state college system. The committee we were on felt compelled to ask every college department (some 1200 of them) what programming languages they felt they would need in the 1975-1980 time period. There were some 800 responses, listing some 183 languages that someone considered essential to his work through 1980. The list included languages like SOAP and TYDAC, and five or six that no one (on the committee of 15 experts) could even identify. Now, you can't ask for bids on machines and require SOAP, since even IBM couldn't deliver that. The winners in the survey were—surprise—Fortran, COBOL, PL/I, and BASIC, and those are the only ones you can legally ask for anyway. In the same sense as the man who asks for SOAP, PL/I will surely be around in the year 2050 because there will be at least one clown who has to have it.

KREHBIEL: Will whole companies be using PL/I? Will 3-man service bureaus be using it? Will the University of California be using it?

BRADDOCK: There was a survey of some 900 IBM users in which 14% claimed to be using PL/I in some (unstated) way.

KREHBIEL: I'm under the impression that it takes a very large machine to run PL/I, and hence I conclude that only large corporations can use it.

PARKIN: But technology continues to improve. You ought to be able to implement PL/I on an 8K byte machine.

KREHBIEL: But right now it takes a big machine, doesn't it?

PARKIN: Yes, for the particular implementation that exists, but that's not the state of technology.

GREENWALD: The Burroughs 6700 has a design that should lend itself to an efficient PL/I compiler, both for compile time and run time. The point is that PL/I is attractive enough to be cast into hardware, and eventually the compile time will tend toward zero. The same architecture could be cast into smaller machines.

KREHBIEL: But that's some ways away from me. I'm a small user, and I don't rate a 370/167. I'm dealing with a Gremlin that has its tail end chopped off, and the operating system keeps feeling around for that missing piece. Give me PL/I in a 370/115 and I'll start being interested.

PARKIN: There's no real reason why PL/I couldn't be implemented on a mini computer before long.

KREHBIEL: I still don't understand what a mini is. I understand the characteristics of a computer, such as the fact that instructions and data are stored in the same medium and instructions can be treated as data by other instructions. But whenever I ask any vendor anything about "Can your machine do such and such?," the answer is always "yes." So what really differentiates the minis?

GREENWALD: I disagree that minis are going to take over the computing world. There's the question of centralization vs. decentralization. I think there's a big market for both sizes of machine, and I think that centralized computing will increase.

PARKIN: IBM will not discontinue the sale of big machines; there will always be a market for the biggest and most expensive machine. But more and more people are going to question the wisdom of having a super-large machine that is cut up, at great cost, into many little machines, which is what the users see. Technology will eventually produce small packages of computing power (defined any way you wish) accessible and available in clusters to the users. The *number* of minis will far exceed the number of other machines.

Let me try a provocative point. I run an advanced concepts research laboratory. One thing we worry about is the time when the hardware is so cheap that you could essentially give it away and charge only for the system or the software or something else. The cost per bit of storage or of logic element is ever-decreasing, and at a steady rate. It is easy to see ahead to the time when it will be feasible to produce something functionally equivalent to a 6600 in a package the size of a cigarette box, for which the most expensive part is the plug. How will we use the technology at that point? We continue to have dramatic breakthroughs in technology; they're evolutionary

⇨

but still dramatic. Such things drop the cost by an order of magnitude. Sometimes it takes a while before they are observable, but they do happen, and apparently without letup. How are we going to adapt to make use of those breakthroughs? It's this thinking that guided most of my responses. We have to look ahead to the time when bits, and logic elements, and redundancy will be so cheap as to be negligible. That's why I think, for example, that languages will proliferate, rather than die out.

GRUENBERGER: The current game among calculator users is "How many function buttons does your machine have?" Pretty soon it will be "How many words of addressable storage does your machine have?" And sometime after that it will be "How many program steps can your machine hold?" When a pocket machine has a button labelled "standard deviation," a lot of people are going to ask "What is that?" just as millions of people must now be observing that their machine has a button labelled "divide," and up to then they had never had any use for division, much less to 8 significant digits. If nothing else, these new machines are going to have a profound effect on understanding, by masses of people, of esoteric mathematical and scientific concepts.

GREENWALD: Will masses of people be able to deal with concepts like storage, sequencing, and complicated functions?

BEMER: Look what APL has done. The people who become familiar with APL think in terms of its functions, which are very powerful. They just naturally think at a much higher level. It may be that we can someday teach kids to start thinking at a higher level of abstraction.

GLASER: You're grossly underrating the customers. I know of many installations where the DP manager knows his business, and his management knows where the money goes. These men have stature, and common sense, and political clout. It's not universal, and it may never be; you can't stamp out idiocy. But I'm encouraged by what I see. The level of review committees is high, and by and large they're smart.

GREENWALD: Much of this management awareness and know-how was generated during the 1970 recession. Perhaps a 1974 recession will increase their awareness.

GLASER: I agree. People don't learn from an executive course or from a Fortran manual; managers learn when the Profit and Loss statement comes out. Along these lines, I'm a very strong advocate of charge back systems; I want the user to pay every nickel of the costs. There are exceptions, of course, but I know that with proper charge back, the quality of the work goes up and its reception is assured; everything gets better. It's painful, I know, since companies can say "This isn't our normal procedure; we don't charge for accounting services, for example." But accounting isn't discretionary, and DP systems should be, and when they're not, the chances of failure go 'way up.

ARMER: You're saying that you want feedback in a system.

GLASER: Yes, it's sharp pointed negative feedback, almost to the point of being punitive, but it has the right effect.

ARMER: If we assume constant productivity of systems programmers, and the demand increases, then what? Will the demand go up faster than productivity?

PARKIN: The cost of the hardware keeps going down. I predict that the demand for systems programming is going to go up, rapidly.

BEMER: The monetary feedback information will operate, when people observe that the systems people cost a fantastic amount relative to the hardware. To reduce those costs, people will turn to automated techniques for software.

PARKIN: Not in my lifetime.

ARMER: I wonder whether the hope for significant improvement in productivity isn't akin to the same hopes for machine translation, or machine chess.

BLOCH: No, it's a different kind of problem, and one that lends itself to new techniques. For example, we know how to apply engineering techniques to the production of software.

GREENWALD: But in the IBM studies, for example, it turns out that if you could double the amount of time actually spent on writing programs (versus everything else the programmer does), you'd still be under 2%.

BEMER: Let me put it this way. Programming is a tricky thought process. The tie-up comes (with long turnaround times) in getting back in context with those tricky thought processes. Just by shortening the turnaround time (to nearly zero), the programmer stays in context and productivity goes up.

⇨

GREENWALD: And all our tools have enabled us to go, in systems programming, from 30 checked out instructions per day down to 5 to 7 per day.

GRUENBERGER: In the scientific area, we have done certain problems once and for all; for example, the solution of simultaneous equations, or gear design, or Bessel function calculations. Isn't there a corresponding body of systems software problems that have been solved, so that each man doesn't have to solve them all over again? Doesn't the building block principle apply here, too?

BEMER: It's more difficult. You might like a packaged tax routine that could be plugged into any program that deals with taxes, but the tax laws are too varied to permit it.

GERALD: But couldn't we create tax modules, that could be parameterized and then collected to fit specific situations?

BRADDOCK: It depends, of course, on how you define systems software. We've all dealt with I/O instructions that deal directly with the peripheral devices. But today's systems programmers don't do that; they don't even know how tape or disk drives actually work, and they don't care. Their level of expertise is much different from that of systems programmers of ten years ago. A lot of people can turn out code in assembly language or Fortran or COBOL, but that doesn't make them systems programmers. We have developed a cadre of competent people who know their jobs, and they are developing the tools (or modules) that everyone else can use. One shouldn't generalize, but to my way of thinking, anyone who writes in Fortran is not a systems programmer; they are applications programmers getting a job done. We'll need a lot more of those.

BLOCH: I can't see what bearing the choice of language has on the matter. If he designs a system and uses Fortran, he's a systems programmer.

GREENWALD: Let's eliminate the semantic problem here. If he writes an operating system, or a language translator, he's a systems programmer and Braddock says there will be less such people. If he *uses* the product of a systems programmer, he's an applications programmer, and Braddock says there will be more such people.

PARKIN: I keep pointing out that the hardware is going to the point where we can give it away, and all we'll have left to sell will *be* systems.

GRUENBERGER: Tell me what I should tell my students (those who are headed toward careers in computing). Do I tell them that after 7 years or so they will be at peak salary unless they go into management?

GLASER: Yes, unless they pick up some merit badges along the way, such as knowledge of production control, or accounting systems, or manufacturing control, or go from sales to statistics to market research.

REINSTEDT: In other words, he must keep himself adaptable, and mobile, rather than narrow.

BRADDOCK: From management's point of view, a man should seek knowledge and constantly improve himself. The big trouble is that most people acquire only that knowledge that is essential to the project they've been assigned to. My big gripe is the man who is immersed in data base work (having been assigned to that task) who remains ignorant of another area (e.g., communications) which he should know about.

REINSTEDT: Here's another example. At one time, linear programming was a big thing. If we had five programmers whose specialty was linear programming, and they had learned nothing else, then they'd all be in trouble now, because linear programming just isn't in demand.

GREENWALD: We're being unfair. A person gets involved with a specific area, like linear programming, because that was the work he was assigned to. When a new problem in that area comes along, he gets it because he's the expert in it. And as long as he's involved with his specialty, we expect him to work at it, and we're not apt to encourage him to be studying other areas. I doubt that that will change.

BLOCH: That's true for a drill press operator, but a professional man has a responsibility to keep himself informed, at least, about other areas.

BEMER: Part of the problem is caused by the people themselves. The tenure in a particular assignment could be halved (say, three years writing Fortran compilers instead of six years at it) if they would learn to document what they had done so they could move on.

▷

PARKIN: Fifteen years ago everyone in our field had a feeling of great excitement at being involved with this new high order of intellectual activity. Everyone could see years ahead of interesting new problems and applications, and everyone was learning at high speed. Today, that feeling seems to be gone. I am appalled at the 25 and 30-year-old people who have stopped learning; who say, in effect "I've learned the trade; I'm an expert; I don't need to learn anything else." They keep going at that level, and they're hacks. What apalls me is how the hack level is appearing at earlier and earlier ages. Maybe it's the "they aren't raising kids like they used to" syndrome.

GREENWALD: Those of us in this room all learned by experience, since that was the only way possible then. We all did everything. But today we can get in a young man who gets assigned to SYSGEN work, and pretty soon he's the local expert and can't be spared for anything else. He could quit and go somewhere else, but he can't get reassigned within his company; he's stuck. Even if he tries for reassignment, we always have deadlines to meet, and we seem to be better off letting him be stuck.

GRUENBERGER: I was startled by the responses to our question about the certification or licensing of programmers. Nearly everyone said "We aren't going to do that." I think we are, and that it may be forced on us in ways we won't like. Perhaps we could define the problem better by making an analogy to the mechanisms for the CPA. The rules for that—the avenues toward getting it—and the enforcement procedures—are all laid out, and they work, and they have been quite stable for over 25 years.

BLOCH: But the technology of the CPA has been the same for 300 years.

GRUENBERGER: Don't believe it. The accounting world changes pretty fast. The changes are not as fast as in our business (and they are far more orderly) but they are first order effects. For example, about 10 years ago they sent a CPA to jail, telling him "You should have known," and not accepting his plea that he didn't know of the shenanigans that were taking place in the firm he was auditing. For 25,000 CPA's in the country, the ball game changed its rules overnight. More recently, we've had Equity Funding, which will cause even more changes.

GLASER: Going back to computing, are the objections to certification and licensing due to a belief that we can't do it right, or that we shouldn't do it?

REINSTEDT: My position is that we can't possibly do it right (but that we're going to do it).

GLASER: If that's true, and it comes about anyway, what will happen? Will we find ourselfes with a lot of people who are certified but incompetent?

GRUENBERGER: Can we agree that the program has worked for the CPA's?

REINSTEDT: They are not all equal, but I get a distinct feeling of what constitutes a CPA, and I think most of us do. But try to extend that same notion to programmers.

BRADDOCK: An analogy with doctors may be appropriate. There is probably a written examination for them, but the real test is their apprenticeship, which goes on for several years. We will face the same problem, and our solution should probably be the same; namely, a long apprenticeship.

GRUENBERGER: I used the word "programmer" only in the catch phrase "Certified Public Programmer," but the question relates to certification of computer people in general. We should be asking, can a man be certified as knowledgable about computers and their uses?

ARMER: For whom will such people work? Would they work for firms that send a man in to certify another firm's programs? In other words, would they function the way CPA's do?

GREENWALD: Companies hire accountants and they hire programmers. They can get a certified accountant if they wish, or they can also get one who is not certified. They could do the same thing with programmers.

GLASER: The CPA certificate has motivated a lot of people to try to reach a stated level of knowledge. It has done a lot for the accounting profession. True, a man crams to pass that set of exams, but it's unfair to conclude that he then stops learning.

REINSTEDT: I'm all for motivating people to learn more and upgrade themselves. But when you take the tests and get the certificate, what are you then certified to do?

GLASER: Well, it's much like requiring a Boy Scout to take a 50 mile hike. It won't guarantee his ability to survive in the woods, but it's evidence of some level of capability, and several such requirements put him ahead of the boy who hasn't done them. As things stand now, you have no evidence at all from anyone who walks in the door and says "I'm a programmer."

⇨

ARMER: I require that the guy I hire has a college degree. It's not that the degree has given him anything specific, but simply that the probability of finding a good man in that population is much higher than that of finding a good man in the non-degree population. The degree is a sifting device, and the certificate could serve the same purpose.

GLASER: The Harvard Law School graduate may not be better trained that the graduate of Podunk, but statistically he's a better bet. If nothing else, his survival ability is better.

REINSTEDT: But don't tell me he's certified.

GLASER: Not as an individual. But in hiring him, your risk is lower if that's all you know.

REINSTEDT: Then the term "certified" is a misnomer; worse, it's a non sequitur.

KREHBIEL: It is any worse than what we expect from a man who can call himself a lawyer?

REINSTEDT: When I go to a lawyer, I know what I can expect from him.

PARKIN: You do? You must be as ignorant about law as most of us are about medicine, then.

REINSTEDT: But what's the alternative? Given a legal problem, I must go to a lawyer, and I know what to expect from him.

GREENWALD: Isn't all this just a substitute for a programming aptitude test? Those were designed to save personnel departments some time and effort.

PARKIN: They turn out to be only IQ tests.

REINSTEDT: Not "turn out to be"; they were *taken* from IQ tests.

GLASER: I think the present DPMA tests are better, for their numbers, than any of us would acknowledge. Clearly, those tests do not apply to numerical analysts, or scientific programmers, or the artificial intelligence boys; the tests just don't apply.

REINSTEDT: In analyzing the results of the last DPMA exams, they broke out those who were taking the test for the first time. Those who had majored in data processing in college came in second from the last (next to accountants) and under education majors, math majors, engineers, and everyone else, on the first two parts of the test. For the other parts, they were on the bottom.

GLASER: Sure; they learned DP from numerical analysts and mathematicians. They didn't learn from people who had practical experience in the DP world.

GRUENBERGER: All this is charming, but totally irrelevant to the question, which was When will half the states require some sort of certificate?—good, bad, or indifferent. You guys are all busy designing the perfect certificate, which isn't the point. It seems to me that if we have two more Equity Funding scandals within six months of each other, then about two months later more than half the states will require licensing of computer people, and they won't care how good it is.

PARKIN: A lot of doctors have killed their patients, but that is not the mechanism that led to the medical examining boards we now have. The medical profession decided to police itself, and quietly keep its mistakes from the public view.

GRUENBERGER: That only supports my statement. We ought to keep our mistakes to ourselves, too, and act to do it before it's forced on us.

GREENWALD: Us old people might have to protect ourselves from the young people.

KREHBIEL: Then you go on to restrict entry into the field, and you add grandfather clauses (in our case, literally).

GLASER: We joke about it, but in five years or so, the economic pressure on the 45-year-olds will be strong enough to make that more likely to happen than not.

REINSTEDT: I guess the answer to the question is that we will have certified programmers pretty soon, and it will be meaningless.

GRUENBERGER: The question was *when*?

BRADDOCK: I voted for a late year, when it might mean something.

GREENWALD: I would now vote much earlier. The politicians will say "We recognize the problem, and we have done something about it."

KREHBIEL: If it takes them as long to recognize this problem as it did the oil shortage, we have a lot of time.

GLASER: The people in DPMA who run the certification program know that it isn't as good as it ought to be; that it needs fixing; and that they acknowledge that it needs fixing. Few people would defend it as the ultimate.

BEMER: My motto in computing has always been these five words: *Do something small useful now.*

REINSTEDT: The certification boys are about to do something large useless now.

□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□

N-SERIES 22

Log 22	1.342422680822206235963938865967517268474892071928562
Ln 22	3.091042453358315853479175699423305867897206988297672
$\sqrt{22}$	4.690415759823429554565630113544466280588228353411737
$\sqrt[3]{22}$	2.802039330655387120665677385665894017585798218769268
$\sqrt[5]{22}$	1.855600736258084334732770521321674794599490537151919
$\sqrt[7]{22}$	1.555158536763463318507348720266186388028707231984833
$\sqrt[10]{22}$	1.362204366553743041822749217702391289113218086364251
$\sqrt[100]{22}$	1.031393112229484533249772315571564116212442930070430
e^{22}	3584912846.131591561681159945978420689222693065037274 934931971841555786919360975329001975957054
π^{22}	86556004191.98134152251135804670373660115625743426539 53850048107098781181657147912973613134003
$\tan^{-1}$ 22	1.525373047373319604208237289205537102419159594135981
22^{100}	17469001504088245598835440079017000089716288685957915 23527044718998741634271162410507673817696314640559387 34482112142486751362076901376

The figures on this page are the eleven possible ways in which four pentagons can be joined at their edges.

PC22-10

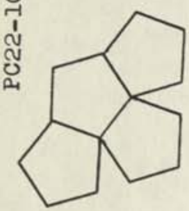
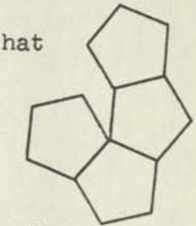
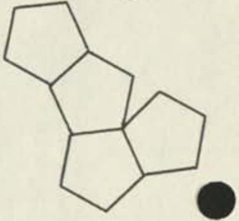


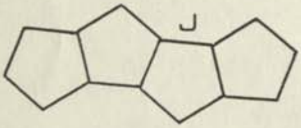
Table H shows the present state of knowledge about such polyominoes made up of squares, triangles, and hexagons. The table is furnished by Thomas R. Parkin, of Control Data Corporation, who first calculated the values for squares, up to case 15. The values for 16, 17, and 18 were calculated by Prof. W. Fred Lunnon for his PhD thesis.

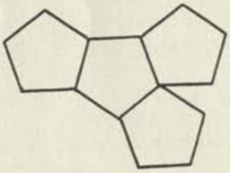
Work on polyominoes has been done only for squares, triangles, and hexagons, because those are the polygons that can tile the plane. Pentagons are a different animal.

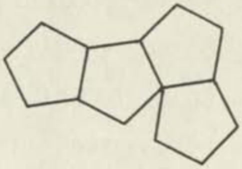


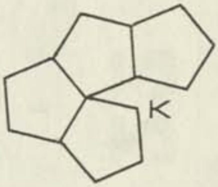


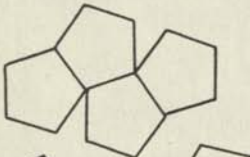
To find out how many pentagonal polyominoes there are for case 5, one could follow this straightforward algorithm: append a pentagon to every possible side of the shapes on this page, and then eliminate the duplicates from the resulting set of figures. Both parts of that algorithm might be difficult to apply. For example, there are clearly 14 places where the next pentagon can be appended to Figure J, but it is not immediately clear how many can be appended to Figure K. As Mr. Parkin points out, "Unfortunately, it requires trigonometry to know if a pentagon can be added to some figures in particular places. Thus, the growth of figures as N increases becomes a question of how accurately one can compute distances and, since trigonometric functions are transcendental, there is no precise integer answer."

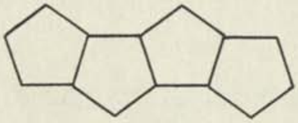


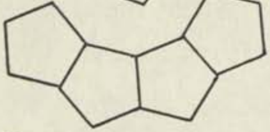


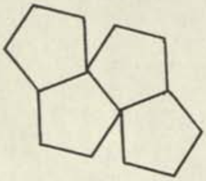












Note: the counts in Table H are for the free shapes, as noted, but the polyominoes on this page are of fixed shapes; that is, left and right versions of the same shape are both shown.

Using either fixed or free shapes, the Problem is to extend Table H in the column for pentagons.

Pentagonal Polyominoes

PC22-11

Table of known information on polyominoes

N	Squares	Triangles	Hexagons	(Pentagons)
1	1	1	1	1
2	1	1	1	1
3	2	1	3	2
4	5	3	7	7
5	12	4	22	
6	35	12	82	
7	108	24	333	
8	369	66	1448	
9	1285	160	6572	
10	4655	448	30,490	
11	17,073	1186	143,552	
12	63,600	3334	683,101	
13	238,591	9235		
14	901,971	26,166		
15	3,426,576	73,983		
16	13,079,255	211,297		
17	50,107,911			
18	192,622,052			

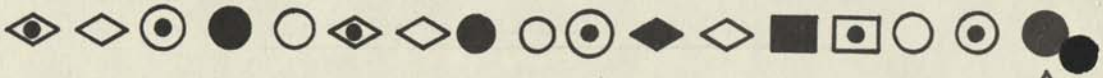
H

Note: These counts are for the free shapes; i.e., those which are free to rotate and reflect in the plane.

●●●●●●●● ○○○○○○○○
●●●●●●●● ○○○○○○○○

The two-dimensional Flagstone Problem

2							
1							
3							
2	3						
3	2	1					
1	3	2	1				
3	2	1	3	1	2		
2	1	3	2	3	1	2	3



The border pattern shows a solution to the Flagstone Problem:

> A man has flagstones of three different colors. How can he lay them so that no pattern of colors is immediately repeated; that is, so that no consecutive pair of stones has the same color; no consecutive pairs have the same colors in the same order; no three stones show the same sequence of colors as the preceding three; and so on for any sets of N stones.

(The shapes in the border design have no meaning; the three colors are represented by white, black, and centered dot.)

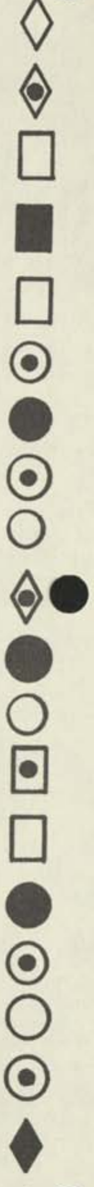
The Flagstone Problem is number F13 in the book Problems for Computer Solution (Gruenberger and Jaffray). It is Problem 33 of the fifth book of Problematical Recreations, Litton Industries. It first appeared in "Unending Chess, Symbolic Dynamics and Problems in Semigroups," Marston Morse and Gustav Hedlund, in the Duke Mathematical Journal, Vol. 11, March, 1944. In mathematical terms, as the Flagstone Problem, it appeared in the problem section of the American Mathematical Monthly, submitted by Hugh Noland, June-July 1963. A solution by C. H. Braunholtz appeared in the same issue.

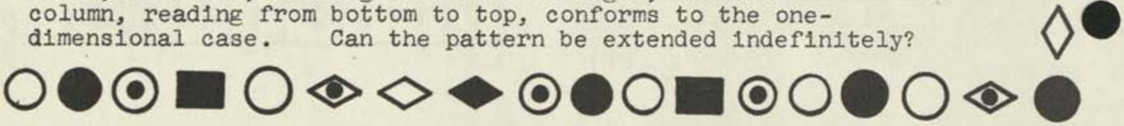
The problem makes an interesting exercise in computer logic. Consider this sequence:

123213231232123132312131232132312321231323121
↑

and the logic of extending it. The next digit cannot be a 1. It also cannot be a 2, since that would repeat the 2-digit sequence 1212. But it also cannot be a 3, since that would repeat the 23-digit sequence starting at the beginning. Therefore, it is necessary to back off and change the last given digit from 1 to 3, and then proceed forward again. It may be necessary to back off many digits.

Braunholtz showed a method of constructing such sequences of any length, so the Flagstone Problem remains only as an exercise in computer coding. The task at hand is to extend the problem to two dimensions: in the array shown, each row, reading from left to right, and each column, reading from bottom to top, conforms to the one-dimensional case. Can the pattern be extended indefinitely?





Speaking of Languages

ROBERT TEAGUE

In the last issue (PC21-12) we discussed some of the workings of the PLATO system for CAI and also its unique terminal. This month, I would like to turn our attention to TUTOR, the language that makes PLATO work.

Modes of operation. There are three modes of operation within the system: (1) system mode; (2) author mode; and (3) student mode. Each of these modes is available to a user only when he has the right access code in his user number. They are downward inclusive; that is, someone in system mode can work in author mode or student mode if he desires, but not vice versa.

The system mode allows a programmer to make changes to the PLATO system itself. This is possible because PLATO is written in TUTOR just as any instructor-prepared course material would be. (One of the old criteria of a good language was whether or not the language compiler could be written in the language. Here the answer is, yes it can and is.)

Author mode is the necessary mode of operation for an instructor (or any user) to be in to create new materials for the system. In this mode he can create, edit, and execute the courses he is developing for the system.

Student mode can only be used to run existing course materials. However, student users normally have priority to use of the system and can frequently get on when an author cannot.

Program structure. The "program" (the term isn't used) in TUTOR is called a course. Every student or author must be listed as being enrolled in each of the courses on the system in order to access those materials. The course is broken down in two ways: (1) in a physical breakdown into blocks, or (2) in a logical breakdown into lessons. A block is given a name by which changes to it are made, and may contain one or more lessons. The lesson consists of the materials to be presented upon a subject. A lesson could, for example, be written on a topic in chemistry, or accounting, or any other discipline. Within each lesson are one or more units. Units comprise the materials to be presented on the display screen at one time, and since they are named, are the logical transfer points in the lesson.

The language commands. The TUTOR language commands have two parts: the command and the tag. The command gives the operation to be performed, while the tag gives various information depending on the command. For example, a "write" (all TUTOR commands are given in lower case) command would have a tag giving the information to be displayed on the screen, while a "jump" would have the name of the unit to which the transfer will be made. Although not exhaustive, the accompanying table gives all the commands needed to write complete course materials in any field.

The lesson has two states of operation that must be considered before going into the actual commands. Normal state will simply execute the display, computation, or utility commands in sequence. Once the "arrow" command is used to elicit a response from the student, however, the state will be shifted to judging state. In this state, a judging command must be used before the three other types of commands will have effect again (this is called "satisfying the arrow"). For example, in the code

```
arrow   1510
answer  4
write   Correct.  Very good.
answer  5
write   Close enough, but it's really 4.
wrong   6
write   Try again.
```

if the input for the "arrow" is 4, the first write will be executed and the next four lines of code will be skipped. If the answer were 5, the second write is executed and the next two lines skipped. But if the input were 6, "Try again." would be displayed and the system would automatically (because the "wrong" command would judge the response 6 as incorrect) go back to the "arrow" command to elicit another response. Because of these states and the way they operate, TUTOR is not actually a totally sequential language. The list of "answer" and "wrong" commands acts like a jump vector, with the one matching the input being selected to execute next. This situation occurs in a few other places as well.

Next month we will go into the commands with sample lessons written in TUTOR, to give more feel for the capabilities of the language.

Table of TUTOR Commands

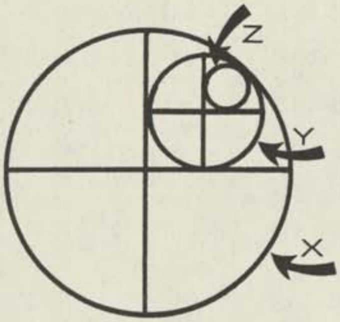
Display	Judging	Computation	Utility
at	arrow	calc	unit
write	answer	randu	next
erase	wrong	define	jump
draw	no	addl	pause
circle	ok	subl	do
show			
ansv			
wrongv			
size			
long			

□

In a circle of unit radius, another circle is drawn in one quadrant, tangent to the quadrant lines and the original circle. In this new circle, the process is repeated; that is, the inner circle is quartered and a new circle is drawn in one of the quarters as before.

The original circle (X in the figure) has an area of pi square units. What is the total area occupied by the infinite sequence of smaller circles? (Circle Y has an area of about .54 square units; circle Z has an area of about .09 square units; thus the total area occupied by just those two circles is about .63 square units.) The summing process should begin with circle Y.

PC22-15



Nested Circles

PROBLEM **75**

In the Check Writing Problem (Problem L2 in Problems for Computer Solution, Gruenberger and Jaffray, Wiley, 1965), amounts up to $499.99 for a check are to be translated into words, as for example:

FOUR HUNDRED NINETY NINE AND 99/100.

Considering only whole dollar amounts, starting with 1, what is the first appearance of each amount that requires more space on the check? If a check protection symbol (**) is printed just to the left of the dollar amount, for what amounts will that symbol first move further to the left? In other words, extend this list:

Check Protection

PROBLEM **76**

```
                **ONE
              **THREE
             **ELEVEN
           **THIRTEEN
          **SEVENTEEN
         **TWENTY ONE
       **TWENTY THREE
      **SEVENTY THREE
    **ONE HUNDRED ONE
  **ONE HUNDRED THREE
 **ONE HUNDRED ELEVEN
```

PC22-16

THE FIRST 1000 ODD PRIME NUMBERS

3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59
61	67	71	73	79	83	89	97	101	103	107	109	113	127	131	137
139	149	151	157	163	167	173	179	181	191	193	197	199	211	223	227
229	233	239	241	251	257	263	269	271	277	281	283	293	307	311	313
317	331	337	347	349	353	359	367	373	379	383	389	397	401	409	419
421	431	433	439	443	449	457	461	463	467	479	487	491	499	503	509
521	523	541	547	557	563	569	571	577	587	593	599	601	607	613	617
619	631	641	643	647	653	659	661	673	677	683	691	701	709	719	727
733	739	743	751	757	761	769	773	787	797	809	811	821	823	827	829
839	853	857	859	863	877	881	883	887	907	911	919	929	937	941	947
953	967	971	977	983	991	997	1009	1013	1019	1021	1031	1033	1039	1049	1051
1061	1063	1069	1087	1091	1093	1097	1103	1109	1117	1123	1129	1151	1153	1163	1171
1181	1187	1193	1201	1213	1217	1223	1229	1231	1237	1249	1259	1277	1279	1283	1289
1291	1297	1301	1303	1307	1319	1321	1327	1361	1367	1373	1381	1399	1409	1423	1427
1429	1433	1439	1447	1451	1453	1459	1471	1481	1483	1487	1489	1493	1499	1511	1523
1531	1543	1549	1553	1559	1567	1571	1579	1583	1597	1601	1607	1609	1613	1619	1621
1627	1637	1657	1663	1667	1669	1693	1697	1699	1709	1721	1723	1733	1741	1747	1753
1759	1777	1783	1787	1789	1801	1811	1823	1831	1847	1861	1867	1871	1873	1877	1879
1889	1901	1907	1913	1931	1933	1949	1951	1973	1979	1987	1993	1997	1999	2003	2011
2017	2027	2029	2039	2053	2063	2069	2081	2083	2087	2089	2099	2111	2113	2129	2131
2137	2141	2143	2153	2161	2179	2203	2207	2213	2221	2237	2239	2243	2251	2267	2269
2273	2281	2287	2293	2297	2309	2311	2333	2339	2341	2347	2351	2357	2371	2377	2381
2383	2389	2393	2399	2411	2417	2423	2437	2441	2447	2459	2467	2473	2477	2503	2521
2531	2539	2543	2549	2551	2557	2579	2591	2593	2609	2617	2621	2633	2647	2657	2659
2663	2671	2677	2683	2687	2689	2693	2699	2707	2711	2713	2719	2729	2731	2741	2749
2753	2767	2777	2789	2791	2797	2801	2803	2819	2833	2837	2843	2851	2857	2861	2879
2887	2897	2903	2909	2917	2927	2939	2953	2957	2963	2969	2971	2999	3001	3011	3019
3023	3037	3041	3049	3061	3067	3079	3083	3089	3109	3119	3121	3137	3163	3167	3169
3181	3187	3191	3203	3209	3217	3221	3229	3251	3253	3257	3259	3271	3299	3301	3307
3313	3319	3323	3329	3331	3343	3347	3359	3361	3371	3373	3389	3391	3407	3413	3433
3449	3457	3461	3463	3467	3469	3491	3499	3511	3517	3527	3529	3533	3539	3541	3547
3557	3559	3571	3581	3583	3593	3607	3613	3617	3623	3631	3637	3643	3659	3671	3673
3677	3691	3697	3701	3709	3719	3727	3733	3739	3761	3767	3769	3779	3793	3797	3803
3821	3823	3833	3847	3851	3853	3863	3877	3881	3889	3907	3911	3917	3919	3923	3929
3931	3943	3947	3967	3989	4001	4003	4007	4013	4019	4021	4027	4049	4051	4057	4073
4079	4091	4093	4099	4111	4127	4129	4133	4139	4153	4157	4159	4177	4201	4211	4217
4219	4229	4231	4241	4243	4253	4259	4261	4271	4273	4283	4289	4297	4327	4337	4339
4349	4357	4363	4373	4391	4397	4409	4421	4423	4441	4447	4451	4457	4463	4481	4483
4493	4507	4513	4517	4519	4523	4547	4549	4561	4567	4583	4591	4597	4603	4621	4637
4639	4643	4649	4651	4657	4663	4673	4679	4691	4703	4721	4723	4729	4733	4751	4759
4783	4787	4789	4793	4799	4801	4813	4817	4831	4861	4871	4877	4889	4903	4909	4919
4931	4933	4937	4943	4951	4957	4967	4969	4973	4987	4993	4999	5003	5009	5011	5021
5023	5039	5051	5059	5077	5081	5087	5099	5101	5107	5113	5119	5147	5153	5167	5171
5179	5189	5197	5209	5227	5231	5233	5237	5261	5273	5279	5281	5297	5303	5309	5323
5333	5347	5351	5381	5387	5393	5399	5407	5413	5417	5419	5431	5437	5441	5443	5449
5471	5477	5479	5483	5501	5503	5507	5519	5521	5527	5531	5557	5563	5569	5573	5581
5591	5623	5639	5641	5647	5651	5653	5657	5659	5669	5683	5689	5693	5701	5711	5717
5737	5741	5743	5749	5779	5783	5791	5801	5807	5813	5821	5827	5839	5843	5849	5851
5857	5861	5867	5869	5879	5881	5897	5903	5923	5927	5939	5953	5981	5987	6007	6011
6029	6037	6043	6047	6053	6067	6073	6079	6089	6091	6101	6113	6121	6131	6133	6143
6151	6163	6173	6197	6199	6203	6211	6217	6221	6229	6247	6257	6263	6269	6271	6277
6287	6299	6301	6311	6317	6323	6329	6337	6343	6353	6359	6361	6367	6373	6379	6389
6397	6421	6427	6449	6451	6469	6473	6481	6491	6521	6529	6547	6551	6553	6563	6569
6571	6577	6581	6599	6607	6619	6637	6653	6659	6661	6673	6679	6689	6691	6701	6703
6709	6719	6733	6737	6761	6763	6779	6781	6791	6793	6803	6823	6827	6829	6833	6841
6857	6863	6869	6871	6883	6899	6907	6911	6917	6947	6949	6959	6961	6967	6971	6977
6983	6991	6997	7001	7013	7019	7027	7039	7043	7057	7069	7079	7103	7109	7121	7127
7129	7151	7159	7177	7187	7193	7207	7211	7213	7219	7229	7237	7243	7247	7253	7283
7297	7307	7309	7321	7331	7333	7349	7351	7369	7393	7411	7417	7433	7451	7457	7459
7477	7481	7487	7489	7499	7507	7517	7523	7529	7537	7541	7547	7549	7559	7561	7573
7577	7583	7589	7591	7603	7607	7621	7639	7643	7649	7669	7673	7681	7687	7691	7699
7703	7717	7723	7727	7741	7753	7757	7759	7789	7793	7817	7823	7829	7841	7853	7867
7873	7877	7879	7883	7901	7907	7919	7927								

Those underlined mark the first appearance of each larger difference between successive primes. For example, the first appearance of a difference of 18 occurs between 523 and 541.

Printed by Argold Press Inc.
16921 Ventura Boulevard, Encino

REPORT

15th Annual One-Day Computing Symposium

December 1, 1973

Exploring the Future

by
Fred Gruenberger

Bureau of Business Services and Research
California State University, Northridge 91324

January, 1974

15th Annual One-Day Computing Symposium

Airport Marina Hotel, Los Angeles, December 1, 1973

EXPLORING THE FUTURE

The attendees listed in Appendix A convened at the Airport Marina Hotel. In September, each invitee was sent a questionnaire, shown in Appendix B, and in November the distribution of the responses (Appendix D) was sent to them. Additional agenda items for the day's discussion had accumulated (Appendix C).

GRUENBERGER: You all received copies of the distribution of replies to our miniature Delphi questionnaire. Roger Mills' approach was quite simple; he simply has everything happening in 1980. But for the rest of us, the responses cover quite a wide range, and one of our tasks today is to see if we can't narrow that range. Some of the items aren't even very controversial, and we ought to be able to converge closer than a 30-year estimate of when they will take place.

GREENWALD: How do you arrive at a mean of 1988 (as in the PL/I question) when the high is "never"?

GRUENBERGER: I weighted the "never" and "after 2000" as 2010, on the grounds that, in our lifetimes, 2010 is virtually never.

Our first order of business is to order the agenda; to determine what we want to discuss. Professor Gerald added the topic of how to make hardware and software responsive to the needs of the users.

BEMER: We have just witnessed what must be one of the greatest failures of the use of computers; I refer to the energy crisis. All the oil companies are big users of computers and they must have huge data bases. We know that they're good public servants, because they tell us so in the press and on TV. But apparently no one was aware of the impending gasoline shortage. Have we given up modeling and simulating?

GRUENBERGER: It wasn't more than six or eight months ago that I was still being urged to use more electricity.

BEMER: Sure. General Electric was advertising about how good it was to light up everything; how much better it was to play football at night, and so on. Can't computers tell us these things? What happened to the notion of using computers to predict trends?

GLASER: That only testifies as to how decisions are made, which has little or nothing to do with computers.

BEMER: But maybe it should have. Perhaps we should insist that information that could lead to decisions be made freely available so that people could use it. The way it has been, only the man who makes the decisions has access to the facts, but maybe that's not right.

GERALD: But in the case of the energy crisis, did anyone ask the proper questions? You're saying that the data should be made more generally available, but if it were, would there be someone to analyze it properly? Did you?

BEMER: Yes, because I know of someone who had compiled data on the energy crisis, and now everyone depends on his data.

GRUENBERGER: I have to admit that I was taken by surprise, because I was brainwashed when I was at the Hanford project. They told me then, and I believed it, that electric power was virtually unlimited; that we could never use all the power we could generate. I wasn't asking any questions last year. In fact, even if I had, the electric power companies were all telling us that a few nuclear generators here and there would solve all problems.

PARKIN: If you look over the literature of the last few years, you'll find quite a few hints that there was a pending energy shortage. Some of these were gentle pokes at the environmentalists who were opposing the Alaskan pipeline and offshore drilling, but basically the warnings were there, quite clearly.

GLASER: It wasn't socially acceptable to agree with the warnings. In a sense, you had to be against clean air in order to be in favor of increased energy production.

ARMER: You wouldn't have to be against clean air to warn of an impending shortage.

*: What does all this have to do with computing?

BEMER: All the pertinent data should be in the hands of the government, but even if it is, it can't be collated and brought together. All the oil companies know the size of their reserves, and their sales figures. Computer people ought to be able to help make some of the decisions.

GREENWALD: There have been published figures that indicate that the current rate of U.S. oil consumption would deplete all the known reserves of Saudi Arabia in 20 years. Similarly for other reserves. It has seemed clear to me for some time that the present trends just can't continue indefinitely. People don't want to contemplate doomsday, and our present administration functions only politically. I can't see that computers have much to do with the situation. The information is there all right, but the people in power don't act on it in a rational way.

BEMER: But is anyone doing any modeling on the available information?

KREHBIEL: The underlying problem is the gap between technology, which can isolate and define the problems (and sometimes find solutions) and the decision processes of management. I'm reasonably sure that the energy crisis was apparent long ago, but it was not expedient to do anything about it when we were using large amounts of fuel in the bombing in Viet Nam. I'm not sure that we're in real trouble even now. We may be seeing only an excuse to cut down excessive energy uses that will eventually, if unchecked, cause trouble later.

BEMER: Specifically, we ought to work toward requiring that information known to the Executive branch of the government be made available to the Congress. There should be an energy data bank whose contents is available as needed. The Freedom of Information Act should make all the needed information public.

PARKIN: I can't believe that you could get all the information. Suppose, for example, you asked IBM how many computers they have in warehouses. Wouldn't they resent it? The government can ask for information, but each company (in this case, the oil companies) regards their way of calculating their reserves and their methods of production, and distribution as their property.

GERALD: But even if this all came about, would it really have any impact? Recall the world model (Rome) of a few years ago; did it cause any significant action? It made people stop and think, but has any tangible action come from it?

ARMER: But the energy crisis has more facets than that. Even if the supply could keep up with an expanding demand, thermal pollution becomes a serious problem. But let me reiterate: it seems to me that you're posing a general question of how we can get better government.

BEMER: My point is that the computer field doesn't push the proper use of its product the way other fields do.

PARKIN: Just what industry does push the proper use of its product?

BEMER: Well, I can point to Honeywell. I try to push sensible use of the product and they provide me with a magazine to do it in (The Honeywell Computer Journal) and seem to be pleased with what I do.

I worry that during a recession people will become disenchanted with computers and will recall that their use as a tool in the energy crisis was of little consequence.

PARKIN: I take a different view of computers. I look on them as intelligence amplifiers: as drudgery--grinders; as tools in exactly the same sense as a lathe. Computers do precisely what we tell them to do. They will probably become as pervasive and all-encompassing in our lives as electric energy. I expect that computers will change civilization more than the industrial revolution did. I don't see how we can fault computers for the ills of our government. The 500-odd men in Congress worry about (1) getting reelected, (2) lining their pockets, and -- maybe -- (3) the country's problems. It's our fault if we don't set up the mechanisms for getting better people in government. You can't blame computers because people chose not to correlate data.

GREENWALD: Do you suppose that most of the uses of computers justify the energy they're consuming?

BEMER: When the EPA was formed, it brought together various units of government, each of which had computers. They found that they couldn't swap information back and forth; the data wasn't self-descriptive.

GREENWALD: But everything needed, both in hardware and software, to communicate efficiently is well known. All you're saying is that people don't have sense enough to use correctly what is already there. There isn't much we can do about that.

BEMER: But within the computing industry itself, there is great disregard of existing standards relating to data interchange.

BRADDOCK: The problem, then, is management of data, irrespective of computers. Will people trade data? Will people allow data to be centralized? We can readily develop procedures for all such things.

BEMER: There are two kinds of data: public and private. I don't worry about public data being made private; we can control that. I worry about private data that can't be made public; that's the tough one. It is very difficult to go into someone's data file and read it, since we aren't accustomed to interchanging data.

GRUENBERGER: I'd like to get us back to the agenda. Let's look at some of the items on our questionnaire for which there was such a wide diversity of opinion. Take number 5 (The world's chess champion will be a computer program by the year ____); that is surely the least emotional of all the items. The responses range from 1980 to after 2000.

BEMER: What's so surprising about that? That's what you'd expect from computer people. Remember the response to language translation 15 years ago? There were the fans and the debunkers, and it turned out that the debunkers were right.

BRADDOCK: Chess is essentially a mathematical equation, albeit quite complex. Conceptually, every chess rule could be remembered, and every alternative programmed.

ARMER: But the total number of paths exceeds the number of molecules in the universe.

BRADDOCK: But all you're asked for is a champion. The program doesn't have to play perfect chess; only winning chess.

GRUENBERGER: The notion of programmed chess goes back at least 20 years. Around 15 years ago, chess playing programs began to get rated, using the same system as is applied to human champions. The rating for various chess programs has been going up steadily, if slowly. Can we not extrapolate this curve to the point when the rating is significantly higher than that of Bobby Fisher, and thus arrive at a realistic date?

REINSTEDT: Recall Hubert Drayfus' analogy of the guy who climbed a tree and announced that he was on his way to the moon.

GRUENBERGER: But that's a poor analogy. The trip-to-the-moon curve can't be extended at all, and the chess program curve has been going steadily up for years, and you can't detect any sudden end to its rise.

GERALD: But the man in the tree might not be aware of the discontinuity he is about to encounter.

PARKIN: There may be a local maximum, and we may be on it. That's a standard fact of life.

GERALD: But Fred asked about the divergence of our opinions, and that may explain it.

GRUENBERGER: Sure; I'm simply wondering why everyone doesn't have the clear view of things that I have.

GLASER: Would our range of dates narrow if we made some assumptions about the amount of funding that could be expected for the chess project?

GRUENBERGER: The main characteristic of such work is that most of it is bootlegged (I guess the proper term is embezzled). Machine time gets chewed up at a place like Boeing, and Boeing doesn't even know about it. Most good, interesting things get done that way. Some of it is done in universities (where it is legitimate), but a lot is done surreptitiously in industry.

BLOCH: It's not clear that more progress would be made in chess programs if funding were made available.

GRUENBERGER: I agree; this sort of work is done for its own sake. My point is that if we look at the curve of progress, as measured by the rating system, we might predict that the rating will exceed that of Bobby Fisher within some finite time (say, ten years). I don't see how, when there has been continuous progress, we can logically say "never."

ARMER: But some of us may see that local maximum and logically conclude that the answer really is "never."

GREENWALD: Remember, we went through only one iteration; we haven't gone through the Delphi exercise yet. This meeting is just the second iteration.

GRUENBERGER: Evidently we still have people who believe that the date is "never."

BRADDOCK: Looking at the common element, none of us seems to think that it could be before 1980, so we all feel that the minimum time (if it's possible at all) is seven years.

GRUENBERGER: Many people have pointed out that short range predictions tend to be optimistic (the time flies by with no progress) and long range predictions tend to be pessimistic (we get there sooner than we expected). The average of the extremes is six years; that is, if you predict for six years away, you tend to hit it.

GREENWALD: Perhaps the success of the chess research depends on a breakthrough that we can't foresee.

BEMER: There's another factor, too. How about Bobby Fisher's progress, assisted by a computer program?

PARKIN: Then, too, the difference between our personal knowledge of chess and the comprehension of the giants like Fisher and Spassky is so great that we can't really understand the problems involved. They have deep knowledge and understanding of just one thing, probably to the point where they can't express what it is they know.

BLOCH: It seems to me that the masters exhibit an element of creativity (in addition to the mathematical aspects of the game), which is why I voted "never."

GRUENBERGER: In actual play between real people, there is also a lot of poker-type psychology, so that elements like daring, boldness, and unorthodox play can have an effect. A computer program playing against a human might have these elements, but a truly championship program (which would have to play against another program or against itself) couldn't use such tricks.

BRADDOCK: Would we advance faster if we taught some chess masters how to program?

ARMER: That's been tried, particularly by the Russians, with no more success than we've had.

GERALD: At USC, they're experimenting with computer programs that can be assisted by people.

ARMER: Let me ask Erich, "Do you believe that the world is knowable (not saying anything about time)? I believe that it is.

BLOCH: I guess I don't, in any complete sense.

ARMER: I think that, given enough time and work, we will be able to trace causes and effects and know why everything (including our brains) works the way it does. That's why I think that there will some day be a computer program chess champion.

BEMER: Of course, the world may evolve faster than you can learn to know it. And it may also be that, compared to perfection, Fisher plays terrible chess.

KREHBIEL: Let me go back to that man climbing a tree. He may not get to the moon, but while he's climbing, he may pass up the next best man, which is all we require.

BEMER: How linear is that chess rating scale?

GRUENBERGER: I don't have the figures, but it goes something like this. Fisher is rated 3500, say, and the best existing chess program rates

2400. But just a few years ago the best program rated only 500 or so.

ARMER: But that doesn't tell you what the rating really rates, or what the change from 500 to 2400 means. I recall talking to Samuels about his checker program, at the time it had just beaten the Connecticut state champion. Some months later it had been beaten by the U.S. champion, and Samuels remarked on the vast difference between the two men -- a difference that put the better man many standard deviations out on the curve of checker-playing ability.

PARKIN: That was my point a while back. The difference between 2400 and 3500 may represent a 50% gain, or it may represent three orders of magnitude in difficulty.

GREENWALD: Would the fact that there are psychological factors involved in play with humans be an advantage or a disadvantage in play against a computer program?

PARKIN: People psych themselves out; no one ever does it to a man.

GRUENBERGER: I can furnish a data point there. I'm fond of the game called Pasta. I've found that it gives me a great advantage to play several people at once. Even though each of those people can regularly beat me, when I play them all simultaneously, I tend to win all the games. The only difference must be the psychology of the situation.

REINSTEDT: There is such a thing as psyching people out, but I don't think that it can be considered a factor in devising a winning chess program.

GREENWALD: Moving on to our question 7 (fingerprint recognition), how can someone justify answering "now"?

ARMER: One of the weaknesses of the Delphi technique is the difficulty of wording the questions so that they're unambiguous without at the same time revealing what the designer considers the "proper" answer. This is a fine example. The problem of fingerprint recognition has two distinct and widely differing meanings:

(1) Here is my fingerprint. Does it indeed match the one in your file labelled "Paul Armer"?

(2) Here is a fingerprint. Whose is it?

PARKIN: Yes, the second meaning is the more difficult, but it was that one for which I said "now" because I had just read an article that claimed that it could be done.

GREENWALD: I think I read the same article, so I said "1975" on the grounds that it's probably not yet operational.

BEMER: I said "now" because I talked to Joe Wegstein, and he's the one who's doing it.

PARKIN: We ought to agree that 1973, 74, and 75 are all now.

ARMER: The work on fingerprint recognition could be in the same state as that of language translation 15 years ago; that is, we could be deluded by small successes into thinking we're almost there.

PARKIN: No; it's a much better defined problem and much more amenable to a realistic solution.

BRADDOCK: I hear a lot of agreement on "now" for this topic, but the average is 1983. Who are the people on the high end, and why are they so high?

KREHBIEL: I probably voted "never," but I hadn't seen that article. I take it that the ability to scan a pattern and reduce it to a set of numbers that describe it is now an accomplished fact.

ARMER: As regards to fingerprints, I'm somewhat doubtful, but what does Wegstein say?

BEMER: He's bullish.

ARMER: That's something short of stating that it works.

KREHBIEL: What's the reliability factor for this work? Is it sufficient to say that a given fingerprint belongs to one of ten people, and let human readers take it from there?

BEMER: In police work in Los Angeles, they figure they're doing all right if they don't exceed 10% of false identifications.

KREHBIEL: I wouldn't buy success in this area until they can do a lot better than 90% correct.

GLASER: I read recently of some success by some IBM people on signature recognition. Is it as good as they said, Erich?

BLOCH: The way it was stated, it was exaggerated. It seems to be good, but there is little data to validate it. It remains to be seen whether or not it holds up over a large sample.

BEMER: There are patents in these areas, but we can't figure out how to use them.

GREENWALD: If the question had been put "Will fingerprint identification be used in such-and-such an application" then my response would be quite different.

BEMER: From what I've been reading, optical pattern recognition just isn't making it. I doubt that we'll see it in our supermarkets for some time.

GRUENBERGER: Let's try an inflammatory item, like No. 2 (PL/I). I tried to word it very carefully. ALGOL is alive in Europe, but is totally dead here. It's still available from many vendors, but they don't brag about it. You can't buy a machine on the basis of its ALGOL capability.

PARKIN: I tended toward "never" on that question because of my personal opinion that languages tend to proliferate, and the proliferation aids in spreading the use of computers.

GRUENBERGER: But will PL/I be supported by the manufacturers? Will they brag about that capability in their ads?

GREENWALD: I'd go even further, and predict that we'll have PL/I machines. The language provides a reasonably good model of computing; the only one that we have in this country since ALGOL 68; the only one that exists and is being used. It is the only model supported by dollars that is capable of being cast into hardware, the way Burroughs did with ALGOL. It's not ideal, but it's the best we have in this country.

BEMER: I work with the second largest vendor, and on the basis of my information, I answered "never." Honeywell isn't writing any new software in anything but PL/I. MULTICS is being offered commercially, and it's written in PL/I.

PARKIN: Sunce 1969, I have supervised, at Control Data, the production of a PL/I compiler. The work has been moved to Europe, to take advantage of less expensive programming talent, but we expect to have a finished product shortly.

BRADDOCK: So IBM and Honeywell have announced PL/I compilers, and CDC will announce one soon; that seems to be the current situation.

GREENWALD: I'm bothered by the response breakdown again; we seem to have too many "nevers."

PARKIN: Actually, I voted "1980" because of the way the question was worded. PL/I will be around for a long time, but I think it will decline in importance because of the pressure of newer languages.

GREENWALD: My answer (after 2000) assumed that there would be dialects of PL/I that would continue.

GRUENBERGER: Prof. Gerald and I were involved in the current procurement of new computers for the state college system. The committee we were on felt compelled to ask every college department (some 1200 of them) what programming languages they felt they would need in the 1975-1980 time period. There were some 800 responses, listing some 183 languages that someone considered essential to his work through 1980. The list included languages like SOAP and TYDAC, and five or six that no one (on the committee of 15 experts) could even identify.

Now, you can't ask for bids on machines and require SOAP, since even IBM couldn't deliver that. The winners in the survey were -- surprise -- Fortran, COBOL, PL/I and BASIC, and those are the only ones you can legally ask for anyway. In the same sense as the man who asks for SOAP, PL/I will surely be around in the year 2050 because there will be at least one clown who has to have it.

BRADDOCK: The situation years from now for PL/I will be much the same as the situation with 1401's today. The machine is dead, but people continue to use it, even if they have to simulate it on a 370.

BEMER: That's Blaauw's law: the persistence of established technology.

BLOCH: I can't speak for IBM, but my opinion is that the question of whether or not PL/I should be offered is irrelevant ten years from now. But APL is different; it's a much more generalized and universal language. I think that APL will survive, but in a limited way. PL/I will be superseded by newer languages.

GRUENBERGER: I collect clippings for a file of "wonders predicted" versus "wonders achieved." For example, I have about five clippings that say that some company expects to market a tape drive that will pack a trillion bits on a standard reel of tape. This thing never appears. Now, we've just heard that CDC will soon offer a PL/I compiler. Every vendor probably has an operating PL/I compiler, but they don't release it, since when they do they have to maintain it. IBM has PL/I, and we've heard that Honeywell does too. My question is: what others have it, released for customer use?

BRADDOCK: There's another stage to it. From what Bemer said, Honeywell is using it internally for their software development. So is IBM; at least they're using PL/S.

GRUENBERGER: OK, let's eliminate that. And let's eliminate cut-down subsets that are written for student use. Considering, then, only full-blown versions that the customers can use, is there anyone besides IBM and Honeywell?

PARKIN: There are customers of CDC who took a machine contingent on having PL/I, and they are using it. But it hasn't been offered generally.

GREENWALD: I believe that Univac also has released PL/I. If so, and Honeywell, IBM, and CDC also offer it, then it doesn't matter whether other vendors do.

GRUENBERGER: That's a different subject. I'm trying to find out which vendors offer it now.

PARKIN: Does it have to be guaranteed to give right answers?

GRUENBERGER: Of course not; the Fortran and COBOL compilers aren't.

GREENWALD: If you had the answer to your question, Fred, what would you do with it?

GRUENBERGER: Not much, except to give me one fact in an industry that seems to shun facts. But it might help me to reach an intelligent answer to our question No. 2. Well, as it stands now, I guess the answer to my question is 2 vendors.

KREHBIEL: What type of person will be using PL/I?

BRADDOCK: Let me put that question the other way around. There are installations that are entirely committed to PL/I now. If it dies, what are they going to do?

GRUENBERGER: The same thing they did to get into PL/I, which was to convert from Fortran and COBOL. They'll have to convert again.

KREHBIEL: Will whole companies be using PL/I? Will 3-man service bureaus be using it? Will the University of California be using it?

BRADDOCK: There was a survey of some 900 IBM users in which 14% claimed to be using PL/I in some way (unstated).

KREHBIEL: I'm under the impression that it takes a very large machine to run PL/I, and hence I conclude that only large corporations can use it.

PARKIN: But technology continues to improve. You ought to be able to implement PL/I on an 8K byte machine.

KREHBIEL: But right now it takes a big machine, doesn't it?

PARKIN: Yes, for the particular implementation that exists, but that's not the state of technology.

BEMER: Our people are amazed to find that PL/I is more than just another language like COBOL, but that it is a well designed system that allows the programmer to do much more than he could before.

GREENWALD: The Burroughs 6700 has a design that should lend itself to an efficient PL/I compiler, both for compile time and run time. The point is that PL/I is attractive enough to be cast into hardware, and eventually the compile time will tend toward zero. The same architecture could be cast into smaller machines.

KREHBIEL: But that's some ways away from me. I'm a small user, and I don't rate a 370/167. I'm dealing with a Gremlin that has its tail end chopped off, and the operating system keeps feeling around for that missing piece. Give me PL/I in a 370/115 and I'll start being interested.

PARKIN: There's no real reason why PL/I couldn't be implemented on a minicomputer before long.

KREHBIEL: But until that comes about, you can't talk freely about choice of languages, because today the size and type of hardware puts constraints on that choice.

BEMER: Our present day software is very inefficient in the sense of being prolix.

REINSTEDT: What is the half life of a programming language?

GREENWALD: Our sample size is small, but any language that IBM supports seems to go on indefinitely.

BEMER: It's 17 years now for Fortran.

GRUENBERGER: Is MAD still alive? For that matter, can we name any language that once had widespread use that had died? I know that the parochial dialects die.

GREENWALD: Apparently any language that is supported by multiple vendors and that is once used by many installations tends to go on forever because of the problem of conversion. It's always cheaper (or seems to be) to go on using the old language. It worries me, because the manufacturer himself has an investment in old software and hence will be cautious about producing hardware that leans toward a new language.

BLOCH: But that's also true another way. There are plenty of 650 programs that are being run on 370's through three or four levels of simulation.

ARMER: And they're probably running slower than they did on the actual 650.

GREENWALD: Whenever a manufacturer comes out with new architecture, he always feels obliged to make it run the old programs. Not only his own programs, but also all of IBM's old programs.

KREHBIEL: Since I still have many old 1401 programs, I tend to favor the vendor who will give me a 1401 simulator or emulator so that I can keep using them. I don't care if the simulation is inefficient; I just want the programs to run.

GREENWALD: But then every user has to pay for the privilege that you demand.

BEMER: A machine is its own best simulator. The way that should be done is to have someone set up a service bureau with a 1401 and do the 1401 jobs cheap.

GREENWALD: Every manufacturer would like to pick up established customers by being able to simulate the machine they're giving up. First, they have to figure out (today) how to run their old 360 programs, but worse, can they legally run OS?

BEMER: We had that problem in GE, and what killed the project involved was the question of duplication of the JCL.

GREENWALD: Not to overlook the problems with old data bases, which hurts even IBM, since they, too, have old data bases to contend with.

PARKIN: Each company's base of installations is its own worst enemy.

I'm appalled at our willingness to discuss this topic on so many levels simultaneously and expose so much ignorance among ourselves. We use the term "users," for example, without being precise. Each of us is a user of computers when we pay our phone bill. We are users in a different sense when we have a problem to be solved by computer; and in a third way when we are preparing a software system. The diagnostic engineer, who has to prove that the machine does or does not work properly, is a fourth type of user. Take that last case; the diagnostic engineer couldn't write his programs in anything but absolute machine code. But I can't imagine a commercial customer (my second type) ever writing programs in assembly language again.

KREHBIEL: Do you have a compiler, then, that will do what has to be done?

PARKIN: There may be exceptions, but the overwhelming bulk of commercial users do not write programs in assembly language, at least for production work.

GLASER: I'm involved right now with a message-switching application that is all written in assembly language. The reports from the data are written in COBOL.

BRADDOCK: That only says that in certain areas, like telecommunications, we don't yet have suitable higher level languages.

PARKIN: Yes, and ultimately cost considerations will make it necessary to produce such a language. The ratio of people costs to machine cycle costs is going to increase continuously. This fact will drive us to higher level languages.

GLASER: I don't agree with that at all.

KREHBIEL: Parkin and I are in two different times frames. Tom is arguing the way it ought to be, and I'm arguing the way it is.

PARKIN: I'm not expressing an opinion; I'm pointing out the way it will be driven.

GRUENBERGER: It's curious, but assembly language is used only at the extreme ends of the spectrum; namely, by beginners and unsophisticated users, and by the true professionals.

PARKIN: This is because the machine technology hasn't reached your hands yet, which will eliminate it.

BLOCH: And it's moving in that direction very fast.

GREENWALD: There's another problem. It's difficult to retrofit the capabilities of a higher level language into a system that was built on an assembly language. A lot of people pay a high price for this fact.

BEMER: Security is a problem, too. If I were writing programs for electronic interchange of funds for a bank that handles $8 billion per day, I wouldn't write in PL/I; I'd use an obscure language.

PARKIN: Better not rely on that for your security.

KREHBIEL: Let's see if I have this straight. You're assuring me that there will be high level languages that will do what I have to do, and they will be available on small machines.

PARKIN: Yes, that's the way we're heading, but it will take several cycles. The first cycle, which is going on now to a limited extent, is the writing of system software in high level languages. You can find Fortrans written in Fortran, for example.

GREENWALD: Somewhere down the line there must be software that interfaces directly with the hardware; in I/O, for example. Such software, I think, will always be written in some form of machine language.

GERALD: You are not precluding the necessity for an assembly language insert into a high level program, to take care of unusual situations, are you?

PARKIN: No, I think that's necessary.

GREENWALD: But a lot of that will be done through microprogramming.

KREHBIEL: I'm all for what I hear you telling me. COBOL makes me sick, because someone keeps changing the compiler, and I get different results from month to month.

BEMER: That's why the Navy is writing COBOL tests.

KREHBIEL: According to IBM, there is no way to test COBOL.

BEMER: If I have to choose between believing IBM and believing Grace Hopper, I'll pick Grace. The specifications on COBOL will not tell you what the standard is; what does that is the test.

PARKIN: The Navy tests for COBOL are beautiful and well written. If you want them, just send a reel of tape to Grace Hopper. It's a fully organized, hierarchical set of tests that are self-documented. They produce printouts that tell you the extent to which each feature is implemented, and each level of the test builds on the previous levels. The tests give you a precise picture of how your COBOL compares to the "accepted" standard.

BEMER: You can't imagine how many anomolies and ambiguities in the accepted standards these tests produce.

PARKIN: Of course. What is standard is what people are using; not a set of statements that someone writes down.

KREHBIEL: I'll change my vote. PL/I will live. I've been educated.

GRUENBERGER: Let's go to the question on minicomputers, which intersects with the question on the number of computers (numbers 1 and 10).

PARKIN: The count of 500,000 machines is never going to be reached with STARs and 7600s; the bulk of the machines will be minis. My personal opinion is that the minis will take over. There will always be a few dinosaurs around, of course.

KREHBIEL: Interdata just announced a 30-bit word megabyte minicomputer. Just what is a mini?

GRUENBERGER: For our purposes, it's what I defined it to be. Interdata can use the term if they wish, but it's not a mini. Some day the 6600 will be a mini.

BEMER: I'll revise my answer down (from 1980) in light of the pending programmable pocket calculators.

GRUENBERGER: They will have an effect, to be sure, but they aren't computers. Hewlett-Packard would be the first to agree to that, since they make both calculators and computers.

BEMER: Where is the line of distinction?

GRUENBERGER: A computer can alter its own instructions -- a calculator can't.

BEMER: That line will surely fuzz.

GRUENBERGER: Certainly; all lines do eventually. But as of today, it's still a sharp and objective distinction.

PARKIN: In spite of the fact that we don't actually do instruction modification anymore.

KREHBIEL: I still don't understand what a mini is. I understand the characteristics of a computer, such as the fact that instructions and data are stored in the same medium and instructions can be treated as data by other instructions. But whenever I ask any vendor anything about "Can your machine do such and such?," the answer is always "yes." So what really differentiates the minis?

GREENWALD: I object to question No. 1 as being meaningless. I don't think that computing power can be expressed in terms of additions per second; the proper measure is data processing power, which implies files among other things.

I disagree that minis are going to take over the computing world. There's the question of centralization vs. decentralization. I think there's a big market for both sizes of machine, and I think that centralized computing will increase.

PARKIN: IBM will not discontinue the sale of big machines; there will always be a market for the biggest and most expensive machine. But more and more people are going to question the wisdom of having a super-large machine that is cut up, at great cost, into many little machines, which is what the users see. Technology will eventually produce small packages of computing power (defined any way you wish) accessable and available in clusters to the users. The number of minis will far exceed the number of other machines.

BLOCH: But the question concerned the dominance of the minis in computing power, not in numbers of machines.

GREENWALD: And a lot of those minis will be connected to a central maxi.

BRADDOCK: The Nov. 28 Computerworld ran some data on minicomputers. For 1971, they state there were 27,500 minis; for 1974 they estimate 99,000 minis. By 1977, they estimate 278,000 minis (and 54,000 others).

BLOCH: You need to think about the uses of those machines. The larger ones will be general purpose, but many of the minis will be devoted to specific tasks and will be hidden. An example is a mini inside a key-to-disk system. Although it is intrinsically a general purpose machine, in practice it is not accessable to other users and it functions as a special purpose machine.

BEMER: By 1980, a lot of sharp kids will be tapping into their automobile computers and getting useful computing done while they drive.

KREHBIEL: At our high school, the kids are using what used to be an Interdata front end machine, which they are rapidly turning into a computing network for the whole school system. It's difficult to convince our business manager that there is any real difference between a 96K 370 at $4800 per month and a 96K mini at $700 per month; he keeps asking the same simple question that I keep asking -- what is the difference?

GLASER: The analogy is to transportation. You can argue about the inherent efficiencies in buses, 747's, and mass-transit, but there are still people who prefer to drive their own cars. The mini computers are like the individual's car.

BLOCH: You could, if you tried, find other interesting uses for an electric toothbrush. Mini computers will probably be used in many different ways in the same sense.

GRUENBERGER: Maybe eventually that will be the line of distinction between minis and others; namely, their dedication.

PARKIN: That's the way the technology is going, but that doesn't mean it's the only way.

GRUENBERGER: All lines of distinction eventually blur.

KREHBIEL: You can afford to do less things with a mini and still justify their cost.

BRADDOCK: Another important distinction is that a mini can operate on a single file, as opposed to a data base.

GLASER: Also, the company controller gets nervous when he sees a large machine idle, but you can hide a mini in a desk drawer and no one worries when it's idle.

PARKIN: I do not expect to see peripheral devices shrink in volume as much as people would like. Keyboards, for example, will always have to conform to the size of people's hands. Storage devices will shrink considerably, but not other devices. A lot of people are going broke trying to engineer reliable, small, and cheap devices (such as disk drives and cassette drives) but we aren't making much progess.

GRUENBERGER: The present cassette drives are not much better than audio devices, and their reliability is very low. For example, I have found that no matter what the brand or price, I run the risk of jamming if I use high speed forward or reverse on this cassette device.

BRADDOCK: I should think that such devices could be made reliably and cheaply by now.

BLOCH: You can have any degree of reliability, but you can't have it cheaply.

BEMER: What you have to do is what IBM does; namely, record 9450 bits to guarantee 6250.

GRUENBERGER: 45% redundancy seems like a high price to pay for reliability.

GERALD: What's the alternative? You'd go to 100% if you had to.

BLOCH: The cost per bit is going down at a much higher rate than the redundancy requirement is going up.

GRUENBERGER: It still seems too high. People pay for 12% redundancy on their half inch tape drives, and object to that. Will people sit still for 45% redundancy?

BLOCH: The difference is that few people will know that they're paying for 45 or 100% redundancy.

PARKIN: And what difference does it make? If they can get the reliability they need, and the cost per bit going down, and the cost of people going up -- it's a good bargain. It's not an issue any more; you really don't care if there is 400% redundancy.

Let me try a provocative point. I run an advanced concepts research laboratory. One thing we worry about is the time when the hardware is so cheap that you can essentially give it away and charge only for the system or the software or something else. The cost per bit of storage or of logic element is ever-decreasing, and at a steady rate. It is easy to see ahead to the time when it will be feasible to produce something functionally equivalent to a 6600 in a package the size of a cigarette box, for which the most expensive part is the plug. How will we use the technology at that point? We continue to have dramatic breakthroughs in technology; they're evolutionary but still dramatic. Such things drop the cost by an order of magnitude. Sometimes it takes a while before they are observable, but they do happen, and apprently without letup. How are we going to adapt to make use of those breakthroughs? It's this thinking that guided most of my responses. We have to look ahead to the time when bits, and logic elements, and redundancy will be so cheap as to be negligible. That's why I think, for example, that languages will proliferate, rather than die out.

BLOCH: There is no question but that hardware will become very cheap. But I don't think that the question you posed is meaningful. When whole computers can be had for almost nothing, people won't be interested in buying them as such, but will be interested in buying a system that is tailored to a specific application. The vendor will be concerned with the system cost, of which .1% may be hardware.

BEMER: There's an analogy to electrical energy. We may be able to generate power very cheaply, through nuclear energy, but the big costs will be in distribution and billing.

PARKIN: Except for one thing; the analogy may break down if the user can have the whole system in a shoe box. Maybe he doesn't need to be connected to a huge distribution system. It may not happen with elective power, but it will happen with computing power.

BLOCH: It's the other way around. It may happen with electric power, but computer users may be more interested in data than in computing power and will therefore want to tie into a network.

BEMER: IBM's policy may be to sell you many small computers so as to avoid that problem. They can't afford to support small users who have the entire system at hand. They may choose to avoid cheap general purpose machines in favor of cheaper special purpose gear.

ARMER: Look at what Hewlett-Packard did for stockbrokers with their model 80. They didn't offer a general purpose calculator; they did the systems work and offered a device to do specific problems in compound interest.

GREENWALD: Are we predicting that as the cost of hardware goes down, there will be a proliferation of turnkey systems?

ARMER: Of course. The vendor can't afford to support the general purpose software and thousands of little customers.

BRADDOCK: Isn't there a danger involved when the Madison Avenue touch is applied to computers? Shouldn't we be concerned with what the computing industry should do about inexpensive machines? We see mass sales of pocket calculators right now, with people buying them as toys. When real computers are sold in drugstores the same way, people may buy them the same way -- what effect will that have on our industry? The people who need software, for example, may become neglected, because a mass production hardware industry can't afford to support them.

BEMER: I don't think you'll see computers as computers at that time. They'll be buried in things like sewing machines and autos.

KREHBIEL: When I analyze what I do, it breaks down to 10 or 15 distinct things. If computers get cheap enough, I could operate with 10 or 15 machines that could be modified -- mainly for printing formats -- with something much like a plugboard. Maybe we could get rid of the software. I certainly wouldn't need multiprogramming, or an operating system, and if I could get rid of COBOL I'd be the happiest man in the world. The language is fine, but I am aware that human beings can't write compilers.

PARKIN: Let's examine the parallels with desk calculators. The old mechanical machines cost from $700 to $1700; they were bulky and awkward; and the whole industry made 200,000 of them in 20 years. Look at the sudden and dramatic change in that industry. Now they make 200,000 <u>a month</u>; they cost from $50 to $400; they're small and fast; and everyone has one. (And "they" incidentally, does not include any of the three makers of the mechanical machines.) I think that the same kind of thing will occur with computers. We won't really build what would now be called big computer capability in a matchbox. We may put 5000 or even 15000 logic elements on a chip, but not 15,000,000. We will see a proliferation of general purpose machines which will fit in a cigarette box, and they will be very cheap. The question is, What are we going to do with them? How will we build the total systems that will properly utilize them?

GREENWALD: What about storage, and physical size?

PARKIN: It all goes together; the costs of all the electronic parts, and the size, keep going down.

GRUENBERGER: The present day one-upmanship among students is "my calculator has more function buttons than yours." Our students are carrying around $400 machines to class. When I went to school (that is, during the Civil War) students balked at laying out $15 for a good slide rule.

GERALD: It's amazing how affluent students are.

GRUENBERGER: Or dumb. I'm not sure they know how to use those machines properly.

BRADDOCK: Besides worrying about what can happen to the hardware and the software, what will happen to the people? For example, my secretary used to have a standard heavy adding machine on her desk, and she was the only one who used it. The company switched over to electronic calculators, and now the small machine she has is being borrowed continuously. Whatever it is that people are doing with it, they didn't do before, or else did it the hard way.

GRUENBERGER: Not only that, but in the home, the wife and kids are using these machines.

GLASER: Yes, my two children are getting pocket calculators as stocking gifts.

GRUENBERGER: I could have had a Monroe on my desk for 20 years and no one would have touched it. The pocket machines don't intimidate their users, and you can't hurt the machines by pushing their buttons.

PARKIN: The programmable pocket machines are going to extend that trend by a quantum jump.

REINSTEDT: Will there be social implications to all this? Will we, for example, stop teaching the multiplication tables and long division in our schools?

BRADDOCK: That's easy to answer: they don't teach those things <u>now</u>. But I'll agree that the changes will be significant. My grandfather was trained in pencil-and-paper arithmetic; my father used log tables; I used a sliderule. My children are already using calculators. What will my grandchildren be taught?

PARKIN: There are lots of things that your grandfather did which, if you did them today, would make you a candidate for the funny farm.

GRUENBERGER: The current game among calculator users is "How many function buttons does your machine have?" Pretty soon it will be "How many words of addressable storage does your machine have?" Shortly after that (perhaps within a year) it will be "How many program steps can your machine hold?" When a pocket machine has a button labelled "standard deviation," a lot of people are going to ask "What is that?" just as millions of people must now be observing that their machine has a button labelled "divide," and up to then they had never had any use for division, much less to 8 significant digits. If nothing else, these new machines are going to have a profound effect on understanding, by the masses, of esoteric mathematical and scientific concepts.

GREENWALD: Will masses of people be able to deal with concepts like storage, sequencing, and complicated functions?

BEMER: Look what APL has done. The people who become familiar with APL think in terms of its functions, which are very powerful. They just naturally think at a much higher level. It may be that we can someday teach kids to start thinking at a higher level of abstraction.

GRUENBERGER: We have drifted into a discussion of question 9 (computing as a standard high school subject). I had great difficultly in understanding your responses to that question. The mean was around 1999, which seems awfully pessimistic to me.

BEMER: But computing might be subsumed under other subjects; that is, it may be spread over a lot of subjects.

GREENWALD: My reaction was that it should come about, but where are the teachers going to come from?

GRUENBERGER: There's a school of thought that says that if you provide computing power to students, then the best function of the teacher (other than scheduling access to that power) is to get out of the way. In other words, the computer itself is a great teacher.

ARMER: But if we do need teachers, we have to face the fact that we will be stuck with the ones we have now for a long time.

GLASER: Perhaps computing will be taught in the home or by cable TV, and we won't rely on the schools.

REINSTEDT: Colleges have the same inertia as high schools, and they have shifted. Even education majors are required at some schools to take computing. I said 1980 because I think that the curriculum in high schools is a reflection of what the colleges do.

KREHBIEL: My daughter is having trouble getting the courses she wants because of the requirement that she take Euclidean geometry.

GRUENBERGER: If the colleges dictate the high school curriculum in any way, you might expect that our incoming students had had reading, writing and arithmetic. I'm dealing right now with a generation that has managed to skip all three of those subjects. I think that perhaps half the problem is simply poor teaching, but the other half is that those three subjects were actually eliminated from our schools for about ten years.

What about the people who voted "after 2000"? Do you agree that computing will be taught, but perhaps not as a distinct subject?

KREHBIEL: That's right. It won't be necessary to teach the notion of a loop, for example. Each student will have a "LOOP" button on his own machine, and he simply furnishes the parameters of the loop.

PARKIN: Precisely; and the teaching will concern the understanding of the term "parameter." Such basic concepts will be as important as algebra. Looping isn't just a computing notion; it's a topic in everyday life. Call it logic, if you will. As we noted before, it does no good to have more words of storage if you don't know what to do with them.

BEMER: Computing won't be a separate subject and it won't be taught -- students will learn it, which is entirely different.

GRUENBERGER: I agree. Not only won't it be taught, but you won't be able to <u>prevent</u> students from learning it.

GERALD: Maybe computing will displace algebra.

KREHBIEL: No, computing may combine with algebra and other subjects, but in any event computing will not be a subject on its own.

BEMER: When a student has a computer to play with, he'll discover all sorts of things about numbers that you'd have to go to great pains to teach any other way.

GREENWALD: Would a course called "How to Program Problem Solutions in BASIC" be an acceptable course in the context of the item we're discussing? Such courses have been around for some time now.

GERALD: If the kids are going to carry pocket calculators around, shouldn't we teach them what's in them?

BEMER: Why? We don't teach the workings of the telephone system.

REINSTEDT: If you make the analogy to algebra rigid, so that there are courses labelled "Computing," and a teacher called the computing teacher, and make it mandatory for graduation -- then I don't think it will come about. But you don't care how it's taught; the important thing is the knowledge you want a high school graduate to have.

KREHBIEL: The reasons for our present formal courses in algebra and geometry are not too good. What we seek is a grasp of symbolism and proof and problem solving.

GLASER: Perhaps a better analogy would be to a typewriting course.

GREENWALD: Some people had a "course" (perhaps only a module) in the slide rule in high school. I think that's the way computing will go.

PARKIN: The academic world at all levels is just as reactionary as anyone else, and perhaps more so.

REINSTEDT: In my observation, the schools are more amenable to change than is industry.

PARKIN: The two groups react to different stimuli. The industrial world can change instantly when the economic forces call for it. The academic world has different motives.

GRUENBERGER: The state has formulas for things on a campus, such as a formula for pencils per faculty member per semester, and so on. Most such formulas are functions of the number of students enrolled. For some time now, there has been a rumor that the state would use the number of students who take the final exams, rather than the number of students who register, and these are different numbers. You'd be astonished at how fast the academic world can react to economic pressures, Tom.

GREENWALD: My observation on the introduction of the New Math in the schools is that the attempt was disastrous simply because the established teachers could not or would not learn it themselves. So my question still is, How do we get the teachers to learn computing?

GRUENBERGER: But in computing, you don't have to. Where are the kids learning how to use pocket calculators? They learn it by themselves or, in the best or worst case, they teach each other, which is great. The programmable machines may be more difficult, but I'm sure it will go the same way.

BEMER: And eventually the machines will have a "HELP" button, as terminals do now, to lend assistance when needed and asked for. And when the student asks for it, he's all by himself in private, and is not embarrassed by seeming to be dumb. It's like bit redundancy; they'll be able to build so much HELP in to the machines, that a teacher will be unnecessary.

BRADDOCK: Probably the greatest single aid is the availability of floating point.

PARKIN: It's the old analogy: you can drive a car without knowing just how a carburetor works. We don't care that a student doesn't know the theory and mechanics of floating arithmetic, as long as he can use it.

BRADDOCK: But the student may miss the fact that the machine basically operates only in integers.

PARKIN: And that's not a fundamental concept that we want him to know, is it?

GREENWALD: We're back to the difference between learning how to use computers, learning about computers, and learning computing. Which is it we want in the high schools?

KREHBIEL: We have five Teletypes in a room with a sign on the door "No Students Allowed." When a new student is brash enough to go in there and ask if he can play, too, several of the old pros (those with three weeks' experience) rush to show him all the tricks. Admittedly, they probably do a lot of things inefficiently, but I don't think they make too many mistakes. It's not a standard course, and it doesn't attract a very large percentage of the student body, but it's a way to teach computing in high school.

PARKIN: Suppose the question had said "nuclear engineering" instead of "computing"? We'd all say, no, it wouldn't be taught -- it's too specialized. Computing isn't specialized: it pervades all our knowledge. Maybe algebra was that way in Abel's day.

GLASER: If a subject enhances a person's personal capability, it will find its way into the educational system and be supported.

BEMER: Typewriting will be a standard subject before computing will.

GRUENBERGER: One of the additional topics on our agenda is "Will hardware and software designers continue to go their merry way and ignore what the end users want and need?"

GREENWALD: Will the end user ever be able to state clearly what he needs?

GERALD: That's part of the problem.

KREHBIEL: It's difficult to get users to define what they need in functional terms. They don't know what they want or need, and they haven't learned how to describe their problems in any kind of operational terms. There's a lot of ego involvement in trying to pin him down; this is where a lot of the irritation and conflict arises.

GREENWALD: There has been some talk lately about system programmers being more responsive to user needs by deliberately slowing down their reaction to demands. They have noticed that more gets accomplished in the long run by increasing the response time. For example, two divergent demands cancel out, and neither has to be implemented. Or, various demands can be amalgamated, or it can be shown that various "needs" are not mutually exclusive.

KREHBIEL: My wife takes care of the subscription list for a publisher. They have been through five service bureaus now, each of whom claimed to have the solution to the problem. What someone should do is sit down with her to determine just what the problem is, because she hasn't been able to formulate her needs. In a larger environment, management has this same situation: they don't formulate the problem, but instead dwell on the format of the reports, or other irrelevant details. They get very defensive, because they don't want to give up their authority, and they feel that this is what they're doing when they talk to the DP expert.

GREENWALD: This is probably because the management man doesn't really know what he wants; but he wants it now and the system doesn't provide it.

GERALD: But there are techniques for finding these things out, usually by iterating, and producing a slightly better approximation to what is actually needed.

GREENWALD: Which means that we are always producing solutions to last year's problems.

GRUENBERGER: Here's an example. The original plugboards, being very large, had to have their bottom edge next to the floor. Hence, all plugboards on subsequent machines were placed down near the floor, where you closed the door in the normal way; namely, with your foot. Eventually someone noticed this, and moved the plugboards up where people live, at arm height. We had responsiveness to the user's needs (although it took 20 years to get there).

PARKIN: I doubt that. What the manufacturer noticed was the cost of replacing broken doors. It was only serendipitously also a response to the user's needs. I'm very cynical about things like that.

BEMER: Krehbiel mentioned the difficulties involved in communicating management's needs to the computer expert. One solution that has been proposed is to give the manager a terminal and let him punch keys to get the information that he thinks he needs. I suspect that this solution is doomed, partly because it is nearly impossible to write a program for it, and partly because managers don't like to do what appears to be what secretaries do. And that still leaves the problem of getting managers to be able to state clearly what they want.

GLASER: I think we have the cart before the horse here. I've been involved in some very successful programs, and it was not the situation that the DP people were trying to ram anything down management's throat. If the user (manager) has to pay, he learns rapidly to specify what it is he really needs. The good manager is not afraid of devoting 20-30% of a project's manpower to writing the functional specifications for the task.

GERALD: Maybe that's the answer. The initiative has to be taken by the user, and not the supplier. But when we talk about many (small) computers for many different users, we are not likely to find sufficient sophisticated users who are able to define their needs.

GLASER: That's right. If industrial users seek to tap a DP slush fund in order to acquire several machines for diverse uses, I submit that they are doomed to fail. The push must come from the demand side, but not blindly.

PARKIN: That may be true in the future, but we've had a "we'll give you what's best for you" de facto situation in our industry for quite some time. The manufacturers have not listened to the collective users and tried to give them what they want. They have been driven by the economics of the market, and as long as the economics were satisfactory, then to hell with the user.

GLASER: There are two different "users." The manufacturer's user is the guy who runs the DP shop, but that man's user is the man who runs production control.

PARKIN: The user Fred meant was the former. When you speak of providing a service, you must be responsive to the user, since that's what service means. Fred was asking how you get the manufacturer to provide you with an error-free assembler. There, the manufacturers give you what they, in their "infinite wisdom" think you ought to have, and that's it. As long as they can sell machines on that basis, what is there to make them change? It has to hurt economically. There have been machines that didn't sell very well, and that got the vendor's attention. He has no social conscience, and no motivation other than selling the next batch of machines. His chief responsibility is to the financial community.

KREHBIEL: And those are the same forces that drive the DP manager. When his management thinks they're spending too much for what they're getting, then they reason that something is wrong.

GERALD: But when something is wrong, perhaps some other manufacturer might think of doing it another way. Can he afford to?

KREHBIEL: There's a trend now to provide more computing power per dollar, which is in direct response to the high prices and relatively inefficient processing that we've gotten from the major manufacturers to date. I see the mini makers making a tremendous thrust toward impacting the markets of the majors. None of the majors furnish anything under $8000 per month, and the mini makers offer complete systems for under $2000 per month.

PARKIN: Fred, I think the answer to your question is that you've lived at the wrong time. Like most of us, you've lived through a period of fantastic, explosive growth; you've been riding a horse while it was in runaway mode. When our industry settles down -- which it is showing signs of doing -- then you'll notice an evolution in responsiveness of the suppliers. I think there's hope for the future.

ARMER: How much hope can we expect, when we examine the auto industry?

PARKIN: Well, the responsiveness takes time; perhaps three to five years.

BEMER: I said "No; responsiveness will increase, not from altruism, but from IBM's need to avoid erosion from independent peripheral and memory manufacturers, which will lead them to make computing applications specialized, but that won't be viable unless the applications are indeed well served.

GREENWALD: Might not this same question be asked of a lot of industries? The thing that makes suppliers appear to be responsive to the needs of users tends to be competition. In many industries, the wants and needs of the users are determined largely by Madison Avenue techniques. But it all reduces to a return on the investment, as Tom points out.

BEMER: One of my big disappointments (when Dr. Rader went back to

GE) came when I was not able to use my Univac software to write software for IBM hardware. It would have been fantastic, since it was just at unbundling time.

PARKIN: I suspect that there have been lots of such ventures. You just can't fight that magnificent marketing organization. IBM could sell horsecollars, if they put their trademark on them; they have 100,000 salesmen to peddle them, and at least 100,000 customers who would buy one without asking any questions. They have had some bombs, but very few.

GLASER: You're grossly underrating the customers. I know of many installations where the DP manager knows his business, and his management knows where the money goes. These men have stature, and common sense, and political clout. It's not universal, and it may never be; you can't stamp out idiocy. But I'm encouraged by what I see. The level of review committees is high, and by and large they're smart.

PARKIN: In large organizations, upper management has noticed the size of the DP budget.

BLOCH: And they've noticed that if things are not done right, you can't do business at all.

GREENWALD: Much of this management awareness and know-how was generated during the 1970 recession. Perhaps a 1974 recession will increase their awareness.

GLASER: I agree. People don't learn from an executive course or from a Fortran manual; managers learn when the Profit and Loss statement comes out. Along these lines, I'm a very strong advocate of charge back systems; I want the user to pay every nickel of the costs. There are exceptions, of course, but I know that with proper charge back, the quality of the work goes up and its reception is assured; everything gets better. It's painful, I know, since companies can say "This isn't our normal procedure; we don't charge for accounting services, for example." But accounting isn't discretionary, and DP systems should be, and when they're not, the chances of failure go 'way up.

ARMER: You're saying that you want feedback in a system.

GLASER: Yes, it's sharp pointed negative feedback, almost to the point of being punitive, but it has the right effect.

PARKIN: It's another example of a mechanism that is driven by economics.

GRUENBERGER: Let's turn our attention to another subject; namely, the future role of programmers. The simplest definition is the guy who is on the payroll with the title "programmer."

BRADDOCK: There will be a lot less of those.

GLASER: Are we talking about programmers who write systems, as in Braddock's shop, or the man who programs for Standard Oil?

GRUENBERGER: The latter -- user programmers.

BRADDOCK: There will be many more of the type who work at places like Informatics or for the manufacturers. These people will form small groups of highly skilled programmers, who will produce the tools (such as higher level languages or packages like Mark IV) that user programmers will utilize. Many people will be using computers, but not doing programming in today's sense.

GLASER: We will eventually see efficient programming techniques applied to systems programming, too, so that eventually we will need relatively few of those men.

ARMER: If we assume constant productivity of systems programmers, and the demand increases, then what? Will the demand go up faster than productivity?

PARKIN: The cost of the hardware keeps going down. I predict that the demand for systems programming is going to go up, rapidly.

BEMER: The monetary feedback information will operate, when people observe that the systems people cost a fantastic amount relative to the hardware. To reduce those costs, people will turn to automated techniques for software.

PARKIN: Not in my lifetime.

ARMER: I wonder whether the hope for significant improvement in productivity isn't akin to the same hopes for machine translation or machine chess.

BLOCH: No, it's a different kind of problem, and one that lends itself to new techniques. For example, we know how to apply engineering techniques to the production of software.

GREENWALD: But be careful; the studies that have been made aren't conclusive at all.

BLOCH: And management, in general, is very tolerant of programming productivity, because they don't understand what is done.

GREENWALD: But in the IBM studies, for example, it turns out that if you could double the amount of time actually spent on writing programs (versus everything else the programmer does), you'd still be under 2%.

BLOCH: Now, is this inefficiency due to poor problem definition, or because programmers are poorly organized, or because the tools are not available, or what?

GREENWALD: I believe it is largely because the problems are poorly defined. They spend very little time on actual design, implementation, and checkout.

PARKIN: Programming is an art, and is not yet ready to have engineering principles applied to it. We'll get there someday, but not now.

GREENWALD: All the productivity tools we're talking about are concerned mainly with implementation.

BEMER: Not so. One vital factor is turnaround time. When you get your programmers on-line, with quick feedback, and perhaps 30 shots per day, you get a fantastic increase in productivity.

GREENWALD: I am convinced that the one thing that will really increase productivity will be the ability to use off-the-shelf components. But we're a long way from that.

BEMER: Let me put it this way. Programming is a tricky thought process. The tie-up comes (with long turnaround times) in getting back in context with those tricky thought processes. Just by shortening the turnaround time (to nearly zero), the programmer stays in context and productivity goes up.

GREENWALD: And all our tools have enabled us to go, in systems programming, from 30 checked out instructions per day down to 5 to 7 per day.

GRUENBERGER: In the scientific area, we have done certain problems once and for all; for example, the solution of simultaneous equations, or gear design, or Bessel function calculations. Isn't there a corresponding body of systems software problems that have been solved, so that each man doesn't have to solve them all over again? Doesn't the building block principle apply here, too?

BEMER: It's more difficult. You might like a packaged tax routine that could be plugged into any program that deals with taxes, but the tax laws are too varied to permit it.

GERALD: But couldn't we create tax modules, that could be parameterized and then collected to fit specific situations?

GRUENBERGER: How many different operating systems doe we need? Won't we hit saturation in systems software, just as we have in scientific applications software?

GREENWALD: We do have off-the-shelf packaged routines, but always with a five year time lag. Today's packages solve the problems we had five years ago, and this time lag seems to stay constant. The trouble is that no one can state clearly the problems that we'll have five years from now.

BEMER: I can't imagine an operating system remaining constant through hardware changes like virtual storage.

BLOCH: Plus software improvements that are needed to provide security and reliability.

GRUENBERGER: Didn't I hear everyone agree that the demand for systems was going to go up?

BRADDOCK: Not from me. It depends, of course, on how you define systems software. We've all dealt with I/O instructions that deal directly with the peripheral devices. But today's systems programmers don't do that; they don't even know how tape or disk drives actually work, and they don't care. Their level of expertise is much different from that of systems programmers of ten years ago. A lot of people can turn out code in assembly language or Fortran or COBOL, but that doesn't make them systems programmers. We have developed a cadre of competent people who know their jobs, and they are developing the tools (or modules) that everyone else can use. One shouldn't generalize, but to my way of thinking, anyone who writes in Fortran is not a systems programmer; they are applications programmers getting a job done. We'll need a lot more of those.

BLOCH: I can't see what bearing the choice of language has on the matter. If he designs a system and uses Fortran, he's a systems programmer.

GREENWALD: Let's eliminate the semantic problem here. If he writes an operating system, or a language translator, he's a systems programmer and Braddock says there will be less such people. If he uses the product of a systems programmer, he's an applications programmer, and Braddock says there will be more such people.

PARKIN: I keep pointing out that the hardware is going to the point where we can give it away, and all we'll have left to sell will be systems.

BEMER: And the systems will be priced by systems programmers, who will know how to fabricate it cheaply.

BLOCH: There is no obvious relationship between cost and price.

BEMER: When you are the proprietor of the technique, you can control both of them.

GREENWALD: Even if the hardware costs become negligible, won't the number of interrupt handlers that you write be quite limited, and many people will use the same ones?

PARKIN: If, by systems programmers, you mean the people who write routines in assembly language to interface the CPU with its peripherals, then, yes, that group will shrink. But I take a much broader view of systems. A system is economically useful to anyone who will pay for it. There are few vendors left who sell computers per se; we all sell systems. I call all the people who work on them -- hardware and software -- systems programmers.

GREENWALD: I made the statement that there would be increased demand for the output of systems programmers (my definition of them) but that the number of people needed to produce that output might decrease. If we can learn how to do it with canned modules, then we can produce turnkey systems with fewer people.

BRADDOCK: Does anyone have a feel for how SOFTEC is doing in this area? They've been at it for over five years.

PARKIN: Ross is a good entrepreneur; he has some basically good ideas and he'll be successful at it, but it won't take over the world.

BRADDOCK: They can apparently turn out PL/I compilers on demand.

BEMER: IBM should take time out to look at their operation.

GRUENBERGER: We're always so good at solving IBM's problems.

BRADDOCK: As I hear it, we seem to be in agreement on this matter.

GREENWALD: I can't be in agreement, because I'm still in doubt. The problem is, how much of it can be automated?

PARKIN: Over the years, new techniques move from the academic world (where they're invented in profusion) to industry, always in response to some economic need. The use of higher level languages is a case in point. It is being adopted by the vendors to increase productivity and portability of their software investment. Ultimately, I think there will be a peak of systems programmers, and then it will go down. But it won't happen for quite a while.

GLASER: I wonder about the prestige level of these people: will they be like plumbers and bricklayers, or like engineers?

GREENWALD: I can see a lot of 50-year-old hacks in another 10 or 15 years.

PARKIN: There has never been another industry that has moved so fast that there can be 25 and 30 year old people who are technologically obsolete and beginning to ossify. It's frightening. When someone is only 5 years out of college, he can be completely out of date and useless for creative work. That's why I think the fundamental mechanisms of computing must be taught (and will be taught) at the high school level.

REINSTEDT: But isn't the time to obsolescence longer now than it used to be?

BEMER: No -- it's shorter.

REINSTEDT: I would think that, if someone had taken a year off around 1960, he would have been completely lost when he returned. Somehow, I don't have that same feeling today. Granted, it would be difficult to get back into things, but not as difficult as it was then.

GRUENBERGER: Of course, you're looking at it from a viewpoint that is 13 years older.

BEMER: I find myself racing all the time, just to keep current with the field.

GERALD: What is it that characterizes a 30-year-old technologically obsolete person? What could the educational system do to avoid that?

GREENWALD: One outstanding characteristic is that he hasn't read anything since he got out of college.

BRADDOCK: They don't even want to upgrade themselves any further. They don't even read the ACM Communications.

GRUENBERGER: Not to overlook the Honeywell Computer Journal.

GREENWALD: There's a British journal called Software Practices and Experiences which deals with "how I did it." It's amazing to see how many of the so-called professionals in our shop don't even know it exists. It seems to me that you can distinguish between a man who has a job and a professional man by the degree to which the man wants to upgrade himself. Management should encourage this, of course, but the individual must want to do it for himself. One way that I find to make the distinction is to note how much reading the man does of the available literature in the field. As a manager, I can encourage the practice. I can provide time for it; I can hold seminars. Recently, there was a series of seminars at RAND, designed to upgrade the professional staff, and the attendance was ludicrously low.

REINSTEDT: Maybe that tells you something. Maybe there's a real danger of obsolescence at the level of the people in this room. But at the level of applied programmers, is there much for them to learn after they've been programming for 3 to 5 years? We've agreed that the programming languages they use have long lives. So are such people obsolete if they don't read?

BRADDOCK: Ask them if they are aware of recent developments, like structured programming.

REINSTEDT: But maybe that tells you that they don't need (or feel that they don't need) that knowledge. I don't believe that, at some lower level, it follows that failure to read implies obsolescence.

GREENWALD: There are two kinds of obsolescence. We've been addressing the notion of technological obsolescence. But there is also economic obsolescence, typified by the man of age 35, making the salary that normally goes with that age, who is replacable by a man of 25. When there's an economic crunch, the economically obsolescent man gets fired.

ARMER: The recent IEEE salary study showed that the group under 40 went up 8%, but the group over 40 went down 5% in salary.

GLASER: A given person who calls himself a professional might well be reading about related subjects (such as his firm's business) rather than about structured programming. A man who is 35 and is no better a manager of his people and his budget than a man who is 25, should be replaced by the 25-year-old.

GREENWALD: Men over 40 have a different effect on pension plans than do younger men. The recent recession caused management to rethink the policies of whom to keep and whom to let go, based somewhat on facts like the pension contributions. We may get to the point where we must make plans to retire men at 40.

GRUENBERGER: Let me tell you about reading. Every semester for the last 4 years, Bob Reinstedt has come out to my campus to conduct mock job interviews for my advanced students; these are videotaped and critiqued by the class. Bob likes to ask the job applicants what they read (besides their textbook)--and usually the response is a complete blank. They don't read anything, in computing or any other subject.

PARKIN: There are large numbers of people in our field who graduated from card walloping to the 407 and on to the 1401 who are now called programmers, using Autocoder or RPG or something. Those people are obsolete. Their thinking is still in terms of card shuffling. They are using equipment that is 10000 times as powerful as the EAM gear, but their thinking hasn't advanced by that amount.

REINSTEDT: Anecdoted data has never impressed me, unless it is truly typical. Take the man who is doing competent work in PL/I programming. He writes code, as opposed to solving problems in the engineering sense.

BEMER: A lot of us were grounded in card walloping, and the principles we learned there are still valid.

GREENWALD: Let me return to economic obsolescence. People generally get paid more as they get older (but we may have to change that). They should, therefore, be worth more--and they're not. Prior to World War II, it was accepted that people would advance to a certain salary level, and then go no further. Since World War II, that idea has become unacceptable; people expect at least a 5% increase every year. They expect that much raise as a sort of pat on the head.

GRUENBERGER: But 5% isn't even keeping up with the cost of living. It was only after World War II that we learned to accept a 5% (or higher) increase in cost of living every year as a "normal" thing, too.

GREENWALD: But the point is, that people simply expect a raise, whether or not their contribution to the economy of their employer has increased.

ARMER: But I just cited to you an overwhelming piece of evidence of how the market place values extra years of experience. The market place is saying that after age 40, the added experience gets outweighed by obsolescence.

KREHBIEL: Is that characteristic just of our industry?

ARMER: No, I think it's characteristic of any profession in which there is rapid change. It's truer in medicine, for example, than you think, despite the shortage of doctors.

GREENWALD: The set of people we're talking about can keep their jobs until the crunch comes, and then they're out.

GRUENBERGER: I don't like what I'm hearing. You guys are saying, if I hear you, that I can be replaced by someone half my age, working at half my salary. Personal considerations aside (I know that I'm unique and all that), you're presenting a terrifying prospect, and I'm looking for the flaw in your reasoning. Has the pace of technology gone off scale so far that all of us are threatened, merely because we're over 40?

GREENWALD: I don't know how to measure your productivity, but I can, in a limited way, measure the productivity of an applied programmer. If I find that I'm paying the 40-year-old 50% more than the 25-year-old for the same productivity, and the crunch comes, what do you think I'm going to do?

REINSTEDT: If an applications programmer has learned all he needs to know (except for technical details of updating) after, say, 7 years of work, then why should he be paid more every year after that?

ARMER: So we have a social problem because we're violating an economic principle.

GREENWALD: We may have to try a scheme whereby people accept a cut in salary at some age, and go down from then on.

ARMER: I would guess that most of the over-40 group in the IEEE sample (and it was a large sample) were people who left one job for another, at a lower salary.

GREENWALD: How many of them were offered the opportunity to stay at the same job at a lower salary? Few companies even give a man that choice.

GLASER: Any company that has large-scale union agreements, is not about to give salary cuts on a large scale (or at least let it be known that they are doing it).

BRADDOCK: Has anyone here ever tried to retain an employee with a salary cut?

GREENWALD: Yes, in a case that involved a demotion and a salary cut. It wasn't quite the Peter Principle; the man had just gone up too fast. He couldn't hack it, and he knew it, and it bothered him. So he took it.

REINSTEDT: But now you may have a disgruntled employee, and they're dangerous to have around.

ARMER: IBM cuts people's salaries all the time, don't they?

GREENWALD: You have to gauge in advance what you think the man's reaction will be. If it is known to be standard company policy, then it is much more palatable.

BRADDOCK: But then there's no problem. The problem arises when it is an isolated case: where you, as manager, have decided that a man is getting paid more than he's worth.

GERALD: We're talking about a dramatic change in payroll policy. It would help a lot if people would read the current literature of the field.

BLOCH: You would have to arrange to change the man's work assignment in a visible way. You can't expect him to do the same work, publicly, at a lower salary.

BRADDOCK: But the economic climate must be such that he can't say "Well, I'll go somewhere else."

BLOCH: He has that privilege, of course.

BEMER: At Univac, I told my supervisors that they had to do two weeks of coding each year to maintain their position.

GREENWALD: Good for you.

REINSTEDT: The important thing is not the cut in salary, but the axe hanging over their head: that is, the insecurity. If you can trade some increased security for that cut, you'll get agreement.

PARKIN: A lot depends on the age level. What might work with young fellows on their way up might not work with older men. Or, looking at it the other way around, with an older man you might be dealing with someone who has selected his retirement institution, or someone who is simply tired. If you offer him tenure, so to speak, he'll leap at it.

GREENWALD: You're assuming that the man involved recognizes that he is getting paid more than his productivity warrants.

REINSTEDT: I doubt that many people would admit to that. Recall the initial remarks: people <u>expect</u> regular raises today, and hence believe that they've earned them.

GREENWALD: I thought we were talking about economic obsolescence of technical people. We seem to have now included managerial people as well. In many cases, the decision to graduate from technical to manager level solves the problem. I chose to stay technical, for example, so I now feel the obsolescence problem. I see other technical people my age in other installations, all worried about the same thing.

ARMER: And you don't think the managers are worried, too?

GREENWALD: I just don't know; I can't speak for those who chose to go into management. I can say this: if I chose to go that way, there are many jobs I'm qualified for.

BLOCH: But managerial credentials deteriorate over time, the same as technical credentials, and probably more so. The technical man may, at least, have established a reputation for himself in technical circles.

GREENWALD: No, the technical people we're talking about, however competent, are not generally known outside their own company.

REINSTEDT: It seems to be true today that RAND is not a place to retire; it's a good place to get 10 years' experience, but you should arrange to move on before you're 40. Now, is the converse true at places like CDC and IBM?

PARKIN: Some people already have chosen those companies as places to retire in, but it's really too early to say, since our industry hasn't matured yet. The flux we see in computing isn't characteristic of mature industries; we see it because of the explosive growth we are still experiencing.

GRUENBERGER: Tell me what I should tell my students (those who are headed toward careers in computing). Do I tell them that after 7 years or so they will be at peak salary unless they go into management?

GLASER: Yes, unless they pick up some merit badges along the way, such as knowledge of production control, or accounting systems, or manufacturing control, or go from sales to statistics to market research.

REINSTEDT: In other words, he must keep himself adaptable, and mobile, rather than narrow.

BRADDOCK: From management's point of view, a man should seek knowledge and constantly improve himself. The big trouble is that most people acquire only that knowledge that is essential to the project they've been assigned to. My big gripe is the man who is immersed in data base work (having been assigned to that task) who remains ignorant of another area (e.g., communications) which he should know about.

GRUENBERGER: I get the same thing (at a lower level) when a student complains that a question on the exam was unfair "because you didn't cover that in class."

REINSTEDT: Here's another example. At one time, linear programming was a big thing. If we had five programmers whose specialty was linear programming, and they had learned nothing else, then they'd all be in trouble now, because linear programming just isn't in demand.

GRUENBERGER: They should at least have the ability to switch to the current topic that is in demand.

GREENWALD: We're being unfair. A person gets involved with a specific area, like linear programming, because that was the work he was assigned to. When a new problem in that area comes along, he gets it because he's the expert in it. And as long as he's involved with his specialty, we expect him to work at it, and we're not apt to encourage him to be studying other areas. I doubt that that will change.

BLOCH: That's true for a drill press operator, but a professional man has a responsibility to keep himself informed, at least, about other areas.

BEMER: Part of the problem is caused by the people themselves. The tenure in a particular assignment could be halved (say, three years writing Fortran compilers instead of six years at it) if they would learn to document what they had done so they could move on.

PARKIN: I want to go back to Irwin's point. Fifteen years ago everyone in our field had a feeling of great excitement at being involved with this new high order of intellectual activity. Everyone could see years ahead of interesting new problems and applications, and everyone was learning at high speed. Today, that feeling seems to be gone. I am appalled at the 25 and 30-year-old people who have stopped learning; who say, in effect, "I've learned the trade; I'm an expert; I don't need to learn anything else." They keep going at that level, and they're hacks. What appalls me is how the hack level is appearing at earlier and earlier ages. Maybe it's the "they aren't raising kids like they used to" syndrome.

GREENWALD: Those of us in this room all learned by experience, since that was the only way possible then. We all did everything. But today we can get in a young man who gets assigned to SYSGEN work, and pretty soon he's the local expert and can't be spared for anything else. He could quit and go somewhere else, but he can't get reassigned within his company; he's stuck. Even if he tries for reassignment, we always have deadlines to meet, and we seem to be better off letting him be stuck.

BRADDOCK: So it's not his problem; it's your problem. It's a managerial problem.

GREENWALD: Sure, and the problem exists all over.

1Bm !
2

BEMER: I can tell you competitors how you can sabotage Honeywell: arrange for us to lose Sarah Fleming. She is absolutely indispensable. Or take the case of our Fortran expert, who went to MIT. It took two men three months, working night and day, to find out what was in the Fortran compiler.

GRUENBERGER: Sarah used to work for GE. How did Honeywell get along up to two years ago?

BEMER: Honeywell wasn't trying to run GECOS-III on the 6000 system.

ARMER: Maybe Sarah doesn't document her work?

BEMER: No, she's thoroughly competent. She doesn't have anyone to train.

ARMER: So there's another managerial problem.

GRUENBERGER: You'd have to live with some of that GE equipment to really understand the problem.

GREENWALD: I was at RAND for 19 years, and the thing that impressed me the most (though it may not be true today) was that everybody -- managers and senior technical people -- felt an obligation to help upgrade other people. The first thing that impressed me in industry was that everyone feels obligated to meet the schedules and get that return on the investment. Of course, this is generally true in industry.

PARKIN: So you've discovered the economic pressures of industry.

ARMER: What's the short term and long term economics? What's the discount rate?

PARKIN: Few people know the difference.

GREENWALD: I agree. We have an economic problem, of the two kinds of obsolescence. We keep tying a man to one job, and he can't do anything else.

REINSTEDT: What is the average length of experience for applications programmers?

ARMER: It's probably greater than 7 years.

GLASER: Does that mean that there are few new entries to that field?

PARKIN: By and large, yes.

REINSTEDT: A study in 1963 showed that the main objective of a programmer was to become a non-programmer. A recent study of mine shows the same thing. "What do you want to be 6 years from now?" Most people want to be something else, like a manager. They don't regard programming as an end in itself.

GRUENBERGER: At the end of my first week at Informatics, I filled out the little time card and a girl came to get it to deliver it to my boss. I could hear his screams from down the corridor. I had put down an hour for time spent reading the literature of the field. I was quickly informed that you don't do that, in industry.

GERALD: That is, you don't report it.

GRUENBERGER: No, you don't do it.

GLASER: If a man calls himself a professional, he should spend some of his time doing that.

BRADDOCK: That's right. I want my people to keep up with the trade journals and books, but I don't want to see them doing it at their desks.

GREENWALD: Then you're not staffed with professionals. What you have is a bunch of wage-earners.

REINSTEDT: I would think that an enlightened company would feel obligated to support some kind of professional advancement on company time.

GLASER: But how many men in Parkin's position read Business Week at their desk?

REINSTEDT: If we say, in effect, to programmers "It's important for your longevity that you diversify your talent," then we ought to do something (say, a seminar once a week) to back that up.

GLASER: The only time I've had the luxury of that treatment was when I was in a RAND-like atmosphere. No other type of outfit feels it can afford it.

BEMER: Some companies feel they can; Honeywell and Polaroid, to name two.

GREENWALD: But what motivation is there for the owner of a company to do what you're suggesting, when he can get what he wants simply by firing the guy and hiring someone else?

REINSTEDT: But that would cost you more than it would to retain him.

GREENWALD: I'm not at all sure of that.

REINSTEDT: Consider recruiting costs, training costs, and all the rest.

GREENWALD: But not when I'm getting rid of a 40-year-old and picking up a 25-year-old.

ARMER: And a lot of that does go on.

REINSTEDT: I'd like to see the cost analysis on that. I don't doubt that it happens, but I'm not sure it's economically sound.

GRUENBERGER: There is no way that I can prepare students for what you men are describing, but I can tell them how it is.

ARMER: There was a report in 1971 in the Harvard Business Review that showed (mostly for aerospace engineers) a peaking on the lifeboat list of the company, as contrasted with their pay. As I recall, it showed a peak at age 30-35. Supervisors peaked 5 years later, but fell off much more rapidly.

KREHBIEL: I don't know why you guys are so surprised at all this. My father's principal told him when he started teaching that the first year would be rough; that he would do his best work the second year; that it would be downhill from there on; and that his (the principal's) job would be to be on his tail to keep him going.

REINSTEDT: But most of our fathers worked at the same job for 30 or 40 years, and didn't get fired at age 40.

KREHBIEL: That's right. The world hasn't changed very much, but firing practices may have, but even that is doubtful if the shop is unionized.

REINSTEDT: So maybe what students should be told is to unionize (which comes right after certification).

ARMER: Is the faculty unionized at your university?

GRUENBERGER: There are four active unions, each scrambling to represent all of the faculty. The four range from militant to completely passive, and each faculty member can pick which (or none) he wants. As things stand, he has to pick one in order to get medical benefits.

Let's move on to item 12 (Certification or licensing of programmers). Here the consensus startled me, since nearly everyone said "We aren't going to do that." I think we are, and that it may be forced on us in ways we won't like.

BEMER: I was one who agreed with you and said it will come by 1980.

GRUENBERGER: We can define the problem better by making a complete analogy to the mechanisms for the CPA. The rules for that; the avenues toward getting it; and the enforcement procedures, are all laid out -- for over 25 years -- and they work.

BLOCH: But the technology of the CPA has been the same for 300 years.

GRUENBERGER: Don't believe it. The accounting world changes pretty fast.

BLOCH: But the changes are second and third order effects.

GRUENBERGER: Not so. The changes are not as fast as in our business (and far more orderly) but they are first order effects. For example, about 10 years ago they sent a CPA to jail, telling him "You should have known" and not accepting his plea that he didn't know of the shenanigans that were taking place in the firm he was auditing. For 25,000 CPA's in the country, the ball game changed its rules overnight. More recently, we've had Equity Funding, which will cause even more changes.

GLASER: Going back to computing, are the objections to certification and licensing due to a belief that we can't do it right, or that we shouldn't do it?

REINSTEDT: My position is that we can't possibly do it right (but that we're going to do it).

GLASER: If that's true, and it comes about anyway, what will happen? Will we find ourselves with a lot of people who are certified but incompetent?

PARKIN: Consider all the certificated teachers in the world. Not all of them are competent.

GRUENBERGER: Can we agree that the program has worked for the CPA's?

REINSTEDT: They are not all equal, but I get a distinct feeling of what constitutes a CPA, and I think most of us do. But try to extend that same notion to programmers.

PARKIN: You would regulate massive mediocrity.

GLASER: Maybe we should seek the same distinction that exists in medicine, wherein the man who practices must be licensed, but the research worker need not be. Perhaps we could distinguish the computer scientist who does not practice computing (for the public, say) and for whom it would be difficult to organize certification criteria that would stand up for more than six weeks. On the other hand, the practitioner of computing might lend himself to some sort of certification.

BRADDOCK: The analogy with doctors may be a good one. There is probably a written examination for them, but the real test is their apprenticeship, which goes on for several years. We will face the same problem, and our solution should probably be the same: namely, a long apprenticeship.

GRUENBERGER: I used the word "programmer" only in the catch phrase "Certified Public Programmer," but the question relates to certification of computer people in general. We should be asking, can a man be certified as knowledgable about computers and their uses?

ARMER: For whom will such people work? Would they work for firms that send a man in to certify another firm's programs? In other words, would they function the way CPA's do?

GREENWALD: Companies hire accountants and they hire programmers. They can get a certified accountant if they wish, or they can also get one who is not certified. They could do the same thing with programmers.

GLASER: The CPA certificate has motivated a lot of people to try to reach a stated level of knowledge. It has done a lot for the accounting profession. True, a man crams to pass that set of exams, but it's unfair to conclude that he then stops learning.

REINSTEDT: I'm all for motivating people to learn more and upgrade themselves. But when you take the tests and get the certificate, what are you then certified to do?

GLASER: Well, it's much like requiring a Boy Scout to take a 50 mile hike. It won't guarantee his ability to survive in the woods, but it's evidence of some level of capability, and several such requirements put him ahead of the boy who hasn't done them. As things stand now, you have no evidence at all from anyone who walks in the door and says "I'm a programmer."

ARMER: I require that the guy I hire has a college degree. It's not that the degree has given him anything specific, but simply that the probability of finding a good man in that population is much higher than that of finding a good man in the non-degree population. The degree is a sifting device, and the certificate could serve the same purpose.

GRUENBERGER: We use calculus the same way within the university; it provides a reasonably logical way to separate students as they approach the upper division courses.

GLASER: The Harvard Law School graduate may not be better trained than the graduate of Podunk, but statistically he's a better bet. If nothing else, his survival ability is better.

REINSTEDT: But don't tell me he's certified.

GLASER: Not as an individual. But in hiring him, your risk is lower if that's all you know.

REINSTEDT: Then the term "certified" is a misnomer; worse, it's a non sequitur.

KREHBIEL: Is it any worse than what we expect from a man who can call himself a lawyer?

REINSTEDT: When I go to a lawyer, I know what I can expect from him.

PARKIN: You do? You must be as ignorant about law as most of us are about medicine, then.

REINSTEDT: But what's the alternative? Given a legal problem, I must go to a lawyer, and I know what to expect from him.

GREENWALD: Do you know if he does it well?

REINSTEDT: No, but I also don't know about doctors, but I want a doctor when my appendix has to come out.

BRADDOCK: But there ought to be a way to establish the credentials of someone who practices the profession we're in. The current certification procedures aren't very good, although I suppose they're better than nothing. I still vote for an apprenticeship lasting several years, with exams along the way.

BEMER: There's a man in Arizona who wants to form a board of four or five reputable people who would administer a sort of Ph.D exam to people who would pay to become certified. The idea is that a day of interviewing and testing might be worth something.

GREENWALD: Isn't all this just a substitute for a programming aptitude test? Those were designed to save personnel departments some time and effort.

PARKIN: They turn out to be only IQ tests.

REINSTEDT: Not "turn out to be"; they were taken from IQ tests.

PARKIN: And what they reduce to is tests of how to solve increasingly more difficult puzzles.

REINSTEDT: I have no objections to the idea of certification. I know of no group that has a certification procedure that has abandoned it. If you'll tell me the specs (that is, what the certification is supposed to certify) I'll design you a program -- apprenticeship, plus tests, plus interviews, etc. -- that will do the certification.

GLASER: I think the present DPMA tests are better, for their numbers, than any of us would acknowledge. Clearly, those tests do not apply to numerical analysts, or scientific programmers, or the artificial intelligence boys; the tests just don't apply.

REINSTEDT: In analyzing the results of the last DPMA exams, they broke out those who were taking the test for the first time. Those who had majored in data processing in college came in second from the last (next to accountants), and under education majors, math majors, engineers, and everyone else, on the first two parts of the test. For the other parts, they were on the bottom.

GLASER: Sure they learned DP from numerical analysts and mathematicians. They didn't learn from people who had practical experience in the DP world.

BEMER: My only criterion is whether or not he can produce programs that work properly in the area I'm interested in. We put too much emphasis on the people involved (students and teachers) and not enough on the product we're seeking. The man may have superb qualifications and still write programs that put someone out of work, or ruin his credit rating.

BRADDOCK: Of course, but you can't tell that until after you hire him. We're seeking ways to minimize the risk of hiring the wrong man.

GLASER: Short of examining a man's work over a long period of time, there is no foolproof system. But consider the experience that San Francisco is having with the BART railroad. It is costing them millions of dollars because there is no group that can determine whether or not the control system works.

BEMER: And that's the same problem as the attempts to prove software correctness. There is no way to do it; it's worse than language translations.

PARKIN: It may be provably unprovable.

GLASER: We ought to be able to show that a given program does what it is supposed to do, but that's different from proving that what it does is adequate for the task. In the case of BART, the question is whether the control system can control 90 mph trains on 90 second headways.

PARKIN: And that can be tested, to any level of probability that you care to name.

GRUENBERGER: All this is charming, but totally irrelevant to the question, which was When will half the states require some sort of certificate? -- good, bad, or indifferent. You guys are all busy designing the perfect certificate, which isn't the point. It seems to me that if we have two more Equity Funding scandals within six months of each other, then about two months later more than half the states will require licensing of computer people, and they won't care how good it is.

PARKIN: A lot of doctors have killed their patients, but that is not the mechanism that led to the medical examining boards we now have. The medical profession decided to police itself, and quietly keep its mistakes from the public view.

GRUENBERGER: That only supports my statement. We ought to keep our mistakes to ourselves, too, and act to do it before it's forced on us.

GREENWALD: Us old people might have to protect ourselves from the young people.

KREHBIEL: Then you go on to restrict entry into the field, and you add grandfather clauses (in our case, literally).

GLASER: We joke about it, but in five years or so, the economic pressure on the 45-year-olds will be strong enough to make that more likely to happen than not.

KREHBIEL: We keep talking about a profession, but in terms of economic reality, it's not much different from auto mechanics. For most people, it's an 8 to 5 job: we are not self-policing: we don't have any of the attributes of a profession like law or medicine.

GLASER: We are closer, perhaps, to engineers, who are licensed.

PARKIN: There is a thing called CPE -- licensed professional engineer -- but no one pays much attention to it.

GLASER: But for certain tasks, you must employ one of them. We could do the same thing. For computing tasks that deal with public welfare, transportation, air traffic control, credit checking, and the like, you could be required to hire a LPP -- a licensed professional programmer. I don't think that it's too far away.

PARKIN: So you're saying that governmental bureaucracy will force it on us, however meaningless.

REINSTEDT: The people who are presently forcing it on us are groups like DPMA, ACM, and AFIPS.

GLASER: We would like to institute a certification program before the state imposes one on us that we would like even less.

REINSTEDT: That's understandable, but it will fall flat on its face if it has no more meaning than the present DPMA certification. You'd better certify something, rather than just certify.

GREENWALD: I disagree. If we establish any sort of certificate, the politicians will accept it as the basis for licensing, which the public will demand. (I voted for 1980.) For example, California will not tolerate many more election foul-ups like the last three we've had.

ARMER: And will certification change that?

GREENWALD: Certainly not, but it will permit the politicans to appease their critics.

PARKIN: Suppose we change the whole argument to apply to, say, mathematicians. Does it still follow?

GREENWALD: No; mathematicians don't affect me; they don't foul up my bills, or mess up my traffic. Whatever they do is completely hidden from view. The worst damage they can do is to students, and that is limited to producing another (poor) mathematician.

This morning's paper had an item about a man who had tried for 18 months to get the courts and the Department of Motor Vehicles to admit that he paid a fine for which he has a receipt. My point is that the public is becoming well aware of computer foul-ups.

PARKIN: And how many certified programmers would it take to solve that?

GRUENBERGER: It's completely irrelevant as to whether it solves anything, or even does any good. The question was, when is it coming?

NO

BEMER: Speaking of mathematicians, I have finally succeeded in installing a project, after 9 years of screaming, to validate all of IFIPS' mathematical subroutines.

GRUENBERGER: I predict that after 4 years they will have been able to certify perhaps 3 subroutines.

REINSTEDT: I guess the answer to the question is that we will have certified programmers pretty soon, and it will be meaningless.

GRUENBERGER: The question was <u>when</u>?

BRADDOCK: I voted for a late year, when it might mean something.

GREENWALD: I would now vote much earlier. The politicians will say "We recognize the problem, and we have done something about it."

KREHBIEL: If it takes them as long to recognize this problem as it did the oil shortage, we have a lot of time.

GLASER: The people in DPMA who run the certification program know that it isn't as good as it ought to be; that it needs fixing; and that they acknowledge that it needs fixing. Few people would defend it as the ultimate.

BEMER: My motto in computing has always been these five words: <u>Do</u> <u>something</u> <u>small</u> <u>useful</u> <u>now</u>.

REINSTEDT: The certification boys are about to do something large useless now.

ARMER: I wonder what percent of our industry will ante up $35 to take that test. I really think it will be a financial flop.

GRUENBERGER: I wouldn't take it -- at any price -- unless forced to.

KREHBIEL: They'll have to take me kicking and screaming. I probably couldn't pass it. I've forgotten what an X punch is.

GREENWALD: One of the versions of that test that I looked at was loaded with statistics, so I could never pass it.

PARKIN: Has anyone here ever hired someone on the basis of his holding the Certificate in Data Processing; that is, where it was the deciding factor?

KREHBIEL: You never really have a case where "all other things are equal," but I did hire a man who had the CDP and the fact did impress me. At the time, my choices included a recent graduate of Control Data Institute, a couple of aerospace dropouts, and others.

PARKIN: So collectively we have one case out of perhaps hundreds of hirings. It can't be too important. We're discussing a subject from a standpoint of massive ignorance.

GRUENBERGER: It's not surprising that we feel we couldn't pass those exams. Most practicing doctors couldn't pass the current medical exams. Most competent faculty members couldn't pass the exams given by other members of their own department. It's true in any discipline, and it's probably a function of age more than anything else.

ARMER: At Stanford, they looked at the medical exams of five years earlier and concluded that half of the answers thought to be correct then were now wrong.

GLASER: The recommended treatment may have changed, or the medicine of choice may have changed, but that's different from saying that the earlier answers were wrong.

GRUENBERGER: Let's turn to item 4: when will virtual storage be an accomplished fact? As worded, the question said "all installations" and it should have been qualified with "large." The responses I got had a very wide range.

GREENWALD: Another way to state it is: when will it be difficult to market a machine without virtual storage, and that's today.

GRUENBERGER: Well, that seems to be the consensus, then, with the qualification put in. Let's discuss No. 11: adherence to standards.

BEMER: I see signs that IBM's traditional, or hereditary, opposition to standards is attenuating.

GREENWALD: For things like character representation, there are two standards: the external way that characters are printed, and the internal way they are represented in the hardware. For the latter, I think we've come a long way, and for the former, I don't think it's all that important.

BEMER: One idea is to take any keyboard that you want to use, as long as the characters that are entered are homomorphic with any other set. If you like the Dvorak keyboard, you just overlay it on any keyboard you want, keeping the internal representation homomorphic.

GREENWALD: We are busy counting the number of different standard ANSI-labelled tapes, and there are at least 149 of them. This makes for a terrible problem, to be able to print out any given tape.

BEMER: There's no way around it, short of recording a description on the tape itself.

PARKIN: How can 149 different things all be "standard?"

GREENWALD: Put it this way: various things that have been produced by manufacturers as labelled tapes that supposedly adhere to ANSI standards add up to 149.

GRUENBERGER: We have two different "standard" keyboard layouts, neither of which are used on any existing keyboard.

PARKIN: Then the "standards" are really rather fuzzy.

GREENWALD: Yes, or people have compromised in the name of expediency, or something; I really don't know.

GRUENBERGER: There was a review of one of my books published in which the reviewer's chief criticism was that every program in the book had the oh's slashed except for one that slashed the zeros (that one being a photo reproduction from a Teletype). We've been fighting that battle for 25 years.

PARKIN: And we will for 25 more. But look: we don't even have standardized English. The only standards we get are de facto, and only those that are dictated by economics. For example, we got standard rail widths on the railroads because it was cheaper to do that than to unload and load cars.

GREENWALD: From an economic point of view it may well be that the technology of the future will dictate that you don't want to standardize. Most vendors are working on word processing systems. The one who captures this market will have a set of standards that he hopes no one knows about.

BLOCH: But standards exist to make it possible to communicate with ease between different makes and different manufacturers.

BEMER: Or between different models of one manufacturer; IBM, for example.

GRUENBERGER: I don't see why it isn't economically advantageous for someone making a new device with a keyboard to use the same layout as some existing keyboard. Why is it felt necessary to design a new layout?

PARKIN: It's considered a sales advantage to be able to point out that the layout is different, since that suggests that it's also better for you. But look what happens when you force standards too soon. In this country, we did just that with the scanning standards for TV, and the Europeans, who standardized much later, could do it more sensibly. I suggest that our industry, for all its explosive growth and pervasiveness, may not be mature enough to freeze too many standards.

GREENWALD: Quite apart from considerations of anti-trust and monopoly, it might not be to IBM's advantage to promote communications standards, in that it would unduly foster the plug-compatible market.

GRUENBERGER: I don't follow all this. Take those TV standards. If you increase the number of lines, you need more band-width, and then you must have less stations. We made a swap, and opted for an acceptable clarity consonant with so many stations. What's wrong with that?

PARKIN: There are already too many stations.

GRUENBERGER: But now you're criticizing programming quality. Better picture quality won't cure that. It seems to me that if we hadn't frozen the standards when we did, the TV industry would still be struggling to get started. From their point of view, the early standards helped to produce a vast industry much sooner. Would you prefer having only two stations to pick from, both with superb quality of pictures, but both dull?

But in computing, we seem to have only two standards that are rigidly adhered to: the dimensions of the 80-column card, and the width of magnetic tape. After 25 years, can't we do better than that?

GREENWALD: I'm with Tom: the only real motivation is economic. As long as we can sell machines that are incompatible with other people's, we will. As soon as we can't sell them, we'll see the beauty of conforming to some standard.

GRUENBERGER: The flaw in that argument is that you never know how many you would have sold if you had standardized.

GLASER: But they're selling enough.

GRUENBERGER: Are they? Would any vendor besides IBM agree to that?

GREENWALD: As long as the stock keeps going up at a satisfactory rate, we keep our jobs.

BEMER: I'm in favor of standards as long as they're not restrictive. Long ago, for example, I argued that card readers shouldn't be restricted to being able to read only a limited number of combinations of holes in a column, like the 53 or so for the 705 card reader. I reasoned then that the card reader should be able to read any possible combination of holes. I want flexible standards, that go in multiples. The standards should match the technology and leave room for innovation.

PARKIN: It's sort of like saying that you can use any number, but it must be a multiple of 10. I don't call such things standards; I call it quantification. When the economics and the technology have achieved an appropriate balance, there comes a time when de facto standards become de jure. For most things in our industry, that time has not yet come.

BEMER: We may almost be there to resolve the conflict between ASCII and EBCDIC.

PARKIN: We cannot yet agree on whether to use 1's complements or 2's complements, but that doesn't bother me too much.

BEMER: But that isn't the level of standards you need.

GREENWALD: We need a standard for the level of standards you need.

GRUENBERGER: I hear people saying "We should standardize on the things I want, but other things are unimportant." It sounds like what I've been hearing for 25 years (about such minor things as zero-slashing, flowchart diamonds, and the like): "It would pay us to standardize, so let's all standardize my way." If you standardize, then someone has to yield some freedom, but everyone should gain in the long run.

ARMER: I hear people saying "The forces of the marketplace will take care of everything. If it's good, it will happen." The effects of standards are a public good, not a private good, and the marketplace doesn't work worth a damn.

GRUENBERGER: I agree. In such a simple thing as phonograph records, we have four standard speeds. You may want to play only 33-1/3 speed records, but you pay for having your turntable operate at four speeds.

GREENWALD: How about the standard for recording level?

GRUENBERGER: That's much worse. If you transcribe from records to tape, you have to adjust the gain for every record, even for two records from the same company the same year. There is virtually no standard at all. All this lack of standardization costs the consumer money.

It took the auto industry 25 years (with some Federal prodding) to standardize the positions on the shift lever.

But we're in computing; we're the clever and intelligent people in the country, it says here. And we can't standardize even the simplest things.

BRADDOCK: At least we're clever enough to recognize all our problems; we just don't seem to want to solve them.

PARKIN: Are there any standards that are legally enforceable?

BEMER: No, except for those that become incorporated into civil codes, such as building codes. We're the only civilized country that operates that way. The American National Standards Institute talks constantly about voluntary standards, but they're as archaic as the dodo.

GREENWALD: And if voluntary standards don't work, then the only alternative for obtaining them is government control.

PARKIN: Every two or three years, someone revives the Dvorak typewriter keyboard and touts it as better than the Sholes layout for various reasons of efficiency in touch typing. Nevertheless, the Sholes arrangement is the de facto standard (in fact, it has been called the "universal" keyboard) and has been for over 50 years, and it will stay that way for as long as we can predict.

GRUENBERGER: The current interest in the Dvorak arrangement comes from the widespread use of electric typewriters and other keyboard devices, where the keys are only switches and are not mechanically connected to the printing mechanism. It is now feasible to have the Dvorak layout on a machine made for the Sholes layout, and have the two schemes switchable.

I guess what really bothers me about our avoidance of standards in computing is that so many other industries are loaded with standards, and they work. The U.S. Pharmacopoeia is just a compilation of standards, and they're legally binding. If you sell "USP aspirin," then it better agree with the USP standard. In the physical hardware business, every gadget made must conform to the standards for screw threads, wire sizes, and so on. And no one would think of deviating from those standards. When you buy a No. 2 pencil, anywhere, you expect it to be the standard No. 2 hardness, and it is. You can buy film for your camera anywhere in the world and it will fit. But when you buy a computer, look up its collating sequence, because it will be different from the collating sequence of your last machine, and for no good reason. I think, for smart people, we come off looking like idiots.

PARKIN: It seems that the existing standards are all analog: the placement of the keys on a keyboard is a digital problem.

GRUENBERGER: The placement of the positions on a gearshift lever is digital, and we've managed to standardize that. All I'm saying is, couldn't we find things in our field -- say, one a year -- that could be standardized? and by that I mean accepted and observed. ANSI declares standards like crazy, but no one pays any attention to them unless they're forced to.

PARKIN: If the government requires adherence to a declared standard as a condition for purchase, then the vendors are economically motivated to observe the standard.

BEMER: That's another reason why IBM will use ASCII in their next line.

GRUENBERGER: Your arguments are all sound. Now apply them to the phonograph record speed situation and tell me why we still have 3- and 4-speed turntables.

ARMER: It was relatively cheap to make a gadget that would play all 4 speeds.

GRUENBERGER: Like hell it was. When the microgroove records came out, both vendors made players for their speed only. It cost a fortune to engineer 4-speed players that would maintain the 4 speeds reasonably well. As soon as you decide to go beyond one speed, then extra speeds are cheap to add; the big jump is the one from one speed to any number more, and that was expensive, and we've all paid for it.

KREHBIEL: They had to go to at least two speeds, since they were struck with 78. Adding a third speed, then, was pretty trivial.

GRUENBERGER: Not at all. You had to add the turnover cartridge, and make the thing work for two (or three) sizes of records, too. With just a little thought, and cooperation, and concessions, a lot of grief could have been avoided. But my point is, they were dumb, and we're supposed to be smart, so why can't we exercise some of this smartness?

REINSTEDT: Just as an example, why don't we produce one good glossary in our field?

BEMER: What would be the point, when the bible we work from (Webster's Third International) has abandoned logic completely?

GRUENBERGER: That's the bible you work from; I work from the Second edition.

BEMER: But when I do that, people scream.

GRUENBERGER: Which way do you want to argue it? Do you want to be logical, or go the way people scream?

BEMER: See my next editorial. I've decided to go with logic.

KREHBIEL: I'm glad we have one good standard, and that's IBM-compatible tape, so you can move your data base from system to system freely.

BEMER: No way! You must be kidding.

GREENWALD: How about the standards for flowchart symbols, even including the correct height-to-width ratio for the rectangles?

PARKIN: We can't even get agreement on the necessity for having flowchart symbols at all, much less standardized ones.

GREENWALD: That's right; I don't agree that they're required. But if I'm forced to use flowcharts, I might as well use standard symbols.

BEMER: If we didn't use flowcharts, we might have a higher probability of getting working programs.

BRADDOCK: If each university instructor teaches his students certain standards, then eventually you get those standards to spread.

GERALD: Except that every instructor has a different set of standards (or none at all).

REINSTEDT: What happened to Curriculum 68?

GRUENBERGER: You've got me; that is one more standard in our field. Every publisher has that thing memorized cold. They won't even look at a book that doesn't fit one of the little numbered boxes on the display that was in Curriculum 68. Right now they're going crazy, because there's a rash of books on Social Effects of Computers, and there's no little box for that one.

Appendix A

ATTENDEES

PAUL ARMER, Center for Studies in the Behavioral Sciences

ROBERT BEMER, Honeywell Information Systems

ERICH BLOCK, IBM

FRED BRADDOCK, Informatics Inc.

PROF. CURTIS GERALD, Cal Poly State University, San Luis Obispo

GEORGE GLASER, AFIPS

IRWIN GREENWALD, Xerox

PROF. FRED GRUENBERGER, Cal State University, Northridge

DON KREMBIEL, Santa Monica City College

TOM PARKIN, Control Data Corporation

ROBERT REINSTEDT, The RAND Corporation

Appendix B

For each of the items 1-12, a statement is made about a future situation. For each item, indicate in the margin one of the following:

A. We're already there.
B. It will occur in _____ (year).
C. It will happen, if at all, after the year 2000.
D. (Some other conclusion.)

1. Half the computing power of the U. S. will reside in what are now called mini computers. (Mini defined as in the July Datamation article, and power defined in terms of addition times per second.)

2. PL/I will be as dead as ALGOL is (in this country) in 1973; i.e., no vendor will boast of offering PL/I as a language.

3. APL will be as dead as ALGOL is (in this country) in 1973; i.e., no vendor will boast of offering APL as a language.

4. User applications based on the use of large virtual storage (e.g., several million words) will have become the accepted mode of operation, in the sense that more than half of all installations will be using it. (Think of the typical data processing customer, and do not intersect with question 1.)

5. The world's chess champion will be a computer program.

6. Language translation (from natural idiomatic language A to natural idiomatic language B) by machine will be economically feasible.

7. Computer programs will be capable of automated fingerprint recognition.

8. Generalized voice recognition by computer will exist.

9. Computing (data processing, information processing, or what you will) will be a standard subject in high schools, in the same sense as algebra is today.

10. The number of installed computers, of all sizes, in the U.S. will be 500,000. (Meaning programmable, general purpose machines, for which the estimate as of mid-1973 is around 84,000.)

11. Things like representation of characters (e.g., the encoding of the = sign on the 026 and 029 keypunches), collating sequences, and keyboard layouts will be standardized (that is, accepted and observed standards--not simply declared standards). Note that the controversy over zero-slashing and oh-slashing is still going on; and that we have declared standards for keyboard layouts, neither of which is followed on any keyboard.

12. More than half the states will have some system of licensing for computists (i.e., the "Certified Public Programmer" concept).

Appendix C

ADDITIONAL AGENDA ITEMS

F. What is your opinion of the future role of programmers? Will they be required in the same numbers as today? What skills will they need? What salaries will they command? What degree of training and education will they be required to have?

G. When is 1984? That is, when will there be centralized data banks of extensive information on individuals: to whom will this information be made available and how readily?

H. What is the trend in the battle between large, centralized computers vs. distributed computing?

J. What will be the trend in responsiveness to the needs of the individual user: will software and hardware designers continue to go their merry ways, ignoring what the ultimate user really needs and wants?

Appendix D

Distribution of the opinions of 15 experts, 11/1/73

1. Mini computers

LOW:	1975
MODE:	1980 (7)
HIGH:	After 2000

2. PL/1

LOW:	Now (1)
MEAN:	1988
HIGH:	Never (1)

3. APL

LOW:	Now (2)
MODE:	1980 or after 2000
HIGH:	Never (1)

4. Virtual storage

LOW:	1973
MEAN:	1980
HIGH:	After 2000

5. Chess champion

LOW:	1980 (2)
MODE:	1995
HIGH:	Never (2)

6. Language translation

LOW:	1980 (1)
MEAN:	1999
HIGH:	Never (2)

7. Fingerprint recognition

LOW:	Now (1)
MEAN:	1983
HIGH:	After 2000 (1)

8. Voice recognition

LOW:	1976 (1)
MEAN:	1989
HIGH:	Never (1)

9. High school curriculum

LOW:	1978 (2)
MEAN:	1995
HIGH:	After 2000

10. 500,000 machines

LOW:	1978
MEAN:	1988
HIGH:	After 2000

(Interesting. If just the 1973 rate of installation maintains, we will have 500,000 machines installed by 1983.)

11. Standards

LOW:	1985
Consensus:	After 2000, or never

12. Certified Public Programmers

LOW:	1980
Consensus:	After 2000, or never.

slation may
or by EDP
Guidelines
iich can be
elpful at the
that drafts
subsequent

re intensive
ge that one
oss national
rather than
on "Legisla-
Council for
ed to submit
nce and ob-
in different

oped, some
a result of
e of organi-
itself as an
ith informa-
e or already
be linked to
entration of

JURIMETRICS J.

COMPUTERS AND OUR SOCIETY[*]

Robert W. Bemer[†]

Computer usage is classified as either (1) advisory, (2) leading to decisions by humans, or (3) with decisions being taken by a preprogrammed computer unless countermanded in time. Some examples of difficulties even in the first two categories imply that caution in the third is imperative. The computer technology learned from the space effort is not yet transferred to the bulk of computer usage.

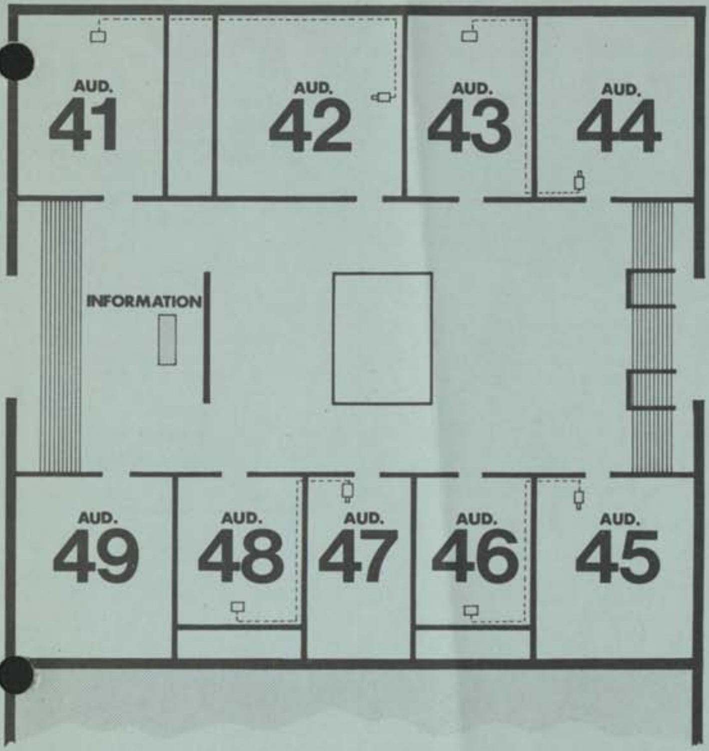
Both legal and voluntary (professional) measures against misuse are discussed.

INTRODUCTION

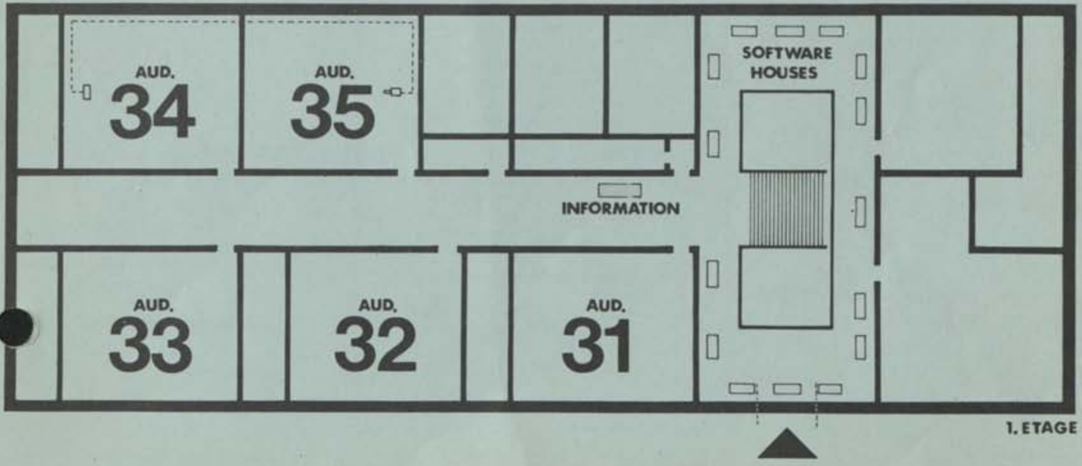
In 1950, after my "graveyard shift" at the RAND Corporation, I was still working at 0830 on a 604 board to take an 8-digit square root of an 8-digit number (until then not accomplished mechanically for that equipment). A round little man approached and asked what I was doing. I told him. He then asked about the calculator, and as I answered each question the next one got more difficult and penetrating, until I was really straining every faculty to answer correspondingly. He did not introduce himself, but I found out later that day that it was John von Neumann.

Naturally the incident remains very clear in my mind. I recall that he did *not* leave me saying "Use the tool well for the social benefit of mankind," or anything else in this vein. There were very few men in the computer world or business then that were considering social ramifications of this sort. Ed Berkeley was, and remains, an exception. To most of us it was just a time of freeing the mind to go far beyond our previous

302


AUD. 41
AUD. 42
AUD. 43
AUD. 44
INFORMATION
AUD. 49
AUD. 48
AUD. 47
AUD. 46
AUD. 45

306


AUD. 34
AUD. 35
SOFTWARE HOUSES
INFORMATION
AUD. 33
AUD. 32
AUD. 31
1. ETAGE

A·1 Bygning 302 Auditorium 49

NATIONAL PLANLÆGNING AF DATABEHANDLING

Emnegruppeleder:
Forsker Svein A. Øvergaard
Regneanlegget Blindern-Kjeller,
Norge

ONSDAG

9.00 **Indledning**
Forsker Svein A. Øvergaard

9.15 **»Planlegging for databehandlingen i forvaltningen«**
Systemleder Arne E. Hilmen,
Statens Rasjonaliseringsdirektorat, Norge

9.45 **»Bakgrunden till utformning av en svensk datapolitikk«**
Fil. lic. Tomas Ohlin,
Datautredningen, Sverige

10.45 **»Finsk datapolitikk«**
Industrirådet, tekn. dr. Pekka Malinen, Handels- och Industriministeriet
Finland

11.15 **»Synspunkter på en statlig datapolitik«**
Direktør, cand. oecon.
Hans Øhrstrøm,
Formand for EDB-Rådet.
Danmark

A·2 Bygning 306 Auditorium 34/35

MAN &
MENNESKET OG EDB-SYSTEMER

Emnegruppeleder:
Lektor, lic. tech. Klaus Kjøller,
Driftsteknisk Institut,
Danmarks tekniske Højskole,
Lyngby, Danmark

TEMA I: OBSERVATION OF INTRODUCTION
Iagttagelser af organisatoriske konsekvenser af indførelse af datamatbaserede administrative systemer.

ONSDAG

9.00 **Hvorledes har indførelsen af et større automatisk databehandlingssystem påvirket de uformelle forhold i organisationen«.**
Kontorchef, cand. jur. & art.
Bernard Bévort,
Kjøbenhavns Telefon Aktieselskab, Danmark

10.00 **»Organisationens forventninger til et integreret informationssystem«**
Systemchef Jørgen Vorsholt,
Højgaard & Schultz A/S,
København, Danmark

11.00 **»Er datamaten katalysator for organisationsudvikling?«**
Direktør J. Meiland Hansen,
A/S LK-NES, Danmark

13.30 **Paneldiskussion tema I:**
Ordstyrer:
civiling., lic. techn.
Thomas Skousen
Paneldeltagere:
temaets foredragsholdere samt systemchef Per Illum, AGA A/S,
København, Danmark

TEMA II: CONSIDERATIONS
Sociale hensyn ved systemkonstruktionsarbejdet.

TORSDAG

9.00 **»Informationssystemer og job design«**
Cand. merc.
Niels Bjørn-Andersen,
Institut for organisation, Handelshøjskolen, København,
Danmark

10.00 **»Personalmotivering vid utveckling av informations system«**
Systemchef, fru Marita Pättö,
Oy W. Rosenlew, Björneborg,
Finland

11.00 **»Decentraliserat systemarbete och medbestämmande rät«**
Fil. kand. Nils Fredholm, Skandia,
och fil. kand. Peter Revay,
Stockholms Universitet, Sverige

13.30 **Paneldiskussion tema II:**
Ordstyrer:
lektor, lic. techn. Klaus Kjøller
og cand. pæd. psyck. Svend Jørgensen
Paneldeltagere:
temaets foredragsholdere

A·2 Bygning 306 Auditorium 34/35 (fortsat)

TEMA III: POSITION
Systemkonstruktørens holdning og synspunkter. Del A:
& VIEWPOINT

14.30 **»Informationsteknikkens magthavere«**
Organisationschef Jan Bendix,
Bang & Olufsen A/S, Danmark

TEMA III:
Systemkonstruktørens holdning og synspunkter. Del B:

FREDAG

9–12 **»Databehandling, planlegging og styring, set fra de fagorganisertes synspunkt – et forskningsprosjekt«**
Forskningschef
Kristen Nygaard og
klubsekretær Svein Ulleberg,
Kongsberg Våpenfabrikk,
Norge
og Olav Terje Bergo,
Norsk Regnesentral
Diskussion av prosjektet
»Om möjligheterna av en alternativ utveckling av datatekniken«
Jan Annerstedt,
Universitet i Lund, Sverige
Forskningspolitiska programmet och statsvetenskabeliga institutionen,

TEMA IV: & THE HUMAN
Datamaten og den menneskelige tilværelse

13.30 **»Leder deduktiva informationssystem till en administrativ revolution?«**
Forskarass. Sten-Åke Tärnlund,
Institutionen för informationsbehandling, KTH/SU, Sverige

14.30 **»Computers and Our Society«**
Staff Consultant
Robert W. Bemer,
Honeywell ~~Information~~ Systems,
USA

15.30 **»Information Networking is a medium – what is the massage?«**
Project leader Michel A. Godard,
Compagnie Honeywell Bull,
Paris, Frankrig

TO NAUR

A·3 Bygning 302 Auditorium 43/44

KVALITETSPROBLEMER

MÅLEPROBLEMER - A.3.1.
Emnegruppeleder:
Konsulent, lektor Poul Sveistrup,
Københavns Universitet, Danmark

ONSDAG

9.00 **»Systemkvalitet genom styrning«**
Systemchef Håkan Karlberg och programmeringschef Henk Bremer, Uddeholms Aktiebolag, Sverige

10.00 **»Vurdering av systemets godhet«**
Cand. real. Trond Thue, Sentralinstitutt for industriell Forskning, Oslo, Norge

11.00 **»Kvalitetsmåling – økonomi eller politik?«**
Diskussion ledet af konsulent, lektor Poul Sveistrup

13.30 **»Kvalitestkontrol av software«**
Siv. ing. Harald Lindvik, A/S Computas, Oslo, Norge

14.30 **»Systemflexibilitet med moderne ADB-teknik«**
Civilingenjör Lars Henningsson, ADB Utveckling AB, Stockholm, Sverige

15.30 **»Lönsamhetsbedömningar vid systemarbete«**
Ekon. lic. Mats Glader, Umeå Universitet, Umeå, Sverige

TORSDAG

9.00 **»A functional Approach to System Performance Evaluation«**
Senior System Engineer Walter O. Bailey Jr., Honeywell Information Systems, Inc., Phoenix, USA

10.00 **»Prestandamätning i komplexa datasystem«**
Civilingenjör Hans Ljunggren, Saab-Scania AB, Linköping, Sverige

11.00 **»Driftsäkra dataudrustningar – är det dyrbart?«**
Sektionschef Jan Frånlund, Telub AB, Växjö, Sverige

13.30 **Paneldiskussion: »Focus på kvalitet«**
Introduceret og ledet af konsulent, lektor Poul Sveistrup
Diskussionsindledere:
Utviklingssjef Peter Hidas, Honeywell Bull A/S, Norge, og professor Christian Gram, Danmarks tekniske Højskole, Danmark

A·3 Bygning 302 Auditorium 43/44

(fortsat)

SIKKERHEDSPROBLEMER - A.3.2.
Emnegruppeleder:
Afdelingsdir. O. Cilius-Nielsen, IBM A/S
Formand for styrekomiteen for EDB-Rådets sikkerhedsprojekt, Danmark
EDB-RÅDETS SIKKERHEDS-PROJEKT
– Sikkerhedsforanstaltninger i og omkring EDB-installationer

FREDAG

9.00 **Datasikkerhed**
Afdelingsdir. O. Cilius-Nielsen, IBM A/S, Danmark
»Fysisk sikring af lokaler m. v.«
Økonomichef E. V. Rasmussen, I/S Datacentralen af 1959, Danmark

»Fysisk adgang til lokaler – EDB installation, Data og programmel«
Direktør Finn Berentsen, Dansk Dataservice A/S, Danmark

10.00 **»Sikring af Data«**
Underdirektør Henning Madsen, Den Danske Landmandsbank A/S Danmark
»EDB-personalet – Dataetik«
Skoleleder Henning Andersen, Siemens Aktieselskab, Danmark

11.00 **»Brandforebyggelse og -bekæmpelse«**
Direktør Mogens Boman, EDB-Rådet, Danmark

PRAKTISKE EKSEMPLER FRA SVERIGE OG FINLAND

13.30 **»Datacentralens fysiska säkerhet«**
1. sekretær, driftschef Walter Holmer, Stockholms Läns Landsting, Sverige

14.30 **»Security of the Production-process in a Computer Centre«**
Department Chief Juhani Ryhänen, Finnish State Computer Centre, Finland

»Security of an ADP-system«
Assistant Department Manager Kyösti Hallikainen, Finnish State Computer Centre, Finland

15.30 **»Backup – Typesetting«**
Dipl. ing. Timo H. A. Koski and dipl. ing. Annikka Tukkanen, Technology, Finland

A·4 Bygning 302 Auditorium 49

VÆKSTENS GRÆNSER

Emnegruppeleder:
Forsker Svein A. Øvergaard, Regneanlegget Blindern-Kjeller, Norge

ONSDAG

13.30 **»Vekstens grenser . . .«**
Forsker Svein A. Øvergaard, Regneanlegget Blindern-Kjeller, Norge

14.30 **»The limiting factors of computer architectures«**
Director Frank Holst, Sperry Univac, USA

15.30 **Paneldiskussion**
Indleder:
Jacob Palme, FOA P, Stockholm, Sverige

B·1 Bygning 302 Auditorium 47/48

ANVENDELSER I INDUSTRIEN

Emnegruppeleder:
Overingeniør Gunnar Holmdahl,
ASEA, Avdeling KD, Västerås,
Sverige

PRODUKTIONSSTYRING I

ONSDAG

9.00 **»Nordforskprojektet TEPS«**
Indledning ved
professor B. Bertil Colding,
Kungl. tekniska Högskolan,
Inst. för mekanisk teknologi,
Stockholm, Sverige
»Automatisering av beredning och schemaläggning för verkstadsindustrien«
Tekn. lic. Björn Svärdson,
Kungl. tekniska Högskolan,
Inst. för mekanisk teknologi,
Stockholm, Sverige

10.00 **»Direkte afsøgning anvendt ved optimering af fabrikationsprocesser«**
Civ. ing. Jørn Christoffersen,
Stærkstrømsafdelingen – DtH,
Lyngby, Danmark
»Implementering av ett detaljplaneringssystem inom metallbearbetningsindustrien«
Dipl. ing. Erkki Suonsivu
Outokumpu Oy, Björneborg,
Finland

11.00 **»Formulering af en generel tids- og ressourceallokeringsmodel i APL«**
Civ. ing. Peter Falster,
Stærkstrømsafdelingen – DtH,
Lyngby, Danmark
Diskussion af TEPS

PRODUKTIONSSTYRING II

13.30 **»ADB-baserat körplaneringssystem – fyra års erfarenheter«**
Pol. mag. Dag Svernlöv,
Saab-Scania AB,
Dataservicesektorn,
Linköping, Sverige
»En metod för körplanering«
Civ. ing. Hans Walfridsson,
Saab-Scania AB,
Dataservicesektorn,
Linköping, Sverige

14.30 **»AUTOPROS 2 – AUTOmatisk PROSessplanlegging av plate- og sveisearbeide«**
Siv. ing. Hans Jørgen Haug,
NAKK, Økern – Oslo, Norge

SPECIELLE ANVENDELSER

TORSDAG

10.00 **»Integrated systems for the construction industry«**
M. F. Bott,
SPL Svenska AB, Stockholm,
Sverige

11.00 **»IBMsystem/7-2790 för laboratorietest vid Munksjö Pappersbruk i Jönköping«**
Fil. kand. Leif Svensson,
IBM Svenska AB, Stockholm,
Sverige

B·1 Bygning 302 Auditorium 47/48

(fortsat)

TERMINALER, DATABASER

13.30 **»Implementering av bill of materialstrukturer med DL/1«**
Civ. ing. Börje Agnvall,
IBM Svenska AB, Stockholm,
Sverige

14.30 **Datainsamling i industrin«**
Ing. Clas Palmberg,
Oy Nokia Ab Elektronik,
Helsingfors, Finland

15.30 **»Direktåtkomst av produktinformation via terminaler«**
Civ. ing. Lennart Åberg,
IBM Svenska AB, Stockholm,
Sverige

ØKONOMI

FREDAG

9.00 **»Ekonomisystem för flygdivisionen, data- och elektronikdivisionen samt koncernstaber inom Saab-Scania AB«**
Bengt Thörnblad,
Saab-Scania AB,
Dataservicesektorn, Linköping,
Sverige

10.00 **»Ekonomisystem för flygdivisionen, data- och elektronikdivisionen samt koncernstaber inom Saab-Scania AB. Insamling, kontroll och lagring av transaktioner«**
Lars Kinell,
Saab-Scania AB,
Dataservicesektorn, Linköping,
Sverige

11.00 **»Ekonomisystem för flygdivisionen, data- och elektronikdivisionen samt koncernstaber inom Saab-Scania AB. Kostnadsrapportering till linje- och projektorganisationer«**
Karin Gruvin,
Saab-Scania AB,
Dataservicesektorn, Linköping,
Sverige

LØNSOMHED

13.30 **»Lønnsomhetskriterier for operativt styringssystem i produksjonsbedrifter«**
Applikasjonschef
Georg Tidemann-Andersen,
Honeywell Bull A/S, Oslo, Norge

14.30 **Open house eller paneldiskussion**
Lønsomhed af EDB indenfor
industrien
Indledning:
»Industrien og edb, status og udviklingsveje for en rationel edb-anvendelse«
Civ. ing. Torben Dybkjær,
IBM A/S, Lyngby, Danmark

B·2 Bygning 302 Auditorium 49

SYGEHUSE OG OFFENTLIG FORVALTNING

Emnegruppeleder.
Direktør Mogens D. Rømer,
Kommunedata I/S, København,
Danmark

TORSDAG

9.00 **»Integrert terminalsystem ved hospital«**
Siv. ing. Ove Lange,
A/S Norsk Data-Elektronikk,
Norge

10.00 **»Administrative/styringssystem innen helsesektoren«**
Siv. ing. Olaf M. Engelhardtsen,
Hartmark & Co. – IRAS,
Oslo, Norge

11.00 **»Electronic Data Processing – Impact on Future Medical Care«**
Civ. ing. Carl Cederlund,
IBM Medical Industri Center,
Sverige

13.30 **»KOMPAS – Kommunalt planeringssystem«**
Fil. mag. Ove Salomonsson,
Planeringssystem AB,
Saltsjö-Boo, Sverige

14.30 **»'Projekt database'«**
Konsulent Børge Hastrup,
Kommunedata I/S,
Aalborg Centralen, Danmark

B·3 Bygning 306 Auditorium 32

BANK OG FORSIKRING

Emnegruppeleder:
Direktør Kjell Hultman,
Sveriges Kreditbank,
Stockholm, Sverige

ONSDAG

9.00 **»The real time service for many users«**
Equipment chief Kari Saarto,
Tietotehdas Oy, Kilo, Finland

10.00 **»Aktiv verdipapirforvaltning ved EDB«**
Direktør Leif Åsenden,
Hartmark & Co. – IRAS,
Oslo, Norge

11.00 **»Fault handling in the central site of Pohjoismaiden Yhdyspankki's real time system«**
Master of philosophy
Esko Mäkinen,
Tietotehdas Oy, Kilo, Finland

13.30 **»Management Techniques for Developing Large Systems«**
Mr. Derek Bandy, MBCS,
Technical Controller,
Leasco Software Limited,
London, England

14.30 **»Computers in banking«**
Deputy manager A. R. Wild,
SPL Svenska AB,
Stockholm, Sverige

(LOST)

15.30 **»4 års praktisk erfaring med realtidssystem i sparekassen Bikuben med ca. 100 real-tids kassesteder«**
Underdirektør Jørgen Almer,
Bikuben, København, Danmark

TORSDAG

9.00 **»'In-house'-terminaler i KOP:s reeltidssystem«**
Dipl. ing. Robert Hoge,
Kansallis-Osake-Pankki,
Helsingfors, Finland

10.00 **»Totalinriktad förstudie avsedd att betjäna planeringen av ADB-system utvecklingen på lång sikt«**
Fil. mag. Kurt Lindgren,
Sparbankernas Central-Aktie-Bank, Helsingfors, Finland

11.00 **»Budgetsimulering inom Skandia«**
Fil. kand. Leif Norrby,
Försäkrings-aktiebolaget
Skandia, Sverige

B·3 Bygning 306 Auditorium 32 (fortsat)

13.30 **Paneldiskussion over emnet »Morgondagens betalingssystem«**

- konkurrensen mellan bankerna
- samverkan mellan bankerna
- kundernas krav på service – nya tjänster
- bankernas skyldighet att sköta betalningsförmedlingen rationellt med risk för att annars andra intressenter bygger betalningssystem
- långsiktig samverkan för standardisering mot bakgrund av ovannämnda frågeställningar
- definationer av standardiseringsområden

Ordstyrer:
Direktør Kjell Hultmann,
Sveriges Kreditbank,
Stockholm, Sverige

Indleder:
direktør Ivan Ekebrink,
SE-banken, Stockholm, Sverige

Deltagere:

Danmark:
direktør Alf Bagge-Petersen,
Kjøbenhavns Handelsbank,

Norge:
direktør Rolf W. Erichsen,
IDA, Oslo
Direktør Terje Linder Andresen,

Finland:
ADB-chef Jussi Tuori,
Kansallis-Osake-Pankki,
Helsingfors
direktør K. Andersson,
Nordiska föreningsbanken AB,
Helsingfors

B·4 Bygning 302 Auditorium 49

ANDELSANLÆG – TOTAL FACILITIES MANAGEMENT

Emnegruppeleder:
Direktør, dr. John Gunn,
RECKU, København, Danmark

Sessionen søger at belyse lighedspunkter og løsninger i de daglige driftsproblemer i servicecentre med

- forskellige ejerforhold og dermed politikansvarlige (fra privat ejede virksomheder via andelsanlæg til offentlige centre)
- forskellige formål (fra at tjene penge til at assistere undervisning og forskning)
- forskellige tilbud (fra rå regnekraft til Total Facilities Management)

FREDAG

DATAKRAFT
Centret stiller datamater (operatører og terminaler) og operativsystem til rådighed.

9.00 Driftschef A. Dahlstrand, Lunds Universitets Datacentral, Sverige

DATAKRAFT + PROGRAMMERING
Centret assisterer også med programmering af specielle opgaver og standardprogrammer.

9.15 **EDB på andelsbasis**
Salgsdirektør Poul Thornberg, HD,
SAAB A/S, København,
Danmark

9.30 **DISKUSSION**

10.00 **Total Facilities Management – i praksis**
Afdelingsingeniør
Flemming Gustafsson,
Crone og Koch, Danmark

10.15 **Problemer omkring udnyttelsen af de regionale EDB-centre**
Cand. polit. Ole S. D. Hansen,
Socialforskningsinstituttet,
Danmark

10.30 **Diskussion**

DATAKRAFT + PROGRAMMERING + SYSTEMPLANLÆGNING
Centret accepterer brugerens problemformulering, men påtager sig medansvar for løsningsmetoden.

11.00 Civ.ing. K. Thorup,
Honeywell Bull A/S, Danmark

11.15 **Et integreret datacenter i forbindelse med EDB-undervisning, -forskning og -anvendelser**
Civiling. Bent Frystyk Nielsen,
Aalborg Universitetscenter,
Datacentret, Danmark

B·4 Bygning 302 Auditorium 49

(fortsat)

11.30 **Diskussion**

FREDAG

DATAKRAFT + PROGRAMMERING + SYSTEMPLANLÆGNING + PROBLEMFORMULERING
Centret påtager sig ansvar for formulering, løsning og drift af enkelte opgaver, som dog (i modsætning til Total Facilities Management) behandles hver for sig.

13.30 Afdelingsleder Christian Fischer,
A/S Dansk Regnecenter,
København, Danmark

13.45 Direktør Bjarner Sveigaard,
RECAU, Århus, Danmark

14.00 **Diskussion**

14.30 **Paneldiskussion:**
»Total Facilities Management – Hvor langt kan vi mon gå?«
Ordstyrer:
Direktør, dr. John Gunn,
RECKU, København, Danmark

B·5 Bygning 306 Auditorium 33

DRIFTSPROBLEMER

Emnegruppeleder:
Jan Persson,
Chef för driftsteknikgrupp,
Alfa Laval AB, Sverige

ONSDAG

9.00 **»Beskrivelse av et system för maskinell planlegging og kontroll av EDB-produksjonen«**
Siv. ing. Peter Christian Solberg,
NAMIC A/S, Oslo, Norge

11.00 **»Operationsanalytiske modeller for optimal jobafvikling ved datacentre«**
Civ.ing. Bo Munch-Andersen,
Datalogisk Institut,
Københavns Universitet,
Danmark

FREDAG

9.00 **»Driftsutbildning«**
Utbildningsledare
Göran Steinholtz,
Kooperativa Förbundet,
SAR-Sektionen, Sverige

10.00 **Open house diskussion:**
»Organisation av medelstor datacentral«
Indledere:
Jan Persson,
chef för driftsteknikgrupp,
Alfa Laval AB, Sverige,
Rune Öndemar, driftschef,
Alfa Laval AB, Sverige

B·6 Bygning 208 Auditorium 51

OPERATIONSANALYSE OG MATEMATISKE MODELLER

Emnegruppeleder:
Lektor, lic. techn. Jakob Krarup,
A/S Spadille & Datalogisk Institut,
Københavns Universitet, Danmark

LAGERSTYRING,
INDKØBSDISPONERING OG
PRODUKTIONSPLANLÆGNING

ONSDAG

9.00 **»Kan man integrere lagerstyring og indkøbsdisponering?«**
Systemkonsulent Peter Seerup,
Siemens Aktieselskab,
Data Skandinavien,
Lyngby, Danmark

10.00 **»Model för inköp och lagerhållning av strategiska råvaror«**
Dipl. ing. Ralf William Saxén,
Oy Softplan AB, Finland

11.00 **»Adaptive Control of Spare Parts«**
Fil. lic. Pekka Aho,
University of Jyväskylä,
Jyväskylä, Finland

13.30 **»Simuleringsmodell för bandtillverkning«**
Fil. kand. Laila Strömberg och
bergingenjör
Erik von Wachenfeldt
Uddeholms Aktiebolag,
Databehandling, Forshaga,
Sverige

14.30 **»Long-Range Production Planning System For Open Pit Mining«**
K. Aarnio og P. Niskanen,
Outokumpu Oy, Helsinki,
Finland

15.30 **»Syntese af virksomhedsmodeller«**
Civ. ing. Jørgen A. Richter,
Stærkstrømsafdelingen DtH,
Lyngby, Danmark

TRANSPORT, DISTRIBUTION OG
RUTEPLANLÆGNING

TORSDAG

10.00 **»Kørselsplanlægning for Post- og Telegrafvæsenet«**
Civ. ing. Thorsten Ørnberg,
Crone & Koch, Danmark

11.00 **»Route Planning A Man-Machine Interface problem«**
Civ. ing., lic. techn.
Hans Jørgen Rasmusen,
Nielsen & Rauschenberger,
rådgivende ingeniører A/S,
Danmark

B•6 Bygning 208 Auditorium 51

(fortsat)

OPERATIONSANALYSENS RELEVANS SAMT UDVALGTE EKSEMPLER PÅ ANVENDELSE AF OPERATIONSANALYSE

TORSDAG

13.00 **»FAST« – en problemorienterad stridsmodell«**
Tekn. lic. Hugo Brändström,
AB Teleplan, Sverige

14.30 **»Anvendelse af operationsanalysemodeller i dansk landbrug 1967–73«**
Konsulent Torben Krag Nielsen,
Landbrugets EDB-Center,
Danmark

15.30 **»Den Nya Arbetslösheten – En enkel modell«**
Tekn. lic. Björn T. Cronhjort,
IBM AB, Helsingfors, Finland

FREDAG

9.00 **»Interaktiv beskrivning och lösning av nätorienterade problem med hjälp av computer graphics«**
Lars Lundström,
Tekniska Högskolan i Helsingfors, Avd. f. Databehandling, Finland

10.00 **»Anvendelse af operationsanalyse til dimensionering af vandledningssystemer«**
Stud. scient. Marianne Lilholt og stud. scient Yvonne Gørtz,
Datalogisk Institut,
Københavns Universitet,
Danmark

11.00 **»Optimale korttidsdispositioner på et pengemarked«**
Lektor, civ. ing. Karsten Schmidt,
A/S Spadille & Institut for matematisk statistik,
Københavns Universitet,
Danmark

C•1 Bygning 302 Auditorium 41/42

DOKUMENTATION OG STANDARDS

Emnegruppeleder:
Dipl. ing. Tor-Erik Holmberg,
Systemplaneringschef
A. Ahlström Osakeyhtiö,
Helsingfors, Finland

MÅLSÆTNING:
Framhäva betydelsen av en flexibel system- och programdokumentation.
Kartlägga möjligheterna till en automatiserad planering och dokumentation.
Åstadkomma en paneldiskussion om systemdokumentation och dess underhåll.

ONSDAG

9.00 **»Riktlinjer för administrativ systemutveckling«**
Överingenjör Gunnar Sundblad,
Sveriges Standardiseringskommission, Stockholm, Sverige

10.00 **»A New Approach to Program Documentation«**
Editor John Hurd,
IBM Nordiska Laboratorier,
Sverige

11.00 **»Computer Based Tools for System Documentation: Objectives and Problems«**
Univ. lektor Janis Bubenko,
Projekt CADIS,
Inst. för informationsbehandling – ADB, KTH/SU, Sverige

13.30 **»CS3/4 – Verktyg för systemdokumentation och analys«**
Fil. kand. Stig Berild och fil. kand. Sam Nachmens,
Projekt CADIS,
Inst. för informationsbehandling – ADB, KTH/SU, Sverige

14.30 **»Projektarbeide, standardisering, dokumentasjon«**
Lic. techn. Arne Sölvberg,
Regnesentret NTR, Trondheim,
Norge

15.30 **»Systemdokumentation vid Statens datamaskincentral (VTKK)«**
Byråchef Reijo Koski-Lammi,
Statens datamaskincentral,
Helsingfors, Finland

C•1 Bygning 302 Auditorium 41/42

(fortsat)

TORSDAG

9.00 **»Standardisering, dokumentasjon og undervisning i et bedriftsmiljø«**
Datasjef Helga M. Strømme,
A/S Narvesens Kioskkompani,
Oslo, Norge

10.00 **»Informationsstandard, ett sätt att förhindra administrativt kaos«**
Stig Markstedt,
Alfa Laval AB, Sverige

11.00 **Paneldiskussion: »Systemdokumentation och dess underhåll«**

Ordstyrer:
Univ. lektor Janis Bubenko

Deltagere:
Alle foredragsholdere i emnegruppe C.1.

C·2 Bygning 302 Auditorium 45/46

PROJEKTARBEJDE OG PROGRAMMERING

Emnegruppeleder:
ADB-chef Jussi Tuori,
Kansallis-Osake-Pankki,
Helsingfors, Finland

ONSDAG

9.00 **»Ressource-styret projektplaniægning (RSP)«**
Akademiingeniør
Mogens Sandgaard,
industrikonsulent IKO A/S,
Danmark

10.00 **»Effektiv notatteknikk for prosjektledere og andre EDB-ansvarlige«**
Marketingsjef Odd de Presno,
IKO Software Service A/S,
Oslo, Norge

11.00 **»Prosjektorganisasjonen innen en typisk utviklingsbedrift«**
Systemingeniør Svein F. Strøm,
A/S Norsk Data-Elektronikk,
Norge

13.30 **»Værktøj til systemkonstruktion ud fra beslutningstabeller«**
Civ. ing. Jørn Kofoed Møller,
Den danske Landmandsbank,
Danmark

14.30 **»Interaktiv Sokratisk Algoritme til systemkonstruktion«**
Civ. ing. Søren Skogstad Nielsen,
Databehandlingsafd.,
Direktoratet for Københavns
Amts Sygehusvæsen, Danmark

TORSDAG

9.00 **»Interaktiv programutveckling — erfarenheter och framtida möjligheter«**
Civ. ing. Gösta Steneskog,
IBM Svenska AB, Sverige

10.00 **»Chief programmer team – en väg till effektivare programutveckling«**
Civ. ing. Lars-Gunnar Hultin,
IBM, Nordiska Laboratorier,
Sverige

11.00 **»Programmering – från privat konst till offentlig vetenskap«**
Project programmør
Stig Lindberg,
IBM, Nordiska Laboratorier,
Sverige

13.30 **Paneldiskission om virtuelt lager**
Ordstyrer:
Sven Tafvelin,
Institution för informationsbehandling, Chalmers Tekniska
Högskolan, Göteborg, Sverige
Blandt deltagerne:
Björn Norrbom og
Kaj A. Winberg, Sverige,
Monty Mortensen, Danmark

C·3 Bygning 208 Auditorium 52

PROGRAMMERINGSSPROG

Emnegruppeleder:
Fil. kand. Lars Backström,
Helsingfors Universitet,
Räcknecentralen, Finland

TORSDAG

13.30 **»On the Automatic Property Analysis of Assembler Language Programs«**
Fil. mag. Seppo Laube-Pohto,
Oy Softplan AB, Finland

14.30 **»Beskrivning av ett generellt företagsinriktat rapportspråk med procedurbibliotek och grafisk representation«**
Systemman Lars B. Hedberg,
IBM Svenska AB, Sverige

15.30 **»A Generalised Approach to Syntax Analysis«**
Civ. ing. Karl Soop,
IBM Svenska AB, Sverige

FREDAG

9.00 **»Noen høynivå aspekter ved programmeringssproget MARY«**
Siv. ing. Per Holager,
Norges Tekniske Høgskole,
Trondheim, Norge

10.00 **»APL/1800 - Dets implementering i et procesdatamatmiljø«**
Civ. ing. Jørgen A. Richter,
Stærkstrømsafdelingen – DtH,
Lyngby, Danmark

11.00 **»GRACO-1, et sprog for beskrivelse af geometriske strukturer«**
Lektor, civ. ing. Klaus Illum,
Danmarks Ingeniørakademi,
Bygningsafdelingen,
Aalborg, Danmark

13.30 **»Brugervenlig dokumentation i programmer«**
Civ. ing. Bent Rosenkrands,
IBM A/S, Danmark

14.30 **»Ett system för styrning av ADB-system«**
Civ. ing. Staffan Ranebo,
Saab-Scania AB,
Dataservicesektorn,
Linköping, Sverige

15.30 **»Standardization of Programming Language«**
Professor, dr. phil. Peter Naur,
Datalogisk Institut,
Københavns Universitet,
Danmark

C·4 Bygning 208 Auditorium 53/54

DATABASER, METODER OG ANVENDELSER

Emnegruppeleder:
EDB-sjef Verner Andreassen,
Bergen Kommune, Norge

FREDAG

9.00 **»The Evolution of Data Structures«**
Senior Research Scientist
Charles W. Bachmann,
Honeywell Information Systems,
USA

10.00 **»Den generelle database – praktiske konstruktionsprincipper og -metoder«**
Datamatiker
Mogens Ravn Johansen,
Stærkstrømsafdelingen,
Danmarks tekniske Højskole,
Danmark

11.00 **»Data Structure Programming System Computer Graphics Applications«**
Dipl. ins. Markku Syrjänen,
Helsinki University of
technology, Institute of
Information Processing Science,
Finland

13.30 **»Informationssøgning i en database, repræsenteret ved flows i netværk«**
Systemkonsulent Ellinor Hansen,
IBM A/S, Danmark

14.30 **ZAR II: An integreted storage retrieval and analysis system for survey data**
Diplom Volkwirtschafter
Jürgen Höge:
Zentralarchiv für Empirische Sozialforschung, Universität zu
Köln, Vesttyskland

15.30 **»Databas-Metodik – komponenter av både datorteknisk och icke datorteknisk karaktär«**
Fil. kand. Nils Fredholm,
Skandia, Sverige

C·5 Bygning 208 Auditorium 52

DATANET

Emnegruppeleder:
Civilingeniør Christian F. Rovsing,
Christian Rovsing A/S, Danmark

ONSDAG

9.00 **»Datakommunikation på hemmaplan idag och imorgon«**
Universitetslektor Olle Dopping,
Uppsala Universitet,
Info-behandling, Sverige

10.00 **»Eksperimentelt datanett«**
Cand. real. Dag Haveråen,
A/S Computas, Oslo, Norge

11.00 **»A Blacksmiths View on Routing or Adaptive routing by binary choices«**
Lic. techn., avdelingsingeniør
Tor A. Ommundsen,
A/S Computas, Oslo, Norge

13.30 **Framtida intelligenta maskiner«**
Civ. ek. Ben Wikman,
Saab-Scania AB, Sverige

14.30 **»Datakraftverk – ideal, utopi eller realitet? (Betraktninger på bakgrunn av MULTICS)«**
Utviklingssjef Peter Hidas,
Honeywell Bull A/S, Oslo, Norge

C·6 Bygning 208 Auditorium 53/54

UNDERVISNING

Emnegruppeleder:
Direktør Mogens Boman og
konsulent E. Næsborg,
EDB-Rådet, København, Danmark

ONSDAG

9.00 **»AV-midler i EDB-undervisningen«**
Foredrag med demonstration af bl. a. internt TV

Kursuschef Palle Mogensen,
Scanticon, Århus, Danmark

10.00 Afdelingsleder John Arentoft,
Landbrugets EDB-Center, Århus,
Danmark

11.00 **»Voksenundervisning – om TV-serien: 'På talefod med datamaskinen'«**
Civ. ing. Sven Thygesen,
Kommunedata I/S, Danmark

13.30 **»Omtale af J. D. Warnier's programmeringsmetodik«**
Civ. ing. Paul Møller Nielsen,
Honeywell Bull A/S, Danmark

14.30 **SYSKON ('disputats'-form) »Præsentation af SYSKON-projektet«**
Amanuensis Anders Petersen
('Præses'),
Handelshøjskolen i Århus,
Danmark

14.45 **»Hvad kan SYSKON bruges til?«**
Konsulent Ole Heise
('Opponent'),
Ole Heise Organisation A/S,
København, Danmark

C·6 Bygning 208 Auditorium 53/54

(fortsat)

FREDAG

9.00 **Paneldiskussion:**
Status for EDB-uddannelserne i de nordiske lande

Ordstyrer:
Professor Allan Malmberg,
Danmarks Lærerhøjskole,
Danmark

Indleder:
Professor Chr. Gram,
Danmarks tekniske Høiskole,
Danmark

Deltagere:

Mogens Lyster Knudsen:
Status for EDB-uddannelserne i Danmark

Arun Sarmanto:
Status for EDB-uddannelserne:
Planering af ADB-skolningen i Finland

Staffan Persson:
Status for EDB-uddannelserne:
Sverige

11.00 **»Erfaring med bruk av minidator i postgymnasial, yrkesorienteret EDB-utdanning«**
Undervisningsleder Tor Brattvåg,
Agder distriktshøgskole,
Kristiansand, Norge

13.30 **»Methods to produce ADP-training«,**
Training service manager
Kari Kilpi, Tietotehdas Oy,
Finland

13.55 **»Gennemførelse af brugeruddannelse«**
Konsulent, cand. merc.
Rolf B. Harløf,
Ole Heise Organisation A/S,
Danmark

14.30 **Paneldiskussion:**
»EDB-uddannelse, konfrontation mellem kursusleverandør og kursus 'forbruger'
Indleder:
EDB-chef
Carl Johan von der Recke,
Multi-Data A/S, Danmark
Danmark

D

Bygning 306
Auditorium 33

MINIDATAMATER

Emnegruppeleder:
Lektor Olle Dopping,
Uppsala Universitet, Sverige

ADMINISTRATIVE
TILLÄMPNINGAR

ONSDAG

13.30 **»Användning av minidatorer – en översikt i anslutning till projekt Miniforsk«**
Lektor Olle Dopping,
Uppsala Universitet, Sverige

14.30 **»Key-processing, minidatorns användning för dataregistrering, erfarenheter av CMC-installation«**
Datachef Nils-Göran Svensson,
AB Findus, Sverige

15.30 **»Minicomputers in Administrative Data Processing«**
Bureau Chief Simo Töyrä,
Statens Datamaskincentral,
Helsingfors, Finland

PROGRAMVAROR M. M.

TORSDAG

9.00 **»Operativsystemer for minidatamaskiner«**
Civ. ing. Torbjørn Skramstad,
A/S Computas, Oslo, Norge

10.00 **»Programutvikling for miindatamaskiner«**
Cand. real. Dag Haveråen,
A/S Computas, Oslo, Norge

11.00 **»Nyt datasystem der kombinerer kalkulator, minidatamat og terminal i en kompakt praktisk enhed«**
Salgschef Svend Elvers,
A/S Danbridge, København,
Danmark

11.25 **»Erfaringer ved anvendelse af COM (Computer Output Microfilm)«**
EDB-chef Henning Jensen,
De Danske Redningskorps
Fællesforbund, Hellerup,
Danmark

D

(fortsat)

Bygning 306
Auditorium 33

TEKNISKA TILLÄMPNINGAR
M. M.

13.30 **»Nord-10, en skandinavisk 4. generasjons datamaskin«**
Cand. real. Jan Aske Børresen,
A/S Norsk Data-Elektronikk,
Oslo, Norge

13.55 **»The multi-computer control system for the new CERN synchrotron«**
Siv. ing. Rolf Skår,
A/S Norsk Data-Elektronikk,
Oslo, Norge

14.30 **»Attaching a graphic display to a large computer«**
Direktør, dr. John Gunn,
RECKU (Regionale EDB-Center
ved Københavns Universitet),
Danmark

15.30 **»Programmeringsspråk och programmering av processdatorer«**
Fil. kand. Staffan Kihl,
IBM Svenska AB, Sverige

X

Bygning 302
Auditorium 45/46

OPEN-HOUSE-DISKUSSION

13.30 Har NordDATA en fremtid?

Diskussion om NordDATA-konferencernes formål, relevans, form og indhold.

Alle kan deltage.

Ordstyrer: Forsker Svein A. Øvergaard, Regneanlegget Blindern-Kjeller, Norge.

SOFTWARE HOUSES

Bygning 306
Auditorium 31

Onsdag, den 15 august

09.00 Siemens Aktiebolag, Stockholm
Nils Nilsson
»Data-Bank«

10.00 Systems Programming Limited, London
Carl Nugent, Manager, Small Machines Division, SPL Int.
»Minicomputers — their application, selection and associated problems«

11.00 Cincom Systems International S. A., London
Michael Hunt, Managing Director
»Database Management«

13.30 Informatics (Norden) A/S, Vanløse
Direktør J. Mundus
»MARK IV File Management Sytem«

14.30 Digital Equipment Corp.
1° Vittori Galasi
»Comptex 11 Data Communication System«
2° Roger Hicks
»Interactive Programming and Simulation«

15.30 UNIVAC A/S, København
Bill Littlewood, Systemkonsulent
»Databasesystemet DMS 1100«

Torsdag, den 16. august

09.00 Norsk Hydro A/S
Siv.ing. Magne Klovman
»Præsentation af Norsk Hydro's Order Entry System«

10.00 UNIVAC A/S, København
Björn Dackner, Systemkonsulent
»Produktions- og lagerstyringssystemet UNIS«

11.00 Data Logic Europe
K. Grude (Norge), K. Fr. Martinsen (Norge), J. Hus (Sverige) & A. Thomas (England)
»Vad är Data Logic »Computer Census««

13.30 Siemens Aktiebolag, Stockholm
Ove Stern
»Produktionsstyring«

14.30 Keyboard Training v/ EDB-Centralen, Herning
Salgschef Kirill Forelius & salgschef J. Kjær Nielsen
»Uddannelse og omskoling af tasteoperatører«

15.30 Dansk Dataservice A/S, Ballerup
Marketingschef Jørgen Elle & Konsulent Erik Krogager
»Erfaringer med salg og implementering af standardsystemer i virksomheder med eget anlæg«

Fredag, den 17. august

09.00 aj — konsulenter i databehandling, Holte
1° Akademiingeniør Tage Fredgård, HD
»Module Testing System«
2° Civilingeniør Arne Jacobsen
»STICO, et system til automatisering og rationalisering af programmeringsarbejdet«

10.00 Leasco Software Limited, London
Alan Leibert, Manager, Customers Support
»Reducing Cost of Computing«

11.00 Leasco Software Limited, London
Chris Atkinson, Manager, Systems Programming Division
»Mixed Hardware Systems«

13.30 ADB Utveckling AB, Stockholm
Fil. kand. Lars Bengtsson & Civilingeniør Lårs Hennningsson
»GARBO — Generellt ADB-System för ekonomisk planering och uppföljning«
&
»SPAK — System för projektadministration«

Selskabeligt program

Tivoli-adgang
Kongresemblemet (navnemærket) giver gratis adgang til TIVOLI fra tirsdag den 14. august til fredag den 17. august, begge dage inclusive.

Tirsdag den 14. august mellem 19.00 og 21.00
Uformelt velkomstparty i bygning 101, Danmarks tekniske Højskole.

Onsdag den 15. august kl. 19.00
Modtagelse på Københavns Rådhus, Frederiksberg Rådhus eller Lyngby Stadion efter indbydelse af Københavns Kommunalbestyrelse, Frederiksberg Kommunalbestyrelse og Lyngby-Taarbæk Kommunalbestyrelse. En stående buffet vil blive serveret.
Efter modtagedsen vil bustransport blive arrangeret fra Lyngby Stadion og Frederiksberg Rådhus til Tivoli.
I Tivoli er restaurant NIMB reserveret konferencens deltagere. Der er på forhånd udleveret drinksbilletter. I begrænset omfang kan disse ombyttes til at gælde andre restauranter i Tivoli.

Ledsager program

Tirsdag den 14. august mellem kl. 19.00 og 21.00
Uformelt velkomstparty i bygning 101, Danmarks tekniske Højskole.

Onsdag den 15. august
Formiddag: Ekskursion til en porcelænsfabrik eller en sølvsmedie samt til Rosenborg Slot. Undervejs serveres en forfriskning.
Afgangstidspunkter: se billetten.
Billetter der er bestilt før konferencen er udleveret sammen med konferencepapirerne. Et begrænset antal billetter kan købes i konferencesekretariatet (pris kr. 35,–).
Kl. 19.00
Modtagelser (se ovenfor).

Torsdag den 16. august
Heldagskursion til Nordsjælland med lunch på Store Kro.
Afgangstidspunkter: se billetten.
Billetter der bestilt før konferencen, er udleveret sammen med konferencepapirerne. Et begrænset antal billetter kan købes i konferencesekretariatet (pris kr. 100,–).

Børne program

Et særligt børneprogram er arrangeret for børn mellem 8 og 15 år.
Billetter, der er bestilt før konferencen, er udleveret sammen med konferencepapirerne. Et begrænset antal billetter kan købes i konferencesekretariatet (pris kr. 90,– for hele programmet).

Onsdag den 15. august
Besøg i Tivoli: Børnene mødes ved Tivolis hovedindgang kl. 15.00 og slutter kl. 17.30.

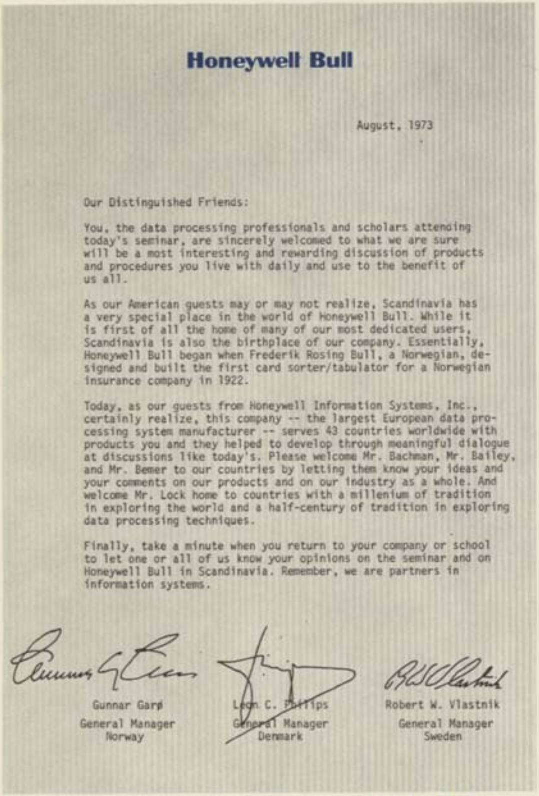
Torsdag den 16. august
Besøg i Cirkus Benneweis: Børnene mødes ved indgangen til Cirkusbygningen kl. 19.30. Forestillingen starter kl. 20.00. Efter nærmere aftale med konferencesekretariatet kan børnene bringes tilbage til hotellet efter forestillingen.

Fredag den 17. august
Besøg i Zoologisk Have: Børnene mødes på Rådhuspladsen ved Palace Hotel kl. 10.00 og bringes tilbage kl. ca. 13.00.

Printed in France - I.F.E. Paris

1973 NORDIC SEMINARS

Honeywell Bull



Honeywell Bull

August, 1973

Our Distinguished Friends:

You, the data processing professionals and scholars attending today's seminar, are sincerely welcomed to what we are sure will be a most interesting and rewarding discussion of products and procedures you live with daily and use to the benefit of us all.

As our American guests may or may not realize, Scandinavia has a very special place in the world of Honeywell Bull. While it is first of all the home of many of our most dedicated users, Scandinavia is also the birthplace of our company. Essentially, Honeywell Bull began when Frederik Rosing Bull, a Norwegian, designed and built the first card sorter/tabulator for a Norwegian insurance company in 1922.

Today, as our guests from Honeywell Information Systems, Inc., certainly realize, this company -- the largest European data processing system manufacturer -- serves 43 countries worldwide with products you and they helped to develop through meaningful dialogue at discussions like today's. Please welcome Mr. Bachman, Mr. Bailey, and Mr. Bemer to our countries by letting them know your ideas and your comments on our products and on our industry as a whole. And welcome Mr. Lock home to countries with a millenium of tradition in exploring the world and a half-century of tradition in exploring data processing techniques.

Finally, take a minute when you return to your company or school to let one or all of us know your opinions on the seminar and on Honeywell Bull in Scandinavia. Remember, we are partners in information systems.

Gunnar Garø	Leon C. Phillips	Robert W. Vlastnik
General Manager Norway	General Manager Denmark	General Manager Sweden

ANDERS GRÖNLUND
08 - 246620

historical highlights

- 1922 - Norwegian engineer Frederick Rosing Bull constructed the first card sorter and tabulator for a Norwegian insurance company.

- 1927 - H.W. Egli Company, Zurich, purchased the Bull patent to market the product in Europe.

- 1931 - The H.W. Egli-Bull company became Compagnie des Machines Bull.

- 1935 - Machines Bull manufactured an alphanumeric printer which operated at 150 lines per minute; a performance rate unequalled during the ensuing 20 years.

- 1952 - After achieving a solid No. 2 position in tabulating equipment, Machines Bull manufactured the first electronic computer employing advanced technology called the Gamma 3.

- 1959 - Ing. C. Olivetti and Company, Milan, Italy, introduced its first solid state Italian-designed business data processing system, the Elea 9003.
  Machines Bull introduced Gamma 60, a powerful large-scale computer system with full simultaneity.

- 1961 - Honeywell, Inc., began marketing computer information systems in the United Kingdom.

- 1962 - Machines Bull introduced Gamma 10, the first punched card computer to bring techniques normally employed on large systems within reach of business of all sizes.

- 1963 - Machines Bull developed CMC-7, a coded magnetic character 7-bar font, adapted as a standard for European banks.
  A Honeywell EDP division was created in the U.K. for the manufacture of H200 and H400 computers.

- 1964 - GE reached financial, technical and commercial agreement with Compagnie des Machines Bull in Paris, and formed Bull-GE to develop, manufacture and market Bull and GE computers in 50 countries.

GE reached a similar agreement with Olivetti in Milan to form Olivetti-GE to develop, manufacture and market computer systems in Italy.

The first computer system was delivered to a customer from Honeywell U.K's Newhouse, Scotland plant.

- 1965 - Bull-GE began manufacturing the G-400 line at its main production facility in Angers, France.

The G-115, designed and developed by Olivetti-GE as the first member of the Series 100 family of small-scale computers was introduced.

- 1966 - Honeywell U.K. expanded Newhouse plant by 60,000 sq. ft. to accommodate additional computer manufacturing demands.

- 1967 - Bull-GE introduced the G-55, a small-scale system with direct access and batch processing capabilities.

GE inaugurated the first time-sharing computer service outside the United States by opening centers in Toronto and London.

- 1968 - GE added 10 more time-sharing centers in Europe and Australia, including: Milan, The Hague, Brussels, Cologne, Copenhagen, Stockholm, Acton and Manchester England, Sydney, and another in Paris.

GE exercised its option and purchased Olivetti's 25 % share holding in O-GE, making the operation a wholly owned subsidiary, and renamed it GEIS Italia.

GEIS Italia introduced the G-130 computer providing step-up capabilities for customers of G-115.

- 1969 - Honeywell received the Queen's Award to Industry for export.

GEIS Italia introduced three more members of the Series 100 line, expanding the family to five.

GE invested $ 20 million to expand time-sharing services in Europe. During the year, worldwide growth had expanded to 75 systems serving 100,000 users in 21 countries on five continents.

Bull-GE introduced a major high performance addition to its Series 50, the G-58.

- 1970 - May 20 - an agreement in principle to merge GE business computer interests and Honeywell computer operations was announced. Merger approved by Honeywell stockholders on Sept. 18 and became official on Oct. 1. This action doubled Honeywell's computer business, and created the second largest force in the computer industry on a worldwide basis.

Honeywell U.K. was awarded the Queen's Award to Industry for export achievement.

- 1971 - Further expansion of the worldwide time-sharing network was announced by HIS Ltd., Honeywell Bull, and HIS Italia.

Honeywell introduced its Series 6000 family of large-scale multidimensional computer systems simultaneously on worldwide markets.

Honeywell U.K., for the third consecutive year, was awarded the Queen's Award to Industry for export achievement.

- 1972 - Honeywell Bull introduced a new data recorder - the K212 - manufactured in Belfort, France.

Honeywell introduced its Series 2000 family of medium-scale computer systems simultaneously on worldwide markets.

- 1973 - Honeywell and Nippon Electric Co. sign licence agreement for peripheral devices.

Honeywell Bull introduced Mark III time-sharing service in France, Sweden, Holland, Belgium, Austria, Germany and Switzerland. Honeywell also markets Mark III in the United Kingdom and Italy.

table of contents

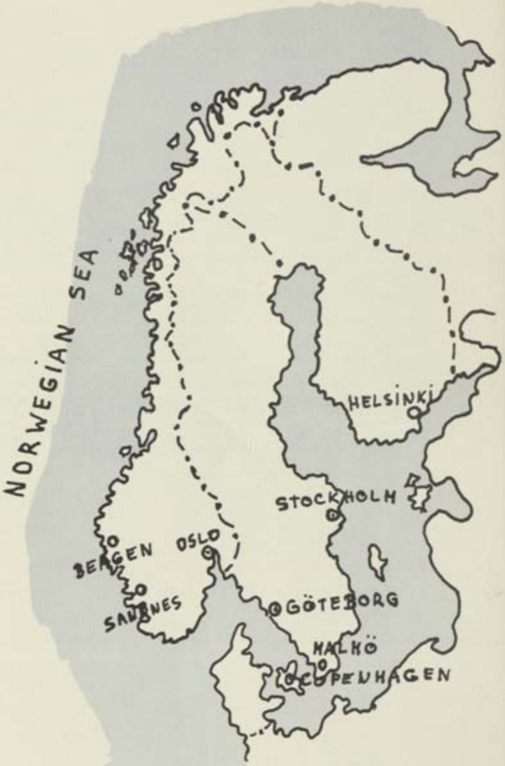
...and services.

- First time-sharing center in Europe inaugurated in 1967 in London; first on the continent, in Paris, the following year.

- Today, services are offered in Europe, parts of Africa, Latin America and Australia — the pioneering Mark I, and the advanced Mark III. All three European-headquartered associates of the Group are time-sharing services leaders in their markets.

- Through an agreement with General Electric Company, U.S.A., the three components will help develop and market a worldwide network linking major cities to a grid of teleprocessing and computer systems via transoceanic link and by satellite.

- Time-sharing service is currently being used by more than 100,000 businessmen, engineers, scientists, analysts and educators worldwide.



Charles W. BACHMAN
Senior Research Scientist
Honeywell Information Systems, Inc.

Charles W. Bachman contributed and continues to contribute to some of the most advanced ideas on one of the computer industry's most far-reaching fields — data base management.

Educated at Michigan State University and the University of Pennsylvania with BS and MS degrees in Mechanical Engineering (1948, 1950), he worked in varying engineering, finance, manufacturing, and data processing assignments for the Dow Chemical Company during the 1950's. Subsequently he held positions with General Electric in manufacturing systems and information systems development from 1960 to 1970, when the General Electric Information Systems business merged with Honeywell's Computer and Communications Group.

Mr. Bachman was responsible for a major portion of the development of the 9PAC, Integrated Data Store and dataBASIC data base management systems. One of the founding members of the CODASYL Data Base Task Group and the inventor of Data Structure Diagrams, he has authored many articles on data base systems concepts.

Most recently, Mr. Bachman was named by the Association for Computing Machinery the recipient of the 1973 A. M. Turing Award — its most prestigious award — for his outstanding contributions to data base technology.

full range of products...

The European-headquartered associates of Honeywell Information Systems manufactures five series' of general purpose and time-sharing computer systems, and markets a full range of computer systems and peripheral devices.

- **Series 50** — Family of three entry-level card- and disk-oriented computer systems for general business and remote batch applications. Family includes Models 53, 55 and 58. Developed and built in France for world markets.

- **Series 100** — Family of small-to-medium-scale computer systems and remote terminals for general-purpose and banking applications. Family includes Models G-105, G-105T, G-115, G-118, G-120, G-130. Developed and produced in Italy for world markets.

- **Series 200/2000** — Family of medium-scale, general-purpose computer systems for batch and communications processing applications. Developed in the U.S., produced in the United States, the United Kingdom, and France. Models include: 2040, 2050, 2060 and 2070 single processors and 2088 dual-processor.

- **Series 600/6000** — Family of large-scale computer systems for business, scientific, and real-time applications in a multi-dimensional environment that includes time sharing, remote and local batch processing, and transaction processing, with capability to access the same large data base. Models include the 6025, 6030, 6040, 6050, 6060, 6070, and 6080. Developed and produced in the U.S. for world markets. Production in the United Kingdom is scheduled to begin in the second quarter of 1973.

- **Series 1640** — Family of four low-cost, high-performance, time-sharing systems using Series 16 minicomputers for processing, communications control, and monitoring functions. Models include 1642, 1644, 1646, and 1648A which provide simultaneous use by 16 to 64 remote terminals.

- **K212 DATA RECORDER** — Fully buffered keypunch-verifier; 6 program format; reads and punches at 80 columns per second; various options including check digit and batch control; developed in France for European market; announced July 1972.

ANGERS France

Angers — Specializes in the manufacture and assembly of Series 50 and Series 2000 computer systems, relays, magnetic drums and memories, printed circuit boards, tape drives; 70,300 square meters.

BELFORT France

Belfort — Specializes in the manufacture and assembly of line printers, punched card equipment, data recorders, other peripherals; 62,700 square meters.

HEPPENHEIM Germany

Heppenheim — Specializes in the manufacture and assembly of disk controllers, disk pack drives; 5,100 square meters.

• With major production facilities at Angers and Belfort, France, and Heppenheim, Germany, Honeywell Bull's manufacturing capability is one of the most powerful in Europe. Each plant is specialized in a particular segment of the product line.

• Research and development operations at Paris, St. Ouen, Belfort and Angers conduct both long-term research in such areas as improved memories and studies of totally new fields of applications, and in short- to medium-term research including advanced systems, basic and applied software development.

Walter O. BAILEY, Jr.
Senior System Engineer
Honeywell Information Systems, Inc.

Walter O. Bailey has seen the computer at work in basic research — including research on itself — as few other men have. Currently a Senior System Engineer in the Central System Design group, he began his professional career with the General Electric Company as a computer applications specialist, working in the areas of flight simulation and trajectory analysis. Subsequently he led or was principal contributor to analysis and prediction technique development in radio frequency interference and information retrieval. In 1967 Mr. Bailey was transferred to Phoenix where he held several management and senior technical positions in advanced system development areas.

Mr. Bailey managed the first implementation of the GPSS Simulation Language of the Honeywell Series 600/6000. His principal technical interests include the definition, organization, and performance prediction of large computing systems.

Mr. Bailey holds a BSE in Electrical Engineering (1959) from the University of Michigan and an MS in System Science (1966) from the Polytechnic Institute of Brooklyn, both highly respected American universities.

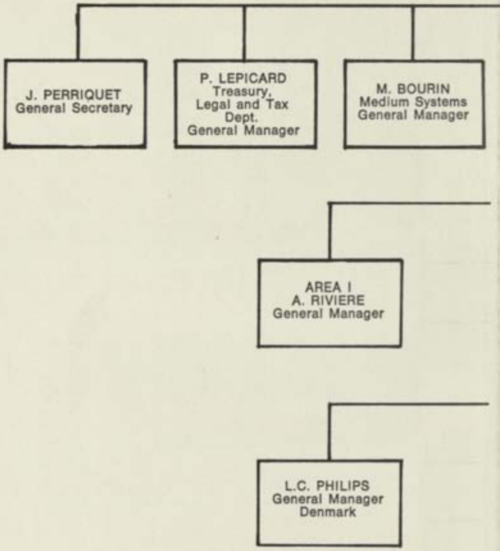
Robert W. BEMER
Staff Consultant
Advanced Systems and Technology
Honeywell Information Systems, Inc.

Robert W. Bemer's experience as a computer professional is as excellent in variety as it is in depth. After his start at the Rand Corporation in 1949, Mr. Bemer spent two years at the Lockheed Aircraft Co. and went on to organize the computing departments at both Marquardt Aircraft and the Lockheed Missiles and Space Company. In 1955 he joined the IBM Corporation as Assistant Manager of Programming Research, later becoming Manager of Programming Systems and then Director of Programming Standards. During the 1960's, he worked for Univac as Director of Systems Programming and Bull General Electric in Paris as Consultant to the General Manager. In 1966, he became a Consultant in Phoenix.

Mr. Bemer has been involved in international and national standardization of computer languages, vocabulary, and character sets since 1960. Currently he chairs the International Standards Organization Subcommittee of Programming Languages.

He was a primary developer of ASCII, and has authored some 50 papers, including many concerning time-sharing. He holds a degree in mathematics from Albion College in the United States. He is a Fellow of the British Computer Society and a member of the Association for Computing Machinery and the Data Processing Management Association.

Honeywell Bull



J. PERRIQUET
General Secretary

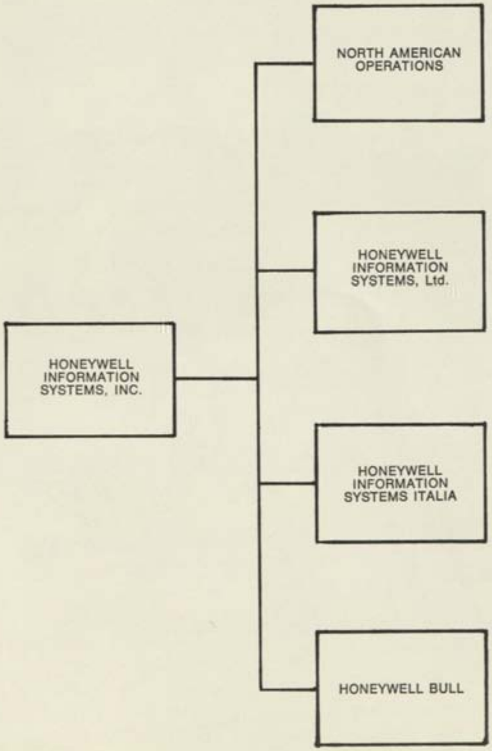
P. LEPICARD
Treasury,
Legal and Tax
Dept.
General Manager

M. BOURIN
Medium Systems
General Manager

AREA I
A. RIVIERE
General Manager

L.C. PHILIPS
General Manager
Denmark

H.I.S. components

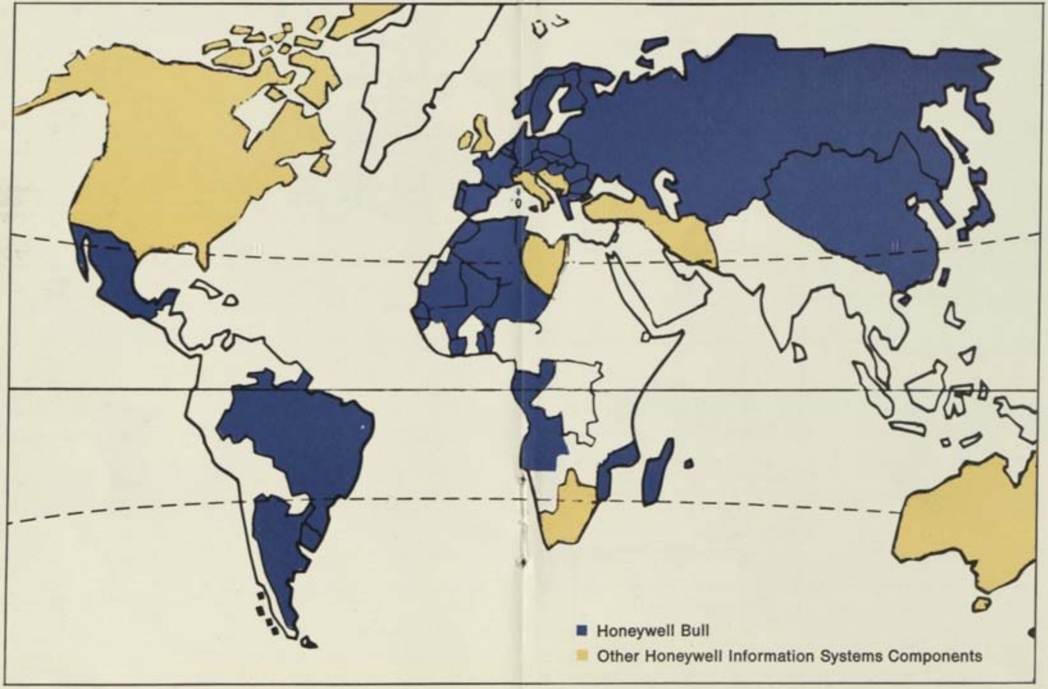

HONEYWELL INFORMATION SYSTEMS, INC.
NORTH AMERICAN OPERATIONS
HONEYWELL INFORMATION SYSTEMS, Ltd.
HONEYWELL INFORMATION SYSTEMS ITALIA
HONEYWELL BULL

Anders G. LOCK
Manager, Software Product Test Unit
Honeywell Information Systems, Inc.

For Anders G. Lock, this visit is a homecoming. Currently responsible for testing Honeywell software — that part of the computer whole that Honeywell considers most visible and therefore most important to the user — Mr. Lock joined the company in Stockholm in 1964. His main responsibilities were in Marketing and Customer Support for both Series 400 and 600 Information Systems. In 1970, Mr. Lock was transferred to Phoenix where he became a member of the FORTRAN Y project. Since May 1971, he has been responsible for various software test functions and is presently Manager, Software Product Test Unit.

Anders G. Lock is a Swedish citizen and holds a Fil. Kand, (B.S.) in Mathematics, Applied Mathematics and Mathematical Statistics from the University of Stockholm. Before joining the then Bull-GE, he was employed by the Swedish State Power Administration as programmer and systems analyst.

Honeywell Bull ...around the world



Honeywell Bull INVITATION

Honeywell Bull A/S inviterer til seminaret

TRENDS IN DATA PROCESSING

****

Tidspunkt: Mandag d. 13. august 1973, kl. 14 - 18

Sted: Kongrescentret KOLLEKOLLE
Frederiksborgvej
3500 Værløse
Tlf. (01) 98 42 22

Seminaret præsenteres af amerikanerne Robert W. Bemer, Charles W. Bachman og Walter O. Bailey, der i august besøger Skandinavien. ?

Disse tre amerikanere er godt kendt i USA for deres banebrydende arbejder indenfor databehandlingsteknikken. Bachman f. eks. er netop blevet tildelt en betydelig amerikansk prisbelønning for sin fremragende indsats indenfor databaseteknologien. Også Bemer og Bailey er pionerer indenfor hver sine områder. Vore tre amerikanske foredragsholdere er nærmere omtalt i DATASYSTEM NYT Nr. 2/73.

I forbindelse med de 3B'ers besøg i Danmark arrangerer Honeywell Bull et seminar om avancerede databehandlingsemner:

W. O. Bailey:	Effects of System Architectures on Performance
R. W. Bemer:	Standards and Compatibility
C. W. Bachman:	The Programmer as Navigator and Communicator

Mere om seminarets indhold og det praktiske arrangement finder De i denne folder.

SEMINARETS INDHOLD

Seminaret åbnes kl. 14.00 af adm. dir. L. C. Philips, Honeywell Bull A/S, Danmark, som vil præsentere vore amerikanske foredragsholdere. Seminarets faglige indhold former sig som tre foredrag med påfølgende diskussion af spørgsmål fra deltagerne.

Walter O. Bailey's foredrag "Effects of System Architectures on Performance" drejer sig om, hvorledes en datamaskines ydelse afhænger af dens arkitektur, og behandler følgende emner:

> The relationship between user needs and system capabilities.
>
> Advantages and disadvantages of performance analysis (simulation models; analytical models; benchmarking etc.).
>
> Performance analysis experiences on Honeywell Series 6000 (methods; results; tools available etc.).

Robert W. Bemer taler om det klassiske kompatibilitetsproblem og dets moderne løsning. Hans foredrag "Standards and Compatibility" omfatter følgende emner:

> Which international standardization efforts are going on, and what are the impacts to be expected?
>
> The impact of database languages, of separating procedure and data description entirely, of labels and of self-descriptive data as seen from the compatibility point of view.

Charles W. Bachman opfordrer i sit foredrag "The Programmer as Navigator and Communicator" til delvis nytænkning og har sendt os følgende resume af sit foredrag:

> Copernicus quietly set our view of astronomical phenomena onto a new road, when he suggested that the earth revolved about the sun instead of the opposite. There is a gathering feeling that data processing personnel would be greatly assisted if they were also to accept a new point of view for their thinking. This new viewpoint would take the application programmer out of the center of core storage and set him moving in two new directions. He would learn to act as a navigator within the database and learn the rules of the road to avoid conflict with other programmers as they jointly navigate the database space. He would also learn to cooperate with the programmers of other work stations with whom he has divided a large business problem to create smaller, more manageable pieces and communicate with them through the message system.
>
> The achievement of this reorientation promises to cause those in our field as much anguish as did the earlier one started by Copernicus; as sacred ideas are challenged, overturned and better ones erected in their place.

Seminaret forventes at slutte ca. kl. 18 med en let servering, hvorunder yderligere diskussion med de tre foredragsholdere kan finde sted.

SOFT-COPY CONTROLS

Thomas O. Holtey
Advanced Computer Design, Billerica, MA
and
Eric H. Clamons
Advanced Systems and Technology, Phoenix, AZ

WHAT ARE SOFT-COPY CONTROLS?

The work of the early 1960's which produced the ISO 7-bit Coded Character Set (ASCII in the US)[1] was aimed at solving the problems of marriage between computers and telegraphic typewriters. In recent years other types of interactive devices, especially cathode ray tube displays, have become more economical to use. The control functions needed to use these devices, with or without being connected to a computer, are known as soft-copy controls.

The National Bureau of Standards (US) first recognized the need to standardize soft-copy controls. At a Soft-Copy Workshop, 1970 October 6 and 7, a group of control functions were isolated and assigned to the repertoire of an 8-bit expanded ASCII (See Honeywell Computer Journal, Vol. 5, No. 3 and Vol. 6, No. 4). The effort rested until an international effort spearheaded by the European Computer Manufacturers Association began a serious study of the requirements for control functions and began to classify them. Similar efforts were authorized in the US's code committee X3L2, chartered under American National Standards Institute (ANSI) rules by its secretariat the Computers and Business Equipment Manufacturers Association. Their effort was not mounted seriously until early European success convinced them of the need and justification for soft-copy controls. It was, and still is, the intent of the two developers to come up with a common solution. However, this report will show them at odds. Their aims differ; ECMA's would produce a core set of controls to which functions could be added later, the US's would produce a comprehensive set from which subsets could be chosen. The stage is set for compromises. The Honeywell Computer Journal provides the scenario for the plays. The libretto is supplied as a supplement on the microfiche of this issue (inside back cover); it contains summaries of the documents needed to study the proposals more seriously.

WHERE TO PUT SOFT-COPY CONTROLS

ASCII as it now stands cannot accommodate more control characters. Fortunately, a code extension standard was developed [2] which permits either the expansion of the set to an 8-bit code in which columns 8 and 9 of the new array are reserved for 32 additional controls, or the ESCape code of the 7-bit code can be combined with one of the graphic characters of columns 4 and 5 to effect a code extension to represent 32 new control characters. The two methods are related by a doctrine. Summaries [3,4,5] are appended to the microfiche of this issue, as noted. Their scope is too broad to be discussed here. These methods do provide, in addition to other characters, 32 more control characters to be applied to the control of soft-copy matter.

CONTROL FUNCTIONS NEEDED

Tabulation Controls. One of the most serious omissions in the ISO Code was that the "tabs" could not be set or cleared. Both horizontal and vertical tab stops can now be manipulated by code:

HTS	Horizontal Tabulation Set
HTC	Horizontal Tabulation Clear
VTS	Vertical Tabulation Set
VTC	Vertical Tabulation Clear

Format Effectors. A number of shortcomings of the ISO Code are corrected. Provisions for moving lines forward or back by half a space at a time, backspacing by half a character, and for differentiating between fixed spaces and variable spaces (for applications involving flush right margin) are made:

FHL	Forward Half Line feed
RHL	Reverse Half Line feed
RLF	Reverse Line Feed
HSB	Horizontal Space Backward
FXS	FiXed Space (in contrast to SP in ASCII)
NL	New Line (CR & LF of ASCII in one code)

Highlighting Controls. Many devices provide more flexibility than typewriters to vary intensity, shape, color, etc. of characters. Underlining was singled out as more common than others. Two methods are provided; continuous underlining and underlining all characters except spaces; both are terminated by the same character:

BHU	Begin Horizontal Underscore
BWU	Begin Word Underscore
EHU	End Horizontal Underscore (for BHU and BWU)

The other highlights are classified as "alternate" graphics. The alternate form of the graphic is declared by a selection of a graphic mode character sequence (see SGM):

PGR	Primary Graphic Rendition
AGR	Alternate Graphic Rendition

Privileged Areas. Soft-copy devices are sometimes used to fill in a form. The printed areas are fixed or "protected" from alteration:

SPA	Start Protected Area
EPA	End Protected Area

Blanks are provided on the form to be filled in; they are "selected" for insertion:

SSA	Start Selected Area
ESA	End Selected Area

Command controls. In many applications it is found convenient to intersperse text with commands e.g., "indent" the next paragraph. These commands are delimited by characters which set them off from text. Two opening command delimiters permit two levels of commands, one for general use, the other for addressing the operating system. A common closing command delimiter is provided:

CD	Command Delimiter (opening)
SYU	SYstem Use (opening)
TD	Terminating Delimiter

Miscellaneous functions. One of the characters needed is one which prints the display contents or records them on a medium:

MC	Media Copy

Because much redundant data may be on the display, e.g., the fixed portion of a form, a control is provided which permits transmission of only the characters preceding it in a line:

LEL	Logical End of Line

Private use. Four control characters are set aside for the user to use as he sees fit:

PU1	Private Use 1
PU2	Private Use 2
PU3	Private Use 3
PU4	Private Use 4

Control extenders. These functions are similar to those of ESCape and Data Link Escape of the ISO Code. Four are provided; two act as modifiers for the single graphic characters which follows them. Together, the two characters and the associated graphic take on a new meaning. One extender expands the coded graphic repertoire of a device beyond that of the ISO Code:

SGS	Single Graphic Shift

The other extender is used to encode normally local controls for systems which prefer to have the local controls performed remotely:

EDT	EDiT function

The other two extenders are characters which begin a string of characters indicating a dimension or pointer associated with a function. The string is terminated by a character which also identifies the function which uses the parameter(s) it helps bracket. They are of the form:

Control character	parameter	function character

PCD	Parametric Control (Dimensional)
PCP	Parametric Control (Pointer)

The specific format is not yet defined. It could be of fixed format, i.e., all character sequences of equal length and containing only one parameter, or free format, i.e., one or more parameters of unequal length separated and terminated by defined delimiters. It is likely that the formatted parametrized control sequences will be defined as fixed because of the hardware orientation of the functions they invoke.

Character Sequences Defined By Control Extenders

Single character extensions. No specific assignment has been made; these examples only illustrate the method:

SGS A = α
SGS B = β
SGS % = $\int$ etc.

Local editing controls. The functions for these controls have been defined. The specific assignments shown here are only for purposes of illustration. They are initiated by an EDT:

Clearing functions:

CAS	Clear All Selected (areas)
CPS	Clear all Protected and Selected (areas)
CLB	CLear Buffer
CLS	CLear Screen

Cursor functions:

CUU	CUrsor Upward (one line)
CUD	CUrsor Downward (one line)
CUL	CUrsor Left (one character)
CUR	CUrsor Right (one character)
CUH	CUrsor Home (upper left corner)
CNL	Cursor New Line (left side of next line)

Delete functions:

DCD	Delete Character in Display (and close the gap, moving all characters to the end of the display)
DCL	Delete (active) Character in Line (and close this gap in the line)

Erase functions:
(the gap created remains in text)

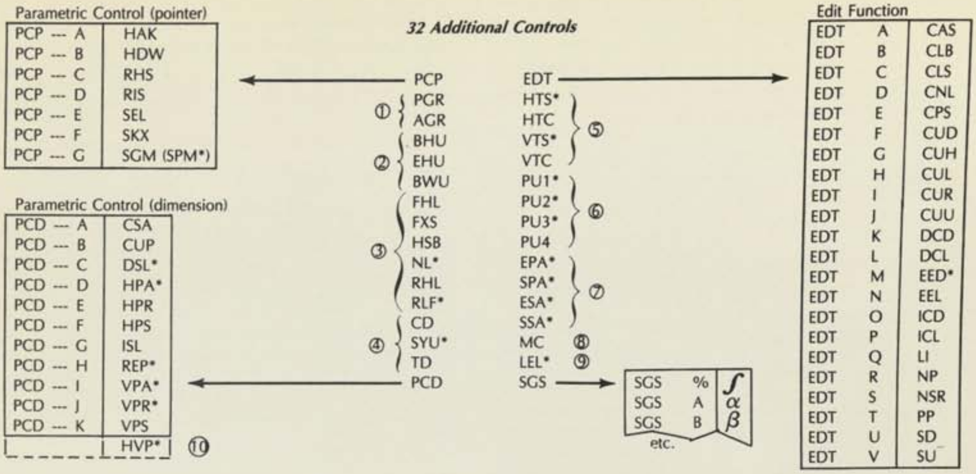
EED	Erase (from active position) to End of Display (all unprotected data)
EEL	Erase (from active position) to End of Line

Insertion functions:

ICD	Insert Character in Display (see DCD)
ICL	Insert Character in Line (see DCL)
LI	Line Insert (active line and all others move down, leaving blank line)

Page functions:

NP	Next Page (to display)
PP	Previous Page (to display)
SD	Scroll Down (display)
SU	Scroll Up (display)
NSR	NonSelective Read (copy entire buffer)



Parametric Control (pointer)

PCP --- A	HAK
PCP --- B	HDW
PCP --- C	RHS
PCP --- D	RIS
PCP --- E	SEL
PCP --- F	SKX
PCP --- G	SGM (SPM*)

Parametric Control (dimension)

PCD --- A	CSA
PCD --- B	CUP
PCD --- C	DSL*
PCD --- D	HPA*
PCD --- E	HPR
PCD --- F	HPS
PCD --- G	ISL
PCD --- H	REP*
PCD --- I	VPA*
PCD --- J	VPR*
PCD --- K	VPS
	HVP* ⑩

Edit Function

EDT	A	CAS
EDT	B	CLB
EDT	C	CLS
EDT	D	CNL
EDT	E	CPS
EDT	F	CUD
EDT	G	CUH
EDT	H	CUL
EDT	I	CUR
EDT	J	CUU
EDT	K	DCD
EDT	L	DCL
EDT	M	EED*
EDT	N	EEL
EDT	O	ICD
EDT	P	ICL
EDT	Q	LI
EDT	R	NP
EDT	S	NSR
EDT	T	PP
EDT	U	SD
EDT	V	SU

① Alternate Graphic Forms (Highlighting)
② Underlining "
③ Format Effectors
④ Command Controls
⑤ Tabulation Controls
⑥ Private Use
⑦ Privileged Areas
⑧ Media Copy
⑨ Logical End of Line
⑩ Horizontal and Vertical Position (ECMA only)

* Controls found in the ECMA proposal

Pointer type controls. These controls are mostly hardware instructions; they are initiated by a PCP:

RHS	Request Hardware Status (from remote device)
HAK	Hardware AcKnowledge (by remote device)
HDW	HarDWare malfunction (by remote device)
RIS	Reset (remote device) to Initial State.
SEL	SElect device (by parameter)
SKX	SKip to channel X (advance medium by amount indicated by parameter)
SGM	Select Graphic Mode (e.g., blink, alternate color, italics, bold, increase intensity, change font)

Dimension type controls. Almost invariably these controls are associated with one (x or y) coordinate; they are initiated by a PCD. There are three functional groups:

Positioning controls move the active position:

to the position indicated by the parameters:

HPA	Horizontal Position Absolute
VPA	Vertical Position Absolute
HVP	Horizontal and Vertical Position (ECMA only)

from the active position forward by parameter:

HPR	Horizontal Position Relative
VPR	Vertical Position Relative

from the active position to the next tab stop:

HPS	Horizontal Position Select
VPS	Vertical Position Select

Remote Editing Controls:

CSA	Clear Selected (Area) to address indicated by parameter)
CUP	CUrsor Position (to position indicated by parameter)
DSL	Delete Specified Line (specified by parameter)
ISL	Insert Specified Line (specified by parameter)

Repeat:

REP	Repeat (number of times specified by parameter)

The material presented here represents an effort which when completed will impact industry as much as ASCII did a decade ago. There is still time to take an interest in the outcome. If interested, contact:

R. M. Brown
Director of Standards
CBEMA
1828 L Street, NW
Washington, DC 20036

or

D. Hekimi
Secretary General
ECMA
Rue du Rhone 114
1204 Geneve, SWITZERLAND

REFERENCES

1. ISO Standard 646, ISO Code (see ANSI X3.4).
2. ISO Standard 2022, Code Extension.
3. ISO TC97/SC2/(ECMA-50)639, Additional Control Characters.
4. ISO TC97/SC2/677, Additional Control Characters (ANSI X3L2/1383).
5. ISO TC97/SC2/N646, Additional Control Characters.

authors

PAUL A. ANDERSON

Systems and Research Center, Minneapolis, MN, US

Mr. Anderson is a Senior Systems Analyst, currently developing digital simulations and a variety of computer analysis tools for the study of command and control problems in urban mass transit systems. Previously he had technical responsibility for conducting simulation studies to help define the best transit alternatives for the Minneapolis - St. Paul area; this investigation studied buses both on exclusive guideways and in mixed traffic, with consideration for operating strategies and traffic signal timing patterns. Mr. Anderson has had a wide variety of simulation experience in connection with space vehicle and aircraft display/control problems, and has published other papers on simulation and transportation systems. He holds a BSAE (1964) and an MS in Engineering (1966) from the University of Minnesota (US), and has done additional graduate work in control theory and economics. He is a member of AIAA and the Highway Research Board, and is an instructor in new transportation concepts in the Honeywell Continuing Education Program.

VAUGHN M. AUTREY

Ernst & Ernst, Phoenix, AZ, US

Mr. Autrey is a Consultant with the public accounting firm of Ernst & Ernst. He holds a BS in Mathematics and Physics, and has done postgraduate work in business administration. 12 years of experience in data processing system analysis and design, hardware and software, has included work on communication network and concentrator design for computer utility operations, applications design for student information systems, and directing development of several software systems. Recent experience includes a number of EDP audits. Until recently, Mr. Autrey was responsible for security research in the area of network processing and personal identification systems for Honeywell's Data Security Project, Advanced Systems Technology Operation, in Phoenix.

ROBERT W. BEMER

(See Vol. 6, No. 4)

JEFFREY P. BUZEN

HIS Technical Office, Waltham, MA, US

Dr. Buzen is a member of the HIS Technical Office. His primary responsibilities are in the areas of data management, performance evaluation and system architecture. He is also a member of the Computer Science faculty at Harvard University, where he teaches advanced courses in operating system design, stochastic analysis methods, and queueing theory. His previous computer experience, dating from 1961, includes work at the National Institutes of Health and research and teaching positions at the Brown University Computer Center and the Aiken Computation Laboratory. He holds an ScB from Brown Univ. (US) and an MS and PhD from Harvard Univ. (US), all in Applied Mathematics. He is a member of the Association for Computing Machinery.

ERIC H. CLAMONS

Advanced Systems and Technology, Phoenix, AZ, US

Mr. Clamons is Consultant - Privacy and Security, to the Director of ASTO in Phoenix. He has been associated with computers since 1948; in the Department of Applied Mathematics and Mechanical Engineering at the University of Minnesota (US), and subsequently at the Honeywell Aero Division and at Univac in Product Planning and Data Systems Standards. He joined General Electric (the division now part of Honeywell Information Systems) in 1969, becoming Director of Data Systems Standards for Technical Resources Planning in Waltham, MA. He received a BA in Mathematics from Macalester College (US) and an MS in Applied Mathematics from the University of Minnesota (US).He has long been active in international standardization work.

UGO O. GAGLIARDI

HIS Technical Office, Waltham, MA, US

Dr. Gagliardi is Director of the HIS Technical Office, responsible for HIS-wide product specifications. He is also lecturer on Computer Sciences at Harvard University in the areas of operating systems design and computer architecture. He has been associated with computers since 1954; in the E.E. Department of Kansas State University, subsequently in the Systems Division of Dunlap & Assoc., Darien, CT and at the USAF Labs at ESD, Hanscom Field, MA. He joined Honeywell in 1970 and Harvard in 1966. He has been involved in a number of projects to develop and implement state-of-the-art computer graphics and timesharing techniques. He received a Dr. Eng. Degree from the Polytechnic Institute of Naples University (Italy) and did postgraduate work at Columbia University, NY (US).

INTEGRATING COMPUTER TEXT PROCESSING WITH PHOTOCOMPOSITION

ROBERT W. BEMER and A. RICHARD SHRIVER

Abstract - Using a computer text processing system as the entry and change vehicle for a photocomposition system affects the publishing function in many ways. Costs are reduced, quality and readability are enhanced, esthetics are more controllable, and entry personnel require little training. Proofreading is almost entirely replaced by a computer-generated concordance. Mechanicals for reproduction are completed at the editor's site, not at the printer's, completing one more step in the movement to the automated office.

The *Honeywell Computer Journal* is published concurrently on hard copy, microfiche, and magnetic tape. The tape can be used to drive other photocomposition systems that differ from our own, just as a computer can translate COBOL programs to the running instructions of a particular computer. Thus our work has shown the way to a common composition language that can describe all formats and identify uniquely the universe of printed symbols.

INTRODUCTION

The introduction of computers to the composition process began in 1961, but not much thinking was applied to the system aspects. This led to some failures and marginal returns. Hyphenation and justification, the earliest uses, are actually trivial. So are text entry and control of character generation.

The challenging functions are page layout, pagination, tabulation, indexing, ruling, proofing, and multiple output from a single file by changing the variables.* However, a danger lies in trying to do these functions automatically by the computer; the amount of difficult programming required often leads to excessive costs, disillusion, and project abandonment - with concomitant prejudice against computers. Shatzkin [1] said:

> "The key contribution that the computer can offer the book publisher is very simply this: the predictability of the final result! This may sound very anticlimactic, an absurdly small benefit from such a mighty instrument, but I assure you that predictability can change procedures and even the nature of book publishing in very revolutionary ways."

Our experience in publishing the *Honeywell Computer Journal* has borne this out. As usual, close cooperation between human and computer pays off best. The basic ingredient of our system design is the cost of photocomposition relative to hand or linotype setting, being cheaper by a factor of more than 20. This leads to the philosophy that we shall always make many photocomposition runs, which governs our procedures from the outset.

SCHEMATIC OF THE SYSTEM

The HIS 6000 system is used for text entry, editing, storage, and running concordances. It is not normally used for the "run-off" function (producing formatted copy on the entry terminal). Even though this feature is available, it is tedious, expensive in line cost, and has little value for final copy.

Formatted copy is produced only by photocomposition. When this is desired, a special postprocessor program converts the text stream and embeds macros for the Page 2 System. This produces a magnetic tape which is (now) transported physically to the facilities of Datagraphics, in Phoenix, AZ, and input to a Univac (nee RCA) 2 driving a III Videocomp 830. The resulting copy is laid up in desired page form, and a cycle of editing and further photocomposition begins.

Final copy is waxed on templates in the traditional manner. Special heads are added (in fonts not available to the computer system, and chosen to symbolize article content, where possible), and it's off to the printers.

Basically, we have adjoined two free-standing systems, and in so doing removed from the middle the expensive and non-graphic-quality output of the first, and the somewhat tedious and inflexible input of the second. Jury-rigged as it is, it is nevertheless superior to any method formerly available to us, and points the way to integrated systems for the future. We can live for now with our 2-hour turnaround.

TEXT ENTRY

Text entry is accomplished in the timesharing mode with the standard HIS 6000 Text Editor System [2], an embedded format system based upon M.I.T. work and similar to the IBM Script. It is *not* a numbered line system like ATS, and eight years of experience has proved *this* wisdom. Searching and alteration are done primarily in the string mode. "Cut and Paste" is limited to operate by the number of lines moved, but they are not themselves numbered.

All control actions are signified by embedded "dot" commands. This input convention states that a CR (Carriage Return) character followed by a full stop character (period, dot) signifies a control statement, thus:

```
.begin      .center   .indent n   .subpara n   .TAB
.space (n)  .adjust   .undent n   .para        .break
```

These are but a subset of the standard Text Editor and can be learned by an unskilled person in an hour or so. The editing commands will be explained by examples in the running text of this paper.

Manuscript received 1973 June 8. This paper was presented at the 1973 IEEE Conference on the Future of Scientific and Technical Journals, New York, N.Y., May 17-19.

The authors are with Honeywell Information Systems, Inc., Phoenix, AZ 85005.

* For example, this article was photocomposed (by the methods described) in this form and in an alternate form for the 1973 National Computer Conference; the common portions are used with permission of AFIPS.

ENTRY FOR UNSKILLED PERSONNEL

The postprocessor program that converts for the Page 2 System is vital for simple text entry. The standard entry methods for the Page 2 System are certainly *not* simple and require some training and a crib sheet constantly on display to the enterer. Remember that graphic quality output requires a separate font generation for each unique character. It is not sufficient to overprint an umlaut (in its fixed position) for both the upper and lower case "u", for example. The postprocessor does extensive string analysis, much of it based upon backspace and overstrike for entry, which makes it simple for personnel. Examples:

- Characters with diacritical marks (accent acute, accent grave, tilde, umlaut, etc.) are produced by backspace on the terminal and overstrike with the proper character (double quote is used for umlaut).
- Double and single quotes are used as they are for entry. The postprocessor determines whether they are opening or closing quotes. A double quote is two single quotes in photocomposition, and this is called automatically.
- For minor occurrences in text, boldface may be indicated by overstriking single characters three times. This is visible on the terminal when the line is verified. For a longer string of bold characters, the font is altered by a **.bold** command and turned off by a **.bold end** command. These commands do not force a new line.
- For minor occurrences in text, italics may be indicated by backspacing the length of the word and underlining. This is visible on the terminal when the line is verified. For a longer string of italics, the font is altered by a **.ital** command and turned off by a **.ital end** command. These commands do not force a new line.
- The bulleting seen here is accomplished by a **.indent 3** followed by a **.undent 3** (which is operative only for the next line), a lower case "oh", 2 blanks, and then the text. The uniqueness of this string permits the convention.
- To the regular Text Editor convention of using the "at" symbol to delete the previous character (guess why our articles never contain this character!), and CAN to delete the entire line of entry, we have added the caret to indicate the en space, which is incompressible to the justification process. Thus a new paragraph is caused by a **.break** and an initial line with two carets for indention.
- The normal font sizes for *Honeywell Computer Journal* are:

  9 point - text
  8 point - references, some displays as necessary
  7 point - sub- and superscripts, figure captions

Point size may be changed at any point in the text by inserting the ESCape sequence:

ESC g (7-pt), ESC h (8-pt), ESC i (9-pt)

These override the original settings, and are used for formulas, etc.

CONTROL OF PAGE LAYOUT

It has been a remarkable discovery to us that reader attraction and satisfaction is increased significantly by tight control of page layout. Only in the most exceptional cases will a column start in the middle of a sentence, and then only on the second column of the same page. Usually a column will start with at least a paragraph (not just an arbitrary paragraph, but one that makes sense), and very often with a heading. The appearance of a figure or table will never precede its first mention in text, nor will it often be on a page that is not visible when that mention is made. "Widows" never occur.

Under traditional methods, the editor loses control of page layout after the galley stage; all of the niceties must be left to a composer who has little understanding of the subject matter, and is often less interested in reader satisfaction. With the low cost of text processing taken in conjunction with photocomposition, we do not mind expending many runs to get just what we want.

A quick reading of the first galley copy gives an estimate of the author's redundancy or flowery speech factor and other ways that compression can be achieved if necessary. Accordingly, the actual film is cut to lay out an approximation of the article. As the last page is always full, we work backward. Whatever is left for the first page we leave for artistic treatment and the "From the Editor" commentary. Great attention is paid to aspects of future readability, left or right page assignment, pleasing placement of tables, figures, and photos. Virtually no attention is paid to typos and other mistakes that exist in the copy. Accordingly, the single columns are taped on with more lines than our standard, trusting to judicious editing to cut back to the right number (60).

The beauty of this system is that many things can be changed simultaneously to create correctness, harmony, and interest: point size for certain paragraphs or tables, tab settings, subparagraphing, font style, and text changes and corrections. Imagine a situation where the column copy has to be reduced by two lines, and yet previous editing has taken advantage of all short lines at the end of paragraphs, filler words have been removed, and big words replaced by commoner smaller words with equivalent or clearer meaning. Now you have to get into the guts of the author's meaning and say it shorter and clearer, without altering the flavor or meaning in any way! Being forced to do this by our aesthetic standards for page layout yields a big dividend in increased readability.

Depending upon the content, we may photocompose the text from 2 to 5 times. Do the authors complain about the alterations? Never, in our experience. When it reads well, they just assume that they wrote it that way, never checking their original copy. We have also experimented in putting the author's work in to typeset even when it is only rough draft; results seem to indicate that the visualization of final copy permits him to improve it more that he could by editing from a typed draft.

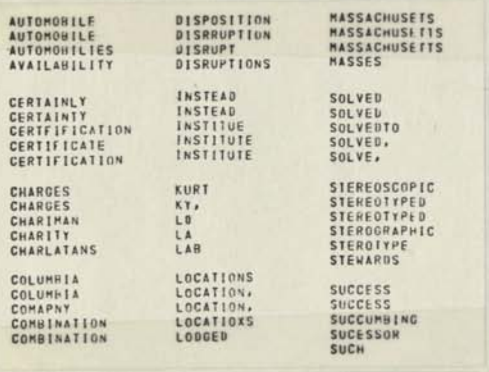
Obviously, taking this much work for readability means high acceptance standards, and we insist that this is a good thing. Dung coated with 53 layers of Chinese lacquer is still dung, and we do not intend contributing to information pollution.

Hopefully, it is now clear why we do not use the computer for automatic pagination.

PROOFING OF COPY

An optional feature, or byproduct, is the concordance run, usually exercised on what is expected to be the next-to-last photocomposition run. This produces two listings on the high-speed (upper case only) printer. The first listing is a Key Word Out of Context (KWOC) listing; each numeral and word (except for the very small common ones) is listed on the left in collating sequence order, with its entire entry line on the right. The lines *are* numbered here, for cross-reference to the second listing, which is the consecutive text.

The concordance is now scanned visually, primarily to detect input errors ("typos"). See Figure 1 for some examples. It is our experience that these fairly jump out at one in scanning a concordance, whereas they remain stubbornly glossed over by the eye and mind in traditional proofreading. However, we do read the text - for style and making sense, not for typos. In fact, knowing that you are freed from the typo-hunting task creates a different frame of mind for doing *real* editorial work.



```
AUTOMOBILF          DISPOSITION        MASSACHUSETS
AUTOMOBILE          DISRRUPTION        MASSACHUSETIS
AUTOMOHILIES        DISRUPT            MASSACHUSETTS
AVAILABILITY        DISRUPTIONS        MASSES

                    INSTEAD
CERTAINLY           INSTEAD            SOLVED
CERTAINTY           INSTITUE           SOLVED
CERTFIFICATION      INSTITUTE          SOLVEDTO
CERTIFICATE         INSTITUTE          SOLVED.
CERTIFICATION                          SOLVE,

CHARGES             KURT               STEREOSCOPIC
CHARGES             KY,                STEREOTYPED
CHARIMAN            LB                 STEREOTYPED
CHARITY             LA                 STEROGRAPHIC
CHARLATANS          LAB                STEROTYPE
                                       STEWARDS
                    LOCATIONS
COLUMBIA            LOCATION,          SUCCESS
COLUMBIA            LOCATION.          SUCCESS
COMAPNY             LOCATIOXS          SUCCUMBING
COMBINATION         LODGED             SUCESSOR
COMBINATION                            SUCH
```

Figure 1. Typos exposed by concordance.

CONTROL OF READABILITY AND STYLE

The concordance produces a histogram of word size distribution as a byproduct, and the average word length may be calculated. We target 5.0 characters per word, and are very suspicious of readability when the author gets above 5.5.

One aspect of style, or rather one of our rules, is that an acronym shall always be given the spelled-out version in parentheses the *first* time it is encountered in text. One has only to spot the first occurrence in the concordance and look to the corresponding line on the right to see if this has been done. If not, edit.

The *Honeywell Computer Journal* has other style rules. Most important is adherence to ISO Standard 1000, or the International System of Units (SI). Check the concordance for inches, feet, yards, miles, pounds, etc. If they occur, and are for measurement, they had better be in parentheses following a metric value. Other examples are: $2 million - not 2 million dollars; 0.5 s - not .5 sec; focused - not focussed.

ECONOMIC CONSIDERATIONS IN WORKING METHODS

As there is no way to predict the pagination of printed copy when entering text, one could enter it all under a single file name. However, the 6000 Text Editor keeps the entire file in the main store for faster processing (and it is *really* fast), and these facilities must be paid for. Thus original input is made in judiciously separated and named files, breaking at headed sections, for example. These are then adjoined for the photocomposition run.

After page layout is determined, they are adjoined again and resplit by page into files with new names, and the old ones purged. This permits single columns to be reworked into final form. The present rate is $1.75 per column. Thus a page costs from $6 to $10 to compose, comparing rather favorably with the $70 per page we were paying for linotype setting to our standards before our system was operable. The 6000 cost is not included, as we have been unable to get real figures because we work on an inhouse "exposure" system used for checking out new software releases. We do, however, feel that this cost is compensated by the system doing automatically what we would have to do ourselves otherwise (like proofreading), and by the added quality. We do need to modify our programs in order to be able to set double column on the last run.

Economy dictates that we should process as much text as possible on each photocomposition run. This means linking several files and saving them as a single file. But this increases the risk that something going wrong early will spoil the balance. Care must be taken to separate and insulate each file from any other. Convention starts each file with **.begin** (for a new galley), **.indent 0** (in case the file ahead of it lacked a command to restore indention to 0), and **.adju** (in case the preceding file had been using tabulation and was not restored to the justification mode).

The power of the Text Editor is of great assistance in checking for correctness of the adjoined file, particularly for closure. Type:

fs:/.bold/;*   (meaning "find all occurrences of that string")

and you will almost instantly get a message like:

end of file - request executed 122 times

Hit "b" and CR (for backup to the file beginning), and type:

fs:/.bold end/;*

If the message doesn't say 61 times - trouble! A 60 would mean that bold did not get turned off somewhere, and the copy following will be in useless boldface. Do the same for italics, subparagraphs, point size changes, etc.

The files must always be correct for the magnetic tape edition, and identical to the printed copy. Yet it is often wasteful to rerun the entire file for simple patches. A copy is made, and the correct parts wiped out by string replacement, leaving only the changed copy to be reset as a patch (with due consideration to leaving enough text so that paragraphing, etc., is unchanged). These patches are saved under a different name; a number of them are adjoined and run at one time.

INCIDENTAL ADVANTAGES

A number of dividends have shown up that we amateurs did not really foresee:

- Doing our own typesetting permits laying up mechanicals for articles as soon as they are ready, without waiting to group an entire issue for the typesetter to schedule in some time slot. Exclusive of conditions of extreme timeliness, this permits better selection for issue makeup and content.
- Having the feel of the final product, by mockup during the editing and changing stages, affects everyone - author, editor, and reviewer. For the latter, particularly, it gives psychological impetus to hurry up - lest what he dislikes might be in the finished product. All can work simultaneously to correct and improve the copy and make it more readable.
- The Page 2 System hyphenates to English rules and/or custom. Normally we run our French, German, Italian, and Spanish sections in "fill mode" (stretching the spacing between words to fill the line without hyphenation). But if glaring gaps exist we remove them easily by doing a dummy hyphenation, splitting the first word of the next line into two components:

```
rs:/whippersnapper/
ENTER
*whipper- snapper
*
READY
```

This technique can also be used in our English text when Page 2 fails to hyphenate opportunely or (rarely) incorrectly.

On one occasion the entire article was side-by-side in both German and English. Here we could proceed more elaborately, removing Page 2 hyphenation that was incorrect for German, forcing correct hyphenation paragraph by paragraph.

- Page 2 also has the flaw of assuming that a change in font style permits a break for a new line just as hyphenation or a space does:

```
...... Protection A
       gency ...
```

Text Editor can force a correction by replacing sufficient spaces between words by incompressible en spaces.

- We don't have to worry about losing corrected galleys in the mails, as the *Journal of the Association for Computing Machinery* did in 1971 October. We also know that the corrections have actually been made in the printer's copy, without waiting for a blue to be returned and show that they were *not* made. This often shortens the production cycle, and certainly cuts costs.
- Secretaries can make very creditable copy inhouse by cutting and pasting galley segments with Scotch Tape, and then using a reproduction method such as Multilith. Interoffice memos are becoming artistic, easier and pleasanter to read, and certainly use less paper.

OUR WISH LIST

End users should tell suppliers the nature of their applications and what they would like to have to do these applications better, cheaper, and faster. We would like:

- A larger portion of terminals to be equipped with cassettes. Entering text in the timesharing mode is not efficient in line cost.
- Cassettes attachable to office typewriters. If this means new office typewriters, then let them have standard keyboards! By this is meant that not only the placement of the printing symbols, but also the placement of the controls, either as separate keys, or in the control position on the regular keys. For example, Control-X is the usual position for CANcel (deletes the line just typed). Some keyboard designers have not realized that this makes Control-Z a poor place for EOT, because a slip of one position turns off transmission, with resultant loss of all one's work to that point!

  With an increased portion of input being generated offline, it would appear that the introduction of the computer at the proper point in the copy production cycle permits entry by *less skilled* people, possibly to the point where the original creator of the text and the enterer are one and the same person. One can imagine an author out in the woods typing his rough copy and getting a cassette record. He would mark up the pages as needed and send both pages and the cassette to an editing service, which would enter the cassette contents and make online corrections to the author's copy according to his indications.
- Alternatively we would take a CRT display if it corrects certain faults of existing systems in line runaround, etc.
- And perhaps a pointer system that could indicate both the beginning and end of a string to be identified for a working purpose.
- A registry of available digitized symbols, so that one would know where to buy their representations in a transferable form.
- More than any hardware imaginable, we would like to see the development of a common composition language, and its elements, that is, universally-agreed encodings for printed symbols - their graphemes, their placement, and their style. Elements of a proposal follow:

FEASIBILITY OF A COMMON COMPOSITION LANGUAGE

Production of graphic copy from encoded data is an important component for present and future information retrieval systems. Dot matrix characters on a CRT screen will just not be satisfactory for some purposes. Production of graphic hard copy from an information bank may in the future be cheaper than ordering an existing printed reproduction to be invoiced, found, packaged, mailed, and delivered.

Because future information retrieval will consider many more symbols than those of the present ISO Code, existing and future graphic devices must be connectable to the retrieval system.

Equipments that produce hard (or film) copy may be viewed in the same way that we view computer central processors utilizing different instruction sets and object code, and as we view various numerically-controlled machines. There are single programming languages that are common to many central processors. In N/C, the APT language is processed to produce the CL Tape, which is also common to many processing machines. In both cases the common language is processed by computer to produce instruction for specific and multiple equipments. In both cases the translation capability to specific equipment is usually the responsibility of the manufacturer of that equipment. That this is not so in the composition industry is due to the lack of a standard composition language and metarepresentation of text (with associated characteristics of alphabet or other symbol class, font, size, style, weight, and 2-dimensional positioning). If this existed, it would be a high-level language for copy production which is translated, by computer, to instructions for the various hard-copy equipments. The industry suffers from this lack.

To be feasible, the basic functions of copy production must be similar, even if not carried out in the same way. This appears to be so; it has been proved for the Honeywell Computer Journal, which can also be printed from entry terminals. Indention, font change, size change, etc., seem to operate as primitives.

To construct a general text-processing language, of which the composition language is one part, we need to enumerate the functions and then assign standard encodings to them. The provisions to do so exist in the ISO Code and the associated expansion and extension techniques. The most general mechanism is ESCape, although SO and SI exist. Some 2-character ESCape sequences are now virtually standard in the 7-bit code, and will likely be single characters in the 8-bit expanded code. Examples are Half Line Reverse Feed, Cursor Up.

Utilizing code extension procedures, provisions are made to be able to select unambiguously a group of symbols, a font, weight, size, etc. We then use a key device or pressure display panel with single function buttons. The operator would perhaps press "Cyrillic" (to get the GOST Standard encoding), "8" point on "10", "bold". Each key would generate an ESCape sequence in series, inline in the text. He then uses either a special typewriter keyboard, a standard keyboard with a chart of correspondences, or some other device, to enter the Russian text. One can imagine the total set of symbols paged on a microfiche for back projection on a screen.

Computer programs (postprocessors) are created to translate from this standard language into the actual commands and character inputs for the copy device, which could be 6-level Teletypesetter, Monotype, Photon, RCA Page One and Videocomp, Datel typewriter terminals, IBM Selectric Composer, etc.

Until new entry equipment is made available to conform, similar preprocessors could be written to convert from the various entry conventions to the metarepresentation. This would reduce the translations from $N!$ to $2N$. If all entry equipment would eventually conform, then a further reduction to N occurs, where: N = the number of different composition equipments.

It is expected that this would free the photocomposition industry for expansion in the same way that FORTRAN, COBOL, and ALGOL did so for computational usage. It would provide international standards for alphabet representation.

CLASSIFICATION AND GROUPING OF SYMBOLS INTO PAGES

ISO TC46 (International Standards Organization Technical Committee 46), Documentation, has a Subcommittee 4 on Automation in Documentation. This body has responsibility for collecting and/or developing the pages of encoded symbols. Examples of such pages are:

■ Characters to form natural languages (alphabets)

ISO [DIS 646]
National/accented
Cyrillic [GOST 13052-67]
Greek
Hebrew
Arabic
Sanskrit
Braille
Kata Kana [JISCII]
Kanji
Phonetic
Dactyology [hand signs]
....
Other punctuation [character augments, bullets, rules, bars, leaders, etc.]

■ Symbols of various fields

Aeronautics
Astronomy [Astrology]
Biology, Botany
Business [Commerce]
Chemistry
Ecclesiastic, Fraternal
Electricity, Magnetism
Flowcharts
Games
Heraldry [flags, insignia, arms]
Logic diagrams
Mathematics, Geometry, Physics
Medicine
Meteorology
Money
Music
Philately
Pictorial, Ornaments
Transportation
Typography
Welding
....
Other Scientific

■ Controls for changing point size, weight, slope, font, position relative to the base line, horizontal compression, etc.

An ESCape sequence and prefix character should be proposed for each page of symbols, for registry with ISO TC 97, Computers and Information Processing, which body maintains this registration authority for extension and expansion of the ISO Code.

REFERENCES

[1] R. W. Bemer, Ed., *Computers and Crisis*, Association for Computing Machinery, New York: (1971) 401 pp.

[2] *TEXT EDITOR Quick Reference Manual, Series 600/6000"*, Honeywell Information Systems, Inc., Waltham, MA, DB42, 1972 June.

R.W.Bemer is Staff Consultant to the Vice President, Advanced Systems and Technology, Honeywell Information Systems. His computing career began at the RAND Corporation in 1949. After two years at Lockheed Aircraft Co., he organized the computing departments at Marquardt Aircraft and the Lockheed Missiles and Space Company. He joined the IBM Corporation in 1955 as Assistant Manager of Programming Research, becoming Manager of Programming Systems and then Director of Programming Standards. In 1962 he joined UNIVAC as Director of Systems Programming. In 1965 he spent a year at Bull-GE as Consultant to the General Manager, and was then assigned as Consultant in Phoenix. He received an AB in Mathematics from Albion College (US) in 1940. He has been involved in international and national standardization of computer languages, vocabulary, and character sets since 1960, and is presently Chairman of the International Standards Organization Subcommittee on Programming Languages (TC97/SC5). He was a primary developer of ASCII and has authored some 50 papers, including one on timesharing in 1957 March. He is a Fellow of the British Computer Society and a member of the Association for Computing Machinery and the Data Processing Management Association.

A.R.Shriver is Managing Editor of the *Honeywell Computer Journal*. He started with computers at UNIVAC in 1955 as a programming instructor. While at UNIVAC he trained university faculty members, was responsible for the marketing of large scale computers to colleges and universities, and served as Assistant to the Director of Systems Programming. In 1963 he joined ICL (London) as a Consultant for the world wide introduction of the 1900 Series. He joined General Electric (the division which is now part of Honeywell Information Systems) in 1965 where he has been active in text processing and photocomposition. Mr. Shriver received a BS in Mathematics from Millersville State College (US). He is a member of the Association for Computing Machinery and the American Society for Information Science.

THE ROLE OF A COMPUTER IN THE PUBLICATION OF A PRIMARY JOURNAL

by ROBERT W. BEMER

Honeywell Information Systems. Inc.
Phoenix, Arizona, US

INTRODUCTION

The Honeywell Computer Journal has had some acclaim for social responsibility in the computer milieu and for the extensive and pervasive use of a computer in the publishing function. The basic elements of the latter are described here. The Journal is published simultaneously in hardcopy, microfiche, and magnetic tape with embedded text control. Its mixed-media character is accented by the fact that not all articles in the microfiche and tape editions appear in the hardcopy edition.

Specifically, the copy that you are now reading has been produced by the identical methods of the Honeywell Computer Journal, as are all of the papers in the Methods and Applications Section of these Proceedings. Thus many of the features can be self-descriptive. The only differences are:

- Video Times Roman font is used here (instead of Optima).
- Column width is 242 points (instead of 228).
- Column height is 57 lines maximum (instead of 60).

To reset this paper for the alternate conditions would cost $3.50 per page!

MAJOR COMPONENTS OF COMPUTER USE

The computer plays a major role in:

- Subscription fulfillment.
- Entry of text, tables, and figures.
- Production of photocomposed copy, with justification and hyphenation.
- Control of page layout.
- Proofing of copy.
- Control of readability and style.
- Indexing.

All except the first and last functions are covered in this paper. The first is omitted because it is common, and we have made no innovations; the last because we make little use of this admittedly powerful feature for the Journal per se.

Furthermore, we do not use the automatic pagination features that are available to us, because computers can never be more than dull and pedestrian in this role. It may be suitable for a contract specification, or legal documents, but not for a publication that must be artistic, attractive, and readable. Automatic pagination also chews up expensive store and time to keep the total text in core to work with.

SCHEMATIC OF THE SYSTEM

The HIS 6000 system is used for text entry, editing, storage, and running concordances. It is not normally used for the "run-off" function (producing formatted copy on the entry terminal). Even though this feature is available, it is tedious, expensive in line cost, and has little value for final copy.

Formatted copy is produced only by photocomposition. When this is desired, a special postprocessor program converts the text stream and embeds macros for the Page 2 System. This produces a magnetic tape which is (now) transported physically to the facilities of Datagraphics, in Phoenix, and input to a Univac (nee RCA) 2 driving a III Videocomp 830. The resulting copy is laid up in desired page form, and a cycle of editing and further photocomposition begins.

Final copy is waxed on templates in the traditional manner. Special heads are added (in fonts not available to the computer system, and chosen to symbolize article content, where possible), and it's off to the printers.

Basically, we have adjoined two free-standing systems, and in so doing removed from the middle the expensive and non-graphic-quality output of the first, and the somewhat tedious and inflexible input of the second. Jury-rigged as it is, it is nevertheless superior to any method formerly available to us, and points the way to integrated systems for the future. We can live for now with our 2-hour turnaround.

TEXT ENTRY

Text entry is accomplished in the timesharing mode with the standard HIS 6000 Text Editor System,[1] an embedded format system based upon M.I.T. work and similar to the IBM Script. It is *not* a numbered line system like ATS, and eight years of experience has proved *this* wisdom. Searching and alteration are done primarily in the string mode. "Cut and Paste" is limited to operate by number of lines moved, but they are not numbered.

All control actions are signified by embedded "dot" commands. This input convention states that a CR (Carriage Return) character followed by a full stop character (period, dot) signifies a control statement, thus:

```
.begin      .center   .indent n   .subpara n  .TAB
.space (n)  .adjust   .undent n   .para       .break
```

These are but a subset of the standard Text Editor, and can be learned by an unskilled person in an hour or so. The editing commands will be explained in the running text of this paper.

CONCLUSION

As the 93rd Congress begins its deliberations, the complex question of limiting Federal expenditures will be a primary subject of concern. It is a fact of life that neither man nor nation can live within available resources without reliable information about needs and expenditures. For this reason it is hoped that among the solutions that are devised will be the granting of top priority to the development of the computer system to support the budget and appropriations cycle. Above all other considerations, this is the most critical need of the Congress. With annual expenditures at the $250 billion level, even a minor improvement in the budget and appropriation system would save billions.

Computers are the only hope that our Congress has to acquire the basic data needed to control expenditures. Without this data there can be no effective Congress and, ultimately, no democratic system.

REFERENCES

1. "Strengthening Program Planning, Budgeting, and Management in the Federal Government", McKinsey & Co., 1970 Dec.

Selected Readings

American Enterprise Institute for Public Policy Research, "Congress: the first branch of government", AEI, Wash., DC, 1966. 515 pp. (see papers by C.R.Dechert, K.Janda, J.A.Robinson).

Brooks, Jack, "Data-processing techniques to aid Congress", Extension of remarks in the House, Congressional record [Daily edition], **115**, 1969 Jan 3, pp. E29-E31.

Califano, J.A., Jr., "Congress has been bypassed in analysis technology", the Washington Post, 1971 Jul 13, p. A-18.

Chartrand, R.L., "Congress, computers, and the cognitive process", *In* Planning and politics: uneasy partnership (T.L.Bevle and G.T. Lathrop, *eds.),* 167-187, the Odyssey Press, New York, 1970.

----, "Redimensioning congressional information support", Jurimetrics J., **11**, No. 4, 165-178 (1971 Jun).

Chartrand, R.L., K.Janda, and M.Hugo, *eds.* "Information support, program budgeting, and the Congress", Spartan Books, New York, 1968. 231 pp. (see papers by Chartrand, Janda, W.T.Knox, R.McClory, F.Schwengel).

Comptroller General of the US, "Budgetary and fiscal information needs of the Congress", US Genl. Acctg. Off., Wash., DC, 1972 Feb 17. 49pp.

"Congress looks to computers for legislative help", Congressional Quarterly, 1969 Apr 11, 524-525.

Donham, P., and R.J.Fahey, "Congress needs help", Random House, New York, 1966, 203pp.

"Digitizing Congress", Electronics, 1970 Aug 31, pp. 41-42.

Glass, A., CPR report, "Congress moves into computer age but divides on control of new systems", Natl. J., **2**, No. 22, 1150-1157 (1970 May 30).

Hopkins, B.R., "Congressional reform: toward a modern Congress", Notre Dame Lawyer, **47**, No. 3, 442-513 (1972 Feb).

Hunter, K.W., "Toward better information and analysis support services for the Congress", Proc. Assoc. Comput. Mach., 1971, 139-155, ACM, NY.

Information systems: current developments and future expansion. Proceedings of a special seminar held for Congressional Members and staff, 1970 May 20, AFIPS [Amer. Fed. Information Processing Soc.], Montvale, NJ, 88p.

Janda, K., "Information retrieval -- applications to political science", Bobbs-Merrill, New York, 1968, 230pp.

Kravitz, W., "Preparation of committee legislative calendars by computer: description of a working system", Legislative Reference Service, Library of Congress, 1969 Mar 3. 8pp. + appendices.

Little, A.D., Inc., "Management study of the US Congress", commissioned for special NBC news report, "Congress needs help", 1965 Nov 24, 37p.

McClory, R., Congressman McClory suggests computer uses for Congress. Extension of remarks of Honorable Tom Railsback, Congressional record [Daily edition], **114**, E275-E277, 1968 Jan 29.

Maiorana, Charles. A computer system to aid in Congressional decision making. the George Washington Univ., Wash., DC, 1966 May 1, 40 pp.

Moorhead, William S. Computers for the Congress: the challenge and the potential. An address before the 15th Annual EDP conference of the American Management Assoc., New York, 1969 Feb. 15pp.

Schloss, L., "Congress needs computers -- but they're many years away", Government Executive **2**, No. 6, 42-43 (1970 Jun).

Schneier, E., "The intelligence: information and public policy patterns", Annals of political and social science **388**, 14-24 (1970 Mar).

Semling, H.V., Jr., "Congress and the computer", Modern Data, 1969 Sep, 68-71.

Sisk, B.F., "Toward a more effective Congress", Los Angeles Times, 1969 Oct 22, part 2, p. 1.

Truman, D.B., *ed.*, "The Congress and America's future", Prentice-Hall, Englewood Cliffs, NJ, 1965, 185pp.

US Congress. House Committee on Government Operations, "Use of computers in the legislative process", Report together with supplemental views of the committee, 91st Congress, 1st session. Report No. 91-258, US Govt. Print. Off., Wash., DC, 1969. 39p.

----, House Subcommittee on Government Activities. "Effective and efficient use of computers in Congress", Hearing before the subcommittee, 91st Congress, 1st session, 1969 Apr 23, US Govt. Print. Off., Wash., DC, 1969, 53pp.

US Congress. House Committee on Administration, First progress report of the special Subcommittee on Electrical and Mechanical Office Equipment. 91st Congress, 1st session, October 1969, 64 pp. (see also the second progress report, 20 pp., and the special report of the same committee, 17 pp., 91st congress, 2nd session), US Govt. Print. Off., Wash., DC.

----, House Committee on Rules, "Legislative reorganization act of 1970", Report of the Committee on Rules on H. R. 17654. 91st Congress, 2nd session, House Report No. 91-1215. US Govt. Print. Off., Wash. DC., 1970, 182pp.

----, Public Law 91-510 [84 STAT. 1140]. An act to improve the operation of the legislative branch of the Federal Government, and for other purposes. 91st Congress. H. R. 17654, 1970 Oct 26. 65pp.

----, Senate Committee on Rules and Administration, "Initial report of the Subcommittee on Computer Services to the Committee on Rules and Administration", 92nd Congress, 1st session, 1971 Jul 21, US Govt. Print. Off., Wash. DC, 11pp.

----, Secretary of the Senate. Administrative reorganization of the offices of the Secretary of the Senate. Based on administrative survey by O.B. Potter. 92nd Congress, 2nd session, US Govt. Print. Off., Wash., DC, 1971. 11pp.

ENTRY FOR UNSKILLED PERSONNEL

The postprocessor program that converts for the Page 2 System is vital for simple text entry. The standard entry methods for the Page 2 System are certainly *not* simple, and require some training and a crib sheet constantly on display to the enterer. Remember that graphic quality output requires a separate font generation for each unique character. It is not sufficient to overprint an umlaut (in its fixed position) for both the upper and lower case "u", for example. The postprocessor does extensive string analysis, much of it based upon backspace and overstrike for entry, which makes it simple for personnel. Examples:

- Characters with diacritical marks - accent acute, accent grave, tilde, umlaut, etc. - are produced by backspace on the terminal and overstrike with the proper character (double quote is used for umlaut).
- Double and single quotes are used as they are for entry. The postprocessor determines whether they are opening or closing quotes. A double quote is two single quotes in photocomposition, and this is called automatically.
- For minor occurrences in text, boldface may be indicated by overstriking single characters three times. This is visible on the terminal when the line is verified. For a longer string of bold characters, the font is altered by a **.bold** command, and turned off by a **.bold end** command. These commands do not force a new line.
- For minor occurrences in text, italics may be indicated by backspacing the length of the word and underlining. This is visible on the terminal when the line is verified. For a longer string of italics, the font is altered by a **.ital** command, and turned off by a **.ital end** command. These commands do not force a new line.
- The bulleting seen here is accomplished by a **.indent 3** followed by a **.undent 3** (which is operative only for the next line), a lower case "oh", 2 blanks, and then the text. The uniqueness of this string permits the convention.
- To the regular Text Editor convention of using the "at" symbol to delete the previous character (guess why our articles never contain this character!), and CAN to delete the entire line of entry, we have added the caret to indicate the "en" space, which is incompressible to the justification process. Thus a new paragraph is caused by a **.break** and an initial line with two carets for indention.
- Normal font sizes for the Journal are:

  9 point - text
  8 point - references, some displays as necessary
  7 point - sub- and superscripts, figure captions

  Point size may be changed at any point in the text by inserting the ESCape sequence:

  ESC g (7-pt), ESC h (8-pt), ESC i (9-pt)

  These override the original settings, and are used for formulas, etc.

CONTROL OF PAGE LAYOUT

It has been a remarkable discovery to us that reader attraction and satisfaction is increased significantly by tight control of page layout. Only in the most exceptional cases will a column start in the middle of a sentence, and then only on the second column of the same page. Usually a column will start with at least a paragraph (not just an arbitrary paragraph, but one that makes sense), and very often with a heading. The appearance of a figure or table will never precede its first mention in text, nor will it often be on a page that is not visible when that mention is made. "Widows" never occur.

Under traditional methods, the editor loses control of page layout after the galley stage; all of the niceties must be left to a composer who has little understanding of the subject matter, and is often less interested in reader satisfaction. With the low cost of text processing taken in conjunction with photocomposition, we do not mind expending many runs to get just what we want.

A quick reading of the first galley copy gives an estimate of the author's redundancy or flowery speech factor, and other ways that compression can be achieved if necessary. Accordingly, the actual film is cut to lay out an approximation of the article. As the last page is always full, we work backward. Whatever is left for the first page we leave for artistic treatment and the "From the Editor" commentary. Great attention is paid to aspects of future readability, left or right page assignment, pleasing placement of tables, figures and photos. Virtually no attention is paid to typos and other mistakes that exist in the copy. Accordingly, the single columns are taped on with more lines than our standard, trusting to editing to cut back to the right number (60).

The beauty of this system is that many things can be changed simultaneously to create correctness, harmony, and interest - point size for certain paragraphs or tables, tab settings, subparagraphing, font style, and text changes and corrections. Imagine a situation where the column copy has to be reduced by two lines, and yet previous editing has taken advantage of all short lines at the end of paragraphs, filler words have been removed, and big words replaced by commoner smaller words with equivalent or clearer meaning. Now you have to get into the guts of the author's meaning and say it shorter and clearer, without altering the flavor or meaning in any way! Being forced to do this by our aesthetic standards for page layout yields a big dividend in increased readability.

Depending upon the content, we may photocompose the text from 2 to 5 times. Do the authors complain about the alterations? Never, in our experience. When it reads well, they just assume that they wrote it that way, never checking their original copy. We have also experimented in putting the author's work in to typeset even when it is only rough draft; results seem to indicate that the visualization of final copy permits him to improve it more that he could by editing from a typed draft.

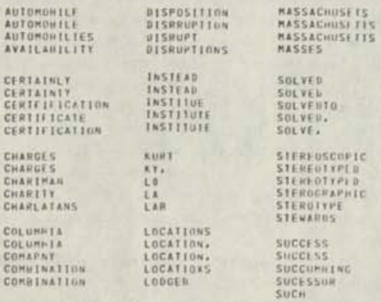
Obviously, taking this much work for readability means high acceptance standards, and we insist that this is a good thing. Dung coated with 53 layers of Chinese lacquer is still dung, and we do not intend contributing to information pollution.

Hopefully, it is now clear why we do not use the computer for automatic pagination.

PROOFING OF COPY

An optional feature, or byproduct, is the concordance run, usually exercised on what is expected to be the next-to-last photocomposition run. This produces two listings on the high-speed (upper case only) printer. The first listing is a Key Word Out of Context (KWOC) listing; each numeral and word (except for the very small common ones) is listed on the left in collating sequence order, with its entire entry line on the right. The lines *are* numbered here, for cross-reference to the second listing, which is the consecutive text.

The concordance is now scanned visually, primarily to detect input errors ("typos"). See Figure 1 for some examples. It is our experience that these fairly jump out at one in scanning a concordance, whereas they remain stubbornly glossed over by the eye and mind in traditional proofreading. However, we do read the text - for style and making sense, not for typos. In fact, knowing that you are freed from the typo-hunting task creates a different frame of mind for doing *real* editorial work.



```
AUTOMOBILF          DISPOSITION        MASSACHUSETS
AUTOMOBILE          DISRRUPTION        MASSACHUSETTS
AUTOMOBILIES        DISRUPT            MASSACHUSETTS
AVAILABILITY        DISRUPTIONS        MASSES

CERTAINLY           INSTEAD            SOLVED
CERTAINTY           INSTEAD            SOLVED
CERTFIFICATION      INSTITUE           SOLVEDTO
CERTIFICATE         INSTITUTE          SOLVED.
CERTIFICATION       INSTITUTE          SOLVE.

CHARGES             KURT               STEREOSCOPIC
CHARGES             KY.                STEREOTYPED
CHARIMAN            LB                 STEREOTYPED
CHARITY             LA                 STEROGRAPHIC
CHARLATANS          LAB                STEROTYPE
                                       STEWARDS
COLUMBIA            LOCATIONS
COLUMBIA            LOCATION.          SUCCESS
COMAPNY             LOCATION.          SUCCESS
COMBINATION         LOCATIOKS          SUCCUMBING
COMBINATION         LODGER             SUCESSOR
                                       SUCH
```

Figure 1. Typos Exposed by Concordance

CONTROL OF READABILITY AND STYLE

The concordance produces a histogram of word size distribution as a byproduct, and the average word length may be calculated. We target 5.0 characters per word, and are very suspicious of readability when the author gets above 5.5.

One aspect of style, or rather one of our rules, is that an acronym shall always be given the spelled-out version in parentheses the *first* time it is encountered in text. One has only to spot the first occurrence in the concordance, and look to the corresponding line on the right to see if this has been done. If not, edit.

The Journal has other style rules. Most important is adherence to ISO Standard 1000, or the International System of Units (SI). Check the concordance for inches, feet, yards, miles, pounds, etc. If they occur, and are for measurement, they had better be in parentheses following a metric value. Other examples: $2 million - not 2 million dollars; 0.5 s - not .5 sec; focused - not focussed.

ECONOMIC CONSIDERATIONS IN WORKING METHODS

As there is no way to predict the pagination of printed copy when entering text, one could enter it all under a single file name. However, the 6000 Text Editor keeps the entire file in the main store for faster processing (and it is *really* fast), and these facilities must be paid for. Thus original input is made in judiciously separated and named files, breaking at headed sections, for example. These are then adjoined for the photocomposition run.

After page layout is determined, they are adjoined again and resplit by page into files with new names, and the old ones purged. This permits single columns to be reworked into final form. The present rate is $1.75 per column. Thus a page costs from $6 to $10 to compose, comparing rather favorably with the $70 per page we were paying for linotype setting to our standards before our system was operable. The 6000 cost is not included, as we have been unable to get real figures because we work on an inhouse "exposure" system used for checking out new software releases. We do, however, feel that this cost is compensated by the system doing automatically what we would have to do ourselves otherwise (like proofreading), and the added quality. We do need to modify to set double column on the last run.

Economy dictates that we should process as much text as possible on each photocomposition run. This means linking several files and saving them as a single file. But this increases the risk that something going wrong early will spoil the balance. Care must be taken to separate and insulate each file from any other. Convention starts each file with **.begin** (for a new galley), **.indent 0** (in case the file ahead of it lacked a command to restore indention to 0), and **.adju** (in case the preceding file had been using tabulation and was not restored to the justification mode).

The power of the Text Editor is of great assistance in checking for correctness of the adjoined file, particularly for closure. Type:

```
fs:/.bold/;*     (meaning "find all occurrences of that string")
```

and you will almost instantly get a message like:

```
end of file - request executed 122 times
```

Hit "b" and CR (for backup to the file beginning, and type:

```
fs:/.bold end/;*
```

If the message doesn't say 61 times - trouble! A 60 would mean that bold did not get turned off somewhere, and the copy following will be in useless boldface. Do the same for italics, subparagraphs, point size changes, etc.

The files must always be correct for the magnetic tape edition, and identical to the printed copy. Yet it is often wasteful to rerun the entire file for simple patches. A copy is made, and the correct parts wiped out by string replacement, leaving only the changed copy to be reset as a patch (with due consideration to leaving enough text so that paragraphing, etc., is unchanged). These patches are saved under a different name; a number of them are adjoined and run at one time.

INCIDENTAL ADVANTAGES

A number of dividends have shown up that we amateurs did not really foresee:

- Doing our own typesetting permits laying up mechanicals for articles as soon as they are ready, without waiting to group an entire issue for the typesetter to schedule in some time slot. Exclusive of conditions of extreme timeliness, this permits better selection for issue makeup and content.
- Having the feel of the final product, by mockup during the editing and changing stages, affects everyone - author, editor, and reviewer. For the latter, particularly, it gives psychological impetus to hurry up - lest what he dislikes might be in the finished product. All can work simultaneously to correct and improve the copy and make it more readable.
- The Page 2 System hyphenates to English rules and/or custom. Normally we run our French, German, Italian, and Spanish sections in "fill mode" (stretching the spacing between words to fill the line without hyphenation). But if glaring gaps exist we remove them easily by doing a dummy hyphenation, splitting the first word of the next line into two components:

  ```
  rs:/whippersnapper/
  ENTER
  *whipper- snapper
  *
  READY
  ```

  This technique can also be used in our English text when Page 2 fails to hyphenate opportunely or (rarely) incorrectly.

  On one occasion the entire article was side-by-side in both German and English. Here we could proceed more elaborately, removing Page 2 hyphenation that was incorrect for German, forcing correct hyphenation paragraph by paragraph.
- Page 2 also has the flaw of assuming that a change in font style permits a break for a new line just as hyphenation or a space does:

  ```
  ...... Protection A
         gency ...
  ```

  Text Editor can force a correction by replacing sufficient spaces between words by incompressible en spaces.
- We don't have to worry about losing corrected galleys in the mails, as the Journal of the ACM did in 1971 October. We also know that the corrections have actually been made in the printer's copy, without waiting for a blue to be returned and show that they were *not* made. This often shortens the production cycle, and certainly cuts costs.
- Secretaries can make very creditable copy inhouse by cutting and pasting galley segments with Scotch Tape, and then using a reproduction method such as Multilith. Interoffice memos are becoming artistic, easier and pleasanter to read, and certainly use less paper.

OUR WISH LIST

A major purpose of the First National Computer Conference and Exposition was to have the end users tell the suppliers the nature of their applications and what they would like to accomplish those applications better, cheaper, and faster. I must follow my own principles. We would like:

- A larger portion of terminals to be equipped with cassettes. Entering text in the timesharing mode is not efficient in line cost.
- Cassettes attachable to office typewriters. If this means new office typewriters, then let them have standard keyboards! By this I mean not only the placement of the printing symbols, but also the placement of the controls, either as separate keys, or in the control position on the regular keys. For example, Control-X is the usual position for CANcel (deletes the line just typed). Some keyboard designers have not realized that this makes Control-Z a poor place for EOT, because a slip of one position turns off transmission, with resultant loss of all one's work to that point!

  With an increased portion of input being generated offline, it would appear that the introduction of the computer at the proper point in the copy production cycle permits entry by *less skilled* people, possibly to the point where the original creator of the text and the enterer are one and the same person. One can imagine an author out in the woods typing his rough copy and getting a cassette record. He would mark up the pages as needed, and send both pages and the cassette to an editing service, which would enter the cassette contents and make online corrections to the author's copy according to his indications.
- Alternatively we would take a CRT display if it corrects certain faults of existing systems in line runaround, etc.
- And perhaps a pointer system that could indicate both the beginning and end of a string to be identified for a working purpose.
- A registry of available digitized symbols, so that one would know where to buy their representations in a transferable form.
- More than any hardware imaginable, we would like to see the development of a common composition language, and its elements, that is, universally-agreed encodings for printed symbols - their graphemes, their placement, and their style. Elements of a proposal follow:

FEASIBILITY OF A COMMON COMPOSITION LANGUAGE

Production of graphic copy from encoded data is an important component for present and future information retrieval systems. Dot matrix characters on a CRT screen will just not be satisfactory for some purposes. Production of graphic hard copy from an information bank may in the future be cheaper than ordering an existing printed reproduction to be invoiced, found, packaged, mailed, and delivered.

Because future information retrieval will consider many more symbols than those of the present ISO Code, existing and future graphic devices must be connectable to the retrieval system.

Equipments that produce hard (or film) copy may be viewed in the same way that we view computer central processors utilizing different instruction sets and object code, and as we view various numerically-controlled machines. There are single programming languages that are common to many central processors. In N/C, the APT language is processed to produce the CL Tape, which is also common to many processing machines. In both cases the common language is processed by computer to produce instruction for specific and multiple equipments. In both cases the translation capability to specific equipment is usually the responsibility of the manufacturer of that equipment. That this is not so in the composition industry is due to the lack of a standard composition language and metarepresentation of text (with associated characteristics of alphabet or other symbol class, font, size, style, weight, and 2-dimensional positioning). If this existed, it would be a high-level language for copy production which is translated, by computer, to instructions for the various hard-copy equipments. The industry suffers from this lack.

To be feasible, the basic functions of copy production must be similar, even if not carried out in the same way. This appears to be so; it has been proved for the Honeywell Computer Journal, which can also be printed from entry terminals. Indention, font change, size change, etc., seem to operate as primitives.

To construct a general text-processing language, of which the composition language is one part, we need to enumerate the functions and then assign standard encodings to them. The provisions to do so exist in the ISO Code and the associated expansion and extension techniques. The most general mechanism is ESCape, although SO and SI exist. Some 2-character ESCape sequences are now virtually standard in the 7-bit code, and will likely be single characters in the 8-bit expanded code. Examples are Half Line Reverse Feed, Cursor Up.

Utilizing code extension procedures, provisions are made to be able to select unambiguously a group of symbols, a font, weight, size, etc. We then use a key device or pressure display panel with single function buttons. The operator would perhaps press "Cyrillic" (to get the GOST Standard encoding), "8" point on "10", "bold". Each key would generate an ESCape sequence in series, inline in the text. He then uses either a special typewriter keyboard, a standard keyboard with a chart of correspondences, or some other device, to enter the Russian text. One can imagine the total set of symbols paged on a microfiche for back projection on a screen.

Computer programs (postprocessors) are created to translate from this standard language into the actual commands and character inputs for the copy device, which could be 6-level Teletypesetter, Monotype, Photon, RCA Page One and Videocomp, Datel typewriter terminals, IBM Selectric Composer, etc.

Until new entry equipment is made available to conform, similar preprocessors could be written to convert from the various entry conventions to the metarepresentation. This would reduce the translations from N! to 2N. If all entry equipment would eventually conform, a further reduction to N occurs, where N = the number of different composition equipments.

It is expected that this would free the photocomposition industry for expansion in the same way that FORTRAN, COBOL, and ALGOL did so for computational usage. It would provide international standards for alphabet representation, to aid the UNISIST project.

CLASSIFICATION AND GROUPING OF SYMBOLS INTO PAGES

ISO TC46 (International Standards Organization Technical Committee 46), Documentation, has a Subcommittee 4 on Automation in Documentation. This body has responsibility for collecting and/or developing the pages of encoded symbols. Examples of such pages are:

- Characters to form natural languages (alphabets)

ISO [DIS 646]
National/accented
Cyrillic [GOST 13052-67]
Greek
Hebrew
Arabic
Sanskrit

Kata Kana [JISCII]
Kanji
Braille
Phonetic
Dactyology [hand signs]
....
Other punctuation [character augments, bullets, rules, bars, leaders, etc.

- Symbols of various fields

Aeronautics
Astronomy [Astrology]
Biology, Botany
Business [Commerce]
Chemistry
Ecclesiastic, Fraternal
Electricity, Magnetism
Flowcharts
Games
Heraldry [flags, insignia, arms]
Logic diagrams
Mathematics, Geometry, Physics

Medicine
Meteorology
Money
Music
Philately
Pictorial, Ornaments
Transportation
Typography
Welding
....
Other Scientific

- Controls - for changing point size, weight, slope, font, position relative to the base line, horizontal compression, etc.

An ESCape sequence and prefix character should be proposed for each page of symbols, for registry with ISO TC 97, Computers and Information Processing, which body maintains this registration authority for extension and expansion of the ISO Code.

REFERENCE

1. "TEXT EDITOR Quick Reference Manual, Series 600/6000", Honeywell Information Systems Inc., DB42, 1972 June.

1972

1971

editorial

Commencing with this issue, you will see a major change in the *Honeywell Computer Journal*. We think that it is a fundamental change (not just a face-lifting), and wish to describe the several facets of it.

Honeywell, the first word in our name, is an international company of major importance. Only 58,000 of the 100,000 employees reside within the US. Our management is committed to promoting strong programs of public service. Therefore, you will see a greater accent on articles of public interest to people worldwide. With Vol. 5, No. 1 we initiated the use of international A4 page size. In this issue we establish full usage of SI (Systéme International) units as the primary means of measurement; in certain cases the old English equivalents will be given in parentheses. In addition, we shall endeavor to interpret (usage) differences between Europe, Asia, the Americas, etc., to the better understanding of all. Many times this will benefit the US most, as the nonmetric country.

The *Computer*, the second word in our name, is ubiquitous. It has application, of varying degree, to every discipline, to every phase of today's existence. Our content will mirror this broad scope. In addition to new development we shall, for example, include flavorful annotated bibliographies, fundamentals, and digests of useful standards. Wherever possible, we shall revisit computer history that needs straightening out or illumination, relate the innovative thread, and retrieve the otherwise overlooked. This is useful for a field that has had such rapid growth.

Journal, the third word in our name, implies a serial publication intended to impart useful information effectively. We are aware of the increasing demands upon people's time and mental capacity. With so much information available today, of highly variable quality and importance, which articles should someone take the time and energy to read? Where should someone go to find the important articles?

Technical knowledge is increasingly employed only by the few, but its impact is becoming increasingly felt by the many. We shall, therefore, frame our articles to explain the importance and value of the work described. Sometimes we may indicate how an application can benefit society, how an innovation may save money, or how a standard may help us all to talk more meaningfully to each other. Because the decision to read certain articles may come primarily from these framings (**"FROM THE EDITOR"**), they and the abstracts they augment have been translated into French, German, Italian, and Spanish. You will find them in the back of the Journal. We recognize that reading the main article in English represents an effort and investment by many of our employees and other readers, and hope that these translations will facilitate their decision to do so.

If, as we have said above, "computers are ubiquitous", then we will be looking for the common thread that holds us all together as humans and yet permits variation in the way we do things. We intend to take our readers on a guided tour of this exciting field so that we may all understand the computer in the context of its contribution to people, as a social tool.

the honeywell computer journal

VOLUME FIVE, NUMBER THREE, 1971

CONTENTS

- WITH PERSONAL COMPLIMENTS OF THE EDITOR -

Bob Bemer

Computer
History
Museum

SEPARATION SHEET
Computer History Museum

Category of item(s):

- □ Audiocassette
- □ Electronic records
- □ Oversized materials
- ☒ Photographs/negatives
- □ Printed materials
- □ Three-dimensional objects
- □ Videotape
- □ Other: ____________

Number of items: 12 35mm slides (in one slidesheet)

Provide information about item(s):
Presentation slides found in 1971 part of folder. First slide has "Manageable Software Engineering" written on it.

Originally filed in:

Catalog #: 102785430 Lot #: X3054.2005

Collection Name: Robert (Bob) Bemer papers

Series: Papers Subseries: ____________

Folder title: Speeches and papers, 1955-1982
folder 3 of 5

Relocated to:

Catalog #: 102785615 Lot #: X3054.2005

Collection Name: Robert (Bob) Bemer papers

Series: Papers Subseries: ____________

Folder title: Presentation slides, 1970-1978

Purpose of removal: To separate slides from text.

Date: 2/28/19

Name: Sydney Olson

DINNER

Roast Sirloin Au jus

Baked Potato   Peas/Mushrooms

Tossed Mixed Green Salad

Hot Rolls
Butter   Choice of
Beverage

Hot Apple Pie
Honey-Rum Sauce

NOTES

ACM

5th ANNUAL SPRING JOINT MEETING

Phoenix - Rio Grande -Student Chapters

CONVENTION CENTER - SAFARI HOTEL
4611 SCOTTSDALE ROAD, SCOTTSDALE

MONDAY, MAY 10, 1971

COMMITTEES

PROGRAM CHAIRMAN
William A. Gardner

VICE CHAIRMAN
Mary Alice McKune

REGISTRATION
Patricia Inman

PUBLICITY
Ken Roberts
Marjorie McCarthy

ARRANGEMENTS
George B. Peck

OFFICERS

RIO GRANDE CHAPTER

CHAIRMAN
Don Robbins

VICE CHAIRMAN
William A. Gardner

SECRETARY/TREASURER
Douglas Haden

PHOENIX CHAPTER

CHAIRMAN
Jerry Smallidge

VICE CHAIRMAN
Mary Alice McKune

SECRETARY/TREASURER
Patricia Inman

ASU STUDENT CHAPTER

CHAIRMAN - Michael Jordan

AGENDA

REGISTRATION............8:00 - 8:30

WELCOME.................8:30 - 8:40
Jerry Smallidge, Honeywell
Chairman, Phoenix Chapter

RESPONSE................8:40 - 8:45
Don Robbins, Sandia
Chairman, Rio Grande Chapter

CONTOUR MODEL OF COMPUTATION
John Johnson, NMSU...8:45 - 9:20

FM LOG - COMPUTERIZED FEDERAL
COMMUNICATIONS..........9:20 - 9:45
Alan Chickinsky, NMSU

SADIE SWITCH............9:45 -10:05
Art Arenholz, Sandia

INTERMISSION...........10:05 -10:25

ON-LINE WITH THE PDP-1010:25 -10:50
John Spitzer Sandia

COMPUTERIZED VOTING IS GIVING
THAT DEVICE A BAD NAME 10:50 -11-20
Bob Bemer, Honeywell

ACM AND THE OMBUDSMAN 11:20 -11:45
Jack Tischhauser, Sandia

LUNCH.................11:45 - 1:00

DATA STRUCTURES - TECHNIQUES AND
IMPLEMENTATIONS.........1:00 - 3:00
(Short Talks and Panel Discussion)

I A Brief Introduction to Data Structures - Jim Morris, LASL

II TIDY and INDEX - Machine Independent FORTRAN Utility Codes
Harry Murphy, Kirtland

III SPIL - Special Purpose Input Languages - Jennie Boring, LASL

IV DATA STRUCTURE IN MADCAP VI
- Jim Morris, LASL

V SYNTAX DIRECTED COMPILING
- Bob Conley, Kirtland

VI A TECHNIQUE FOR LEXICAL ANALYSIS
- Bob Mitchell, LASL

INTERMISSION............3:00 - 3:15

ORGANIZING ANCILLARY FILES OF THE
PERSONNEL DATA BASE.....3:15 - 3:45
George Connor, Sandia

STATUS REPORT ON EPIC...3:45 - 4:10
R. D. Brown Jr., ECPI

COMPUTER SCIENCE NEEDS
ITS LABORATORY......... 4:10 - 4:40
Richard Stark, NMSU

HOW TO GET YOUR COMPUTER
PEOPLE TO WORK AS HARD
AS YOU DO .... 4:40 - 5:00
R. E. Van Allen, Honeywell

Free Time .... 5:00 - 6:00

Attitude Adjustment Time 6:00 - 7:00

Banquet .... 7:00 -

GUEST SPEAKER - Gordon Smith
Executive Director, ACM

TOPIC - What You Always Wanted To Know About ACM, But Were Afraid To Ask.....

1970

I SIMPOSIO INTERAMERICANO DE COMPUTACION

16 de Noviembre de 1970

CENTRO DE CONVENCIONES
PASEO DE LA REFORMA 445, 1ER PISO
MEXICO 5, D. F.

INSTITUTO LATINO AMERICANO
DE CIENCIAS DE LA INFORMACION
Y LA COMPUTACION

ILACIC

INSTITUTO LATINO AMERICANO
DE CIENCIAS DE LA
INFORMACION Y LA COMPUTACION, A. C.
PASEO DE LA REFORMA 445, PLANTA BAJA
MEXICO 5, D. F.

NO ENVIE EFECTIVO POR CORREO

Difícilmente podrá encontrarse persona alguna de aquellas que trabajan en las áreas de la Computación Electrónica y de los Procesos de Información, que no conozca los ágiles y famosos SIMPOSIA que a partir del año 1958 y bajo la dirección del Prof. FRED GRUENBERGER, organizó la RAND CORP., en Santa Mónica, Calif.

Estos Simposia se contaron, año con año, entre los eventos que mayor interés despertaron en la comunidad computística de los Estados Unidos y de otros diversos países, entre los cuales frecuentemente se contó a México, ya que algunos expertos mexicanos asistieron a ellos.

Las Memorias de estos Simposia, constituyen una de las fuentes de referencia más importantes acerca de los problemas y las soluciones que se han planteado en el área de la computación electrónica, de la programación, de la educación en esta área, de la teoría de lenguajes algorítmicos, de la arquitectura de los sistemas de computación y muchos otros más.

A partir de 1970 y bajo la dirección conjunta de FRED GRUENBERGER y de SERGIO F. BELTRAN, estos Simposia serán celebrados en México, ahora con un carácter interamericano y bajo el patrocinio del

INSTITUTO LATINOAMERICANO DE CIENCIAS DE LA INFORMACION Y LA COMPUTACION (ILACIC)

El primer Simposio de esta nueva serie, estará integrado por los conocidos expertos que aparecen en la lista impresa en este mismo comunicado.

Este Simposio será organizado en forma similar a la que rigió los diez simposia anteriores. Los expertos invitados discutirán durante dos horas (de las 9:00 a las 11:00 hs.) y en presencia de los asistentes inscritos, los temas incluídos en la agenda. Y estos asistentes podrán plantear preguntas y aclaraciones (previamente formuladas por escrito al Director de Sesión), durante la hora lsiguiente (de las 11:00 a las 12:00 hrs.)

Programa similar se seguirá en la sesión vespertina, o sea, discusión entre los expertos invitados de las 13:30 a las 15:30 hrs. y discusión ampliada con los participantes, de las 15:30 a las 16:30 hrs.

LISTA PARCIAL DE EXPERTOS INVITADOS

Andree, Richard	Universidad de Oklahoma
Amdahl, Gene	IBM Corp.
Armer, Paul	RAND Corp.
Bemer, Robert	GE Company
Bergstein, Harold	Computer Machinery Corp.
Bright, Herbert	Computation Planning Inc.
Campbell, Sullivan	Graphic Sciences, Inc
Davidson, Charles	Enginering Computing Lab.
Forest, Robert	DATAMATION (Editor)
Glaser, George	McKinsey and Co. Alemania Occidental
Gruenberger, Fred	San Fernando Valley College Computing Center
Halstead, Maurice	Purdue University Computer Sciences Dept.
Little, Jack	Planning Research Corp
McCracken, Daniel	Consultor y autor
Poland, Clarence	IBM Corp.
Powell, Kenneth	IBM Corp.
Tomash, Erwin	Data Products Corp.
Van Norton, Roger	University of Arizona Computing Center
Wagner, Francis	Informatics, Inc.
Weizenbaum, Joseph	M. I. T. Proyecto MAC
White, Robert	Informatics, Inc.

16 de noviembre de 1970
CENTRO DE CONVENCIONES
Paseo de la Reforma 445, Primer piso

Sesión matutina: 9:00 a 12:00 horas.
Sesión vespertina: 13:30 a 16:30 horas.

Las inscripciones se reciben hasta el día 13 de noviembre de 1970

CUPO LIMITADO

Desprenda a lo largo de esta línea discontinua

I Simposio Interamericano de Computación

FORMA DE INSCRIPCION

Nombre ....
Apellido Nombre Propio

Dirección particular .... Telef. ....

Lugar de trabajo ....

Dirección .... Telef. ....

Deseo ser considerado como Participante Local en el I SIMPOSIO INTERAMERICANO DE COMPUTACION.

Para este efecto envío a ustedes:

Cheque [ ] ; Giro Postal [ ] ; Giro Telegrráifco [ ] ; Giro Bancario [ ] ; Efectivo [ ] ;

por la cantidad de $.................... (..)
Número Letra

que cubren el monto de .... inscripción(es), a razón de $300.00 M. N. cada una.

El costo de esta(s) inscripción(es) incluye un número igual de ejemplares de las MEMORIAS de este SIMPOSIO, que recibiré una vez que sean publicadas.

....

INTER-AMERICAN COMPUTING SYMPOSIUM

Mexico City, November 16, 1970

Co-sponsored by: Instituto Latino Americano de Ciencias de la Informacion y la Computacion (ILACIC)

and: San Fernando Valley State College.

Prof. Sergio Beltrán, ILACIC PASEO DE LA REFORMA 445 - TEL 511-9426/5370
PLANTA BAJA
MEXICO 5, DF
HOME 548-4360

Prof. Fred Gruenberger, SFVSC 349-1200 X1137

Dr. Gene Amdahl, IBM

Prof. Richard Andree, University of Oklahoma

Mr. Paul Armer, Harvard University

Mr. Robert Bemer, General Electric Co.

Mr. Mort Bernstein, System Development Corp.

Mr. Herbert Bright, Computation Planning, Inc.

Dr. Sullivan Campbell, Graphic Sciences, Inc.

~~Prof. Charles Davidson, University of Wisconsin~~

Mr. George Glaser, McKinsey & Co.

Prof. Maurice Halstead, Purdue University

? Mr. Jack Little, Planning Research Corp.

Mr. Daniel McCracken,Author and Consultant

Mr. Clarence Poland, IBM

Prof. Roger van Norton, University of Arizona

Mr. Francis V. Wagner, Informatics Inc.

Mr. Robert White, Informatics Inc.

Dr. Heinz Zemanek, IBM (Austria) - SEND SOPL

Sr. Guerra

Heinz Zemanek

Fred Gruenberger

Sullivan Campbell

George Glaser

Enrique Prasnyski

Dr. Pedro Solís Cámara

Bob Bemer

Sergio F. Beltrán

Maurice Halstead

Robert White

VICTOR RAMIREZ

Dan McCracken

Sergio Alcaraz

José Luis Peña

Guillermo Encinas

Lauro Cuevas

Sergio Beltrán Jr.

Paul Armer

Mort Bernstein

ENGR. Miguel Vargas PEMEX

Gene Amdahl

Ulises Beltrán

Frank Wagner

Roger Van Norton

Herbert Bright

J.M. Copeland

Clarence Poland

Sigfrido Hurtado

Roberto Bermúdez

Raúl Camargo

J.A. Sánchez

COMPUTING AND PROFESSIONALISM

- We have accepted professions to use as models: law, medicine, the clergy, engineers, CPA's.

- Each of these professions has standards that are promulgated and accepted; there are avenues to acquiring the accepted body of knowledge; they are licensed by the state; there are procedures for lifting the license; they have codes of ethics; misuse of the license is grounds for civil and criminal action. Those who qualify and are licensed are given a mantle of authority, a seal of quality, privileged communication (for lawyers), a license to kill (for doctors)--in return for which, the user accepts a level of responsibility. Can these things be made to fit computer people?

- How do we establish what quality computing is?

- What mechanism could be set up by the state to license, examine, qualify, and disbar people?

- Thus, even if it might be desirable to have something called a Certified Public Computing Professional, is it feasible to set up the machinery to do it, in the same sense as it has been done for the professions listed above?

1133 AVENUE OF THE AMERICAS
NEW YORK, N.Y. 10036
(212) 265-6300

ACM Association for Computing Machinery

3625 W. Sierra Vista
Phoenix, Arizona 85019
November 12, 1970

Mr. R. W. Bemer
2 Moon Mountain Trail
Phoenix, Arizona

Dear Bob:

On behalf of the Phoenix Chapter-ACM, I want to thank you for your presentation last Tuesday evening at our November meeting.

The results of ACM-70 are even more far-reaching than most of us realized. You are to be congratulated for your contribution to the ACM-70 Program.

We all wish you the best of luck in your new position and look forward to seeing you back at a Chapter Meeting real soon. Again, thanks for a most interesting and timely presentation.

Very truly yours,

A. J. Smallidge, Chairman
Phoenix Chapter - ACM

AJS/fc

MEETING

ASSOCIATION FOR

ACM

COMPUTING MACHINERY

A.J. SMALLIDGE 263-5554

The Phoenix Chapter of the ASSOCIATION FOR COMPUTING MACHINERY will hold its November meeting Tuesday, November 10, at 7:30 p.m. Location is Loyola Hall at Brophy College, 4701 North Central Avenue, Phoenix, Arizona.

Guest speaker for the meeting will be ROBERT W. BEMER, Program Chairman - ACM 70. His subject will be RESULTS OF ACM 70. The impact of ACM 70 will not be lost. Already it is influencing the nature of its successor in 1971. Plans for this conference to be held in Chicago are now being generated, based on the questions and problems defined at ACM 70. Bob Bemer has been involved with these questions and problems and will give us a first-hand report on the results.

PROGRAM CHAIRMAN'S MESSAGE

Computers and information systems, motive power of so many disciplines, will have an unprecedented impact on the future structure of industry, government and society.

The first effects are evident. Many of us are aware that the computer is understood inadequately by its users and even less by the general public. Many of us feel that the computer industry, with its phenomenal growth rate, is in a state of some disarray and lacks a sense of direction and purpose.

The structure of the ACM 70 program tells its purpose—to have the computer industry be responsive to the present and future needs of the end users of its products and services. We realize that these users are the actual reason for the existence of the ACM as a professional society.

There is a need for an integrated plan, or functional specification, for the development and proper utilization of computer based systems in the next decade. Many of the components for such a plan do exist, but they are not yet in understandable national perspective for the several combinations of the end users, professions and data processing resources. Nor do we have adequate inventory of the present computer use to base our projections on it, or to formulate the necessary technological, social, political and legislative strategies.

Attendees to ACM 70 will not hear all the solutions to all of the problems. The Program Committee believes, however, that ACM 70 is a demonstrable model of how to work to achieve these solutions. ACM 70 will unfold an exciting spectrum of the total computer industry in a comprehensive way never before achieved.

Robert W. Bemer

ALL COMPUTER RELATED PERSONNEL IN THE AREA ARE INVITED
VISITORS ARE ALWAYS WELCOME

Report of the SHARE Data Processing Committee

October 2, 1957 - San Diego, California

The SHARE Data Processing Committee under the chairmanship of Tom Steel (RL), met at 8:30, October 2, in the Cotillion Room of the El Cortez Hotel. The following people were in attendance:

Name	Installation	Name	Installation
R. D. Acker	IBM	Peggy Johnson	SC
A. G. Montgomery	SP	Barbara Lemke	NA
Frank R. Heath	WK	H. A. Wood	CV
R. Habermann	GD	D. E. Hart	GM
R. A. Brouse	RL	D. F. Harroff	GM
L. Gatt	LA	Bert Coudriet	CW
R. Danck	NT	G. H. Mealy	BE
J. Clabaugh	GD	R. W. Bemer	IBM
H. E. Williams	GD	D. A. Hemmes	IBM
H. N. Cantrell	GS	B. Oldfield	NY
Ascher Opler	DC	W. F. Bauer	RW
S. R. Shapiro	RL	Eldo C. Koenig	AC
D. C. Leagus	BE	M. Butler	AN
F. B. Smith	GN	F. S. Beckman	IBM
M. T. Guss	GC	H. L. Stevens	NT
H. I. Morrison	CE	B. J. Blasdell	CU
W. T. McKinney	CA	Wm. Orchard-Hays	CE
T. R. Dines	CA	M. Perry	RW
G. Puente	GA	D. W. Gantner	RW
A. Downing	AV	Charles W. Backman	DC
J. R. Stock	UC	David Feign	CU
F. M. Verzuh	MI	Charles Jaeger	NO
J. Heller	NU	M. Senko	PK
H. D. Leeds	VG	W. S. Willis	GI
J. B. Wyatt	CF	T. B. Steel, Jr.	RL

The first hour and a half of the meeting was spent with general considerations contrasting the difference in data processing as envisioned by the people doing data reduction of a scientific or engineering nature and those engaged in commercial data processing. It was agreed that initially there would be no attempt to split into two separate areas and a common ground would be established if possible. The objective as initially stated by the chairman was to draw up a minimum package of programs for the 704 and 709.

These definitions seemed to be very helpful during the afternoon session.

Charles Bachman, (DC), was chairman of the afternoon session in the SHARE suite. Approximately twenty persons were in attendance. Input and output generators were the extensive subjects of the afternoon session. The output generator as constructed by General Electric at Hanford, Washington for their IBM 702 was extensively discussed. (It was understood that they were in the process of converting their 702 routine to a 709 routine.) This output generator reads a series of punched cards which simulate a line of printer form for the 407 in cards plus a small additional set of cards to define extra functions necessary and then will sort, extract, summarize, or convert as necessary to translate the data in a file into the required report. The General Electric Hanford people stated that they would be very pleased to receive suggestions of functions that should be added to the output generator. It was believed by the people at the meeting that the output generator with its logic converted could be a prototype of an input generator which would convert, test for duplication, validate characters, justify data left or right in the particular machine words, test for overflow or data too large for the areas reserved in the machine and sort into the desired sequence to update the file. This input generator should either work from data converted from cards and develop it into the desired file or should take the data from cards plus an old tape and produce a new tape file as desired. It was generally agreed that the computer should continue to process regardless of type of error that it discovered and provision should't have to be made as to alternate courses of action according to the error found.

One danger was felt in attempting to achieve too complicated input and output generators for the 704-709 in that the more complex they were, the greater would be the time to perfect them. The conclusion of the afternoon session was that a working committee should establish a glossary of terms that could be used consistently by the people working in the data processing area and that they should define the type of input and output generators that should be constructed. A number of people offered to work on the committee. They are: Charles Bachman (DC), John Scott (UC), Bob Acker (IBM, New York), Chuck Jaeger (NO), William Orchard-Hayes (CE), Dick Brouse (RL), Bruce Blasdell (CU), and Rowan Coyle (NS).

A meeting is being arranged in Midland during the month of November. All those actively interested are invited to get in touch with Charles Bachman (DC).

Respectfully submitted

T. B. Steel, Jr. (RL)
Chairman,
SHARE Data Processing Committee

Feb. 28, 1958

X

REPORT OF SHARE DATA PROCESSING COMMITTEE

The SHARE data processing committee began their meetings Tuesday afternoon without a great deal of feeling as to what their purpose was and were they were going....

As a point of clarification in the two 709 groups, the macros and subroutine group is starting with SCAT symbolic language and desires to build up to a higher level data processing language. The report (RE-FORMATTING) and file maintenance generator group ← SORTING is starting with the pictorial language now in use at Hanford and desires to build down to SCAT symbolic language.

There is an earnest hope in minds of some that they can and will meet on some middle ground. There are others who have their doubts....

709 Report and File Maintenance Generator Subcommittee
Chairman - Russell McGee (GH)

Installations:			
	GH	G. E. Hanford	
	UC	Union Carbide	JOHN SCOTT
IBM-BOB ACKERMAN?	SP	Northern States Power	
	DC	Dow Chemical	C. BACHMAN

Russell McGee (GH) reported on the logic operation of the Mark II File Maintenance Generator now under preparation.

A present summary of a data processing package is the following:

1) IBM has agreed to prepare a data processing package in accordance with the desires of the majority of the Data Processing Committee.

2) A study of "compilers vs generators" indicates that generators best satisfy the data processing requirements.

REPORT ON THE FIFTH MEETING OF

THE SHARE 704 DATA PROCESSING COMMITTEE

XI

The meeting was held on September 8-9, 1958 at the St. Francis Hotel in San Francisco, California.

In attendance were: Bob Dinsmore (AS), Charles Cooper (CE) William Orchard Hays (CE), Bill Dobrusky (DA), Evelyn Austin (MB), Fred Malone (MD), Fletcher Jones (NC), and Paul Tani (NC).

I. The system is about half-coded. Components will be checked out by October 15. Preliminary users manual will also be written by then. The system should be ready for field test by December 1.

SHARE XII

Harry Nagler of IBM gave a brief report on the 709 Data Processing package which he has developed. This subject evoked sufficient interest to warrant a seperate meeting sometime this week. A similar question and answer session on the generalized routines was felt to be needed. The date and time of this meeting have yet to be arranged and will be posted on the bulletin board in the registration room.

704 Data Processing Committee, Fletcher Jones (NC). Jones announced that the committee is just about to release the 704 Data Processing System now called the SURGE. He indicated that general distribution will be made after some ~~ten~~ installations have done field testing. The manual will be distributed at the same time that general distribution of the SURGE package is made.

704 Data Processing Committee - Fletcher Jones (NC). Jones announced that the committee's work would be completed roughly one month following the SHARE meeting, and suggested that the committee be dissolved. He pointed out that Edwards Air Force Base had volunteered to maintain the 704 Data Processing Package after the field tests would be completed, which he expected would be four to six weeks following the SHARE meeting.

Data Processing Committee - Russel McGee (GH). McGee announced that at the Thursday afternoon meeting of the committee, Ellen Kerksieck of IBM had discussed SORT 709 and MERGE 709 with some supplemental remarks by Jim Liser(AD). He added that as a result of a lively interest in sorting, Chuck Jaeger (NO) will act as a subcommittee chairman on sorting techniques. In addition, he pointed out that Mary Ann Savas (TR) has volunteered to rewrite the manual of the generalized routine with help from Charles Thoma (RW) and Erwin Danziger (GD). Charles Bachman (DC) is attempting to arrive at new names for the report generator and file maintenance programs. Anybody having ideas should forward the suggestions to Bachman.

Mary Ann Savas (TR) volunteered to rewrite the Generalized Routines with the help of Charles Thoma (RW) and Erwin Danziger (GD).

It is generally recognized in the committee that the names Report Generator and File Maintenance leave something to be desired. Charlie Bachman (DC) is attempting to arrive at new names for the systems. Any suggestions would be appreciated and should be sent to Charlie.

The committee's activities at SHARE XII will be concluded in a meeting this afternoon in which Harry Nagler of IBM will discuss his Data Processing Subroutine Package.

Russ McGee

REPORT OF DATA PROCESSING COMMITTEE

The period since SHARE XII has been marked by several significant developments insofar as the Data Processing Committee is concerned. First, the generalized routines (Report Generator, File Processor, and 9PAC SORT) developed by the committee were completed and put into productive use. Second, the meeting of potential 7090 data processing users at WDPC on June 3 and 4 was constituted as a meeting of the Data Processing Committee. During this meeting the 9PAC and COMTRAN Subcommittees were formed. Kendall Wright (WD) was appointed Chairman of the 9PAC Subcommittee; George Tait (PP) will be the Chairman of the COMTRAN Subcommittee.

A working group of the 9PAC Subcommittee was formed to complete the documentation of the Report Generator, File Processor, and IB9SRT modified for 9PAC - the routines contained in 9PAC. Kendall Wright will submit a report of the progress to date of the working group.

Another significant development affecting SHARE Data Processing users is the announcement by IBM Applied Programming that they are looking into a variable record length sort which will use the standard files as defined in 9PAC for its input and output. This is significant both for the basic utility of a variable record length sort and because of the urgent need for such a program in producing a sorting Report Generator. The proposed specifications for this new sorting routine will be evaluated by the Sort Subcommittee under Chairman Chuck Jaeger (NO).

Data Processing: Russell McGee (GH). McGee stated that the activities of the Data Processing Committee since SHARE XII have been concentrated in the 9PAC Subcommittee which was formed in Los Angeles early in June 1959. The Committee organized the working group which has been working in Los Angeles and in Richland since the middle of June. He stated they have done a monumental job of completing the documentation of the 9PAC systems, copies of the 9PAC Manual were to be distributed in the mail boxes for those installations which have not yet received them. In addition, the 9PAC Subcommittee held two days of indoctrination on Report Generator, File Processor, and SORT, the first two days of the week. Ninety-two people were in attendance and the presentation was well received. In reply to a question from Herb Bright (WB), McGee could not state the date of publication for the complete systems manual for 9PAC.

SURGE Committee: Paul Tani (NC). Tani reviewed the several SURGE meetings held during the SHARE Meeting and indicated that there would be some proposed changes under study in the next several weeks. He announced that there would be a meeting for those interested in SURGE for the 709-7090 in Columbus, Ohio on September 14-16, 1959. He closed with the statement, "We no longer have the urge to purge SURGE."

SHARE XIII
C.16.1
59 AUG 19-21

SESSION 5D. REPORT ON SURGE

REMARKS BY PAUL TANI, NC.

HISTORY OF SURGE

At t.he SHARE X meeting in Washington, DC, in February, 1958, an ad hoc subcommittee of the SHARE Data Processing Committee was set up to create a commercial data processing system for 704 users.

At the first meeting in Chicago in April, it was decided to use as a model, the 702 generalized routines prepared at GE-Hanford. The next two me etings in New York City were devoted to agreeing on specifications for the system. By August, flow charts of some of the portions were drawn and some coding was done. More of the confusion was removed at a meeting in Columbus, Ohio, in August. The subcommittee's fifth meeting was held in September, 1958, in conjunction with the SHARE XI meeting in San Francisco. At that time, the system was half-coded. In October, a preliminary manual was published. An attempt to put the parts of SURGE together late in October at Washington D. C. was not too successful. Later efforts at Martin Baltimore were more fruitful.

By February, 1959, the system had a name, SURGE, (Sorter, Updater, Report Generator, Etc.) and at the SHARE XII meeting in New York City, it was announced that SURGE was being field-tested. The second SURGE manual started to take shape soon after that.

By June 15, 1959, the field test was brought to a close, the symbolic decks were updated, re-assembled, and the decks (symbolic and binary) were mailed to Edwards Air Force Base, whose 704 installation accepted the responsibility for distributing and maintaining SURGE. Copies of the second manual have been sent to the SHARE Distribution Agency, which will send these manuals to each SHARE installation if the proper order card is returned to the SDA.

The 704 installations which contributed the system designers, programmers, and 704 time are CEIR (Council for Economic and Industrial Research), Martin-Baltimore, Martin-Denver, Aerojet-General, Douglas-El Segundo, IBM-Endicott, General Motors Research, and North American Aviation-Columbus.

THE SURGE LANGUAGE

It was the intent of the SURGE system designers to create a language which would permit non-704 coders to use the 704 to perform commercial data processing. Earlier, IBM created FORTRAN to permit scientists and engineers to use the 704 without learning the 704 basic language. It was hoped that accountants and tabulating analysts could use the 704 easily via SURGE. As it turned out, experienced 704 programmers like to use SURGE for many commercial data processing problems because SURGE simplifies the preparation of a problem, it shortens the coding time, and the efficiency of SURGE programs is for the most part as good as SAP-coded programs.

SURGE is a file-oriented system. Files are created, updated, read to extract shorter files, sorted, and/or read to prepare reports.

The SURGE Source Statement Layout was decided on rather than a coding format with an operation and operands as in SAP or FORTRAN. Most of those who have used the fixed format, like it. If one wants to read an item, he merely puts the directory code for the item under READ. A three-address system is provided for arithmetic operations. Comparisons are easily coded. Items are written on output tapes by merely writing the code of the file under WRITE. Reports are generated by preparing a "Tab Packet" and filling out lists and preparing a pictorial display of each report line. Sorting a tape is ridiculously simple. Merely write the code of the file to be sorted under SORT.

DEFINITIONS OF TERMS USED IN SURGE MANUALS

BIT: Smallest piece of information in a binary machine. O or 1.

CHARACTER: A symbol relating to one of fifty permissible groupings of six bits. The fifty symbols are the alphabet, the digits, and 14 special characters.

FIELD: Conglomeration of bits or characters. Fields are either integers or alphanumeric.

INTEGER FIELDS (Binary fields): Length on a BCD tape, 1 to 10 characters and the sign is an overpunch over the leading or trailing digit. On a binary tape, 2 to 36 bits. First bit is a sign bit. The other bits express the magnitude of the integer. In core, an integer occupies one full 704 word.

ALPHANUMERIC FIELDS (BCD fields): Length up to 255 characters. BCD fields in core are left justified. For example, an 8 character field occupies the 6 characters of one word and the first 2 characters of the next word.

ITEM: A logical record. The set of fields pertaining to one entity in a particular logical file. In a logical file, all items are constructed identically. Only one type of item per logical file is permitted.

RECORD: A physical record. BCD tapes may have one or more records per item. Binary tapes may have one or more items per record.

SESSION 5D. REPORT ON SURGE

SURGE! AN EVALUATION BY A USING INSTALLATION-George Carroll, NY.

1. Our part in the development - None as such; however, we have contributed machine time at our installation for an important revision.

2. USES:

(1) Non 704 personnel prepared files and generated reports with little difficulty. Prime difficulty was in mis-use of SURGE occasioned by original writeup, which was not clear enough. Present version of writeup has proven satisfactory.

(2) 704 personnel quickly wrote a file maintenance and interrogation report which cleaned up master files of over one million items.

3. FUTURE USES:

Entire installation of about 85 personnel will use SURGE for data processing operations.

4. CONCLUSIONS:

(1) Excellent contributions to data processing.
(2) Simple to use.
(3) Savings in lead time to reports.
(4) Non 704 personnel have become productive very rapidly.
(5) Fine idea of tape files - is recommended for all tape systems.

5. RECOMMENDATIONS:

A. GENERAL - All installations doing Data Processing should use SURGE and compare with 704 SAP type programming. Remember, however, most of our users have employed non 704 personnel.

B. SPECIFIC -
(1) Provide shifting of alpha fields
(2) No stops in compiler without statements
(3) No stops at logical end of data file - just on-line print notice
(4) Un-numbered source statements should be referred to in some way
(5) Set up move and replace statements to operate in same way.

George F. Carroll
Service Bureau Corp. (N. Y.)

SESSION 5D. REPORT ON SURGE - Bert Coudriet (CE)

CE has been using SURGE for roughly four months. We have at this time 5 programmers that are familar with this type of coding system. These do not include those programmers who took part in the actual coding of "SURGE". We have used SURGE in many ways, into many different types of problems. Let me stress that each problem was carefully analyzed before the use of SURGE was adopted. At no time were we sorry we used this coding system to do any problem, but in certain cases its use did not buy us what we anticipated.

In general we have used the coding system in 3 main types of problems.

1. Typical accounting types of problems.
2. One shot data handling endeavors.
3. Feasibility studies of elaborate data handling systems.

To enumerate on these types in sequences we are quite pleased with "SURGE'S" ability to handle the accounting type procedures. It blends itself very well with report generations, besides giving flexability at a time when most needed, i. e., one month after the problem is in production. Also, the customer always has a knack of forgetting some thing very important until two days before the deadline. Instead of going out and committing suicide as was the feeling in the past, we now explain to the customer what a terrible crime he has committed, and then update our flow charts.

In one shot data handling program the advent of SURGE has saved the day. No longer does it take 4 programmer weeks to process Joe Blows non-standard unheard of formats. I believe its use in this field is self explanatory.

In the third field that I have mentioned the use of SURGE is not as profitable as in the other two. Somehow Joe Jones problem always has just enough little dos and donts that with the combination of all existing coding systems, you still have a nasty problem.

When we work with "SURGE" for awhile its good points are often overlooked and the programmer tends only to look at what SURGE will not do. I do not wish to belittle the system in any way but I must state at this time that a few things are lacking. Whether they can be placed in the system easily or not, I do not know. Possibly at this meeting some of these questions can be answered.

1. We need a REW statement
2. We need the ability to manipulate the Tape End of File Counter.
3. We need the ability to write tape pockets in 14 word records, i. e., to simulate the 716 (Card to tape).
4. An octal program picture may be practical as a 3rd file on tape 3.
5. Write a SURGE "EXECUTE" program tape that contains Multi Programs.

SESSION 5D. USERS REPORT ON SURGE - Lee N. Caplan.

GENERAL ELECTRIC EVENDALE INSTALLATION

We first became interested in SURGE through our J-93 Parts History Program. This program will generate a file of information 25 tapes in length by 1964. The J-93 people not knowing at the present time what reports they would desire, asked us to look at variable report generators. We investigated several - MSVD-GE, Hanford GE and North American's SURGE. SURGE seemed to fit our requirements and seemed simple to use. A few hours with Paul Tami convinced us that this was the case. So we made our master tape using Cage instructions but generating it in SURGE Format so that it could be usedfor any report generators desired in future times.

In so doing this we discovered that SURGE, more than being a report generator would also serve as an automatic programming device for most of our data processing problems. We felt it would solve a great deal of problems that have been plagueing us; First off, the high cost of programming a business application and the attendant time lag. Second, our staff has grown from 2 to 14 people in Business Systems, only 3 or 4 experienced in both 704 and data processing problems. Getting these people is difficult to say the least. We feel we can now go with our present staff as SURGE should handle 95% of our applications.

The people we deal with like to use their own systems personnel to do a problem rather than explain it to a programmer as time and an intangible something else is always lost. SURGE makes this possible. We feel that computing is cheapest on the largest machine. Therefore we would like to remove applications from the 650 and place them on the 704. For instance, work has begun on our inventory problem. The 650 requires 4 days, 5 hours a day to do the job. The report is obsolete by the time the job is finished. The 704 will do the job in one day 15 minutes using SURGE. The programming is being done by the Systems people concerned. In addition, we will not chew up 35,000 master cards each week. The only cards we use will be the activity for the week, some couple hundred cards.

I was able to teach the inventory people SURGE in 3 days - 1-1/2 hour sessions and am doing this with other people in our plant. So far we have done a service analysis report for our Field Engineer which strained the capacity of the 8K version but is all right on the new variable 8K to 32K version, and, as I mentioned before, we are working on the inventory problem. We have experienced at least a 10-1 reduction in programming and programs estimated at two months now require a week. Block diagramming takes longer than the source statementing.

Our present problem is our tabulate, sort, collate problem for our commercial engines parts. The tabulation people inform me that the tabulate codes are fantastic as compared to the board wiring they are required to do for sub-totaling. In this program I have married CAGE and SURGE by placing both on our tape and rewinding the tape for each case. I leave SURGE by sub-routine to call in each CAGE program rather than by stopping in the usual manner as I have 256 cases to run. Each case took 14 hours on hand equipment as opposed to 6 minutes by using the CAGE and SURGE marriages on the 704. Naturally in using this we have found some 15 changes to be absolutely necessary to make SURGE applicable to 95% of our problems. I have written up these changes and presented them to Dottie Clark, our technique expert and SHARE representative. I wish to present them to SHARE people also for their comment, suggestions and help in getting them into SURGE as quickly as possible.

Lee N. Caplan
Business Systems Specialist
Evendale Computations
S. E.

SURGE - List of Changes Desired.

The following list of additions and changes for SURGE were submitted to me by L. N. Caplan and A. R. Barton of Business Systems Unit. At present they find SURGE limiting but if these changes can be accomplished it is estimated that 95% of the work can be done with it.

1. An error out on decimal to binary conversion. As the program now stands, program stops, the error is merely noted, the offending bits removed, operator presses 'start' and the program proceeds. We would like a jump to a location where we can place a subroutine to take individual action according to the needs of a specific program instead of the present stop.

2. Only one directory field format is allowed for an input tape. We would like this to be a general field format so that we may read the tape, determine what type card this is by checking a field and then place the card into an individual directory field format. It would be good if we could allow for about five to ten individual directory field formats per tape input.

3. It is difficult or impossible to go back and forth from CAGE to SURGE and vice-versa. Is it possible to facilitate this ?

4. A table look-up feature. As the program now stands, you can tell if a field is in a table, but cannot tell which one matches.

5. An alphabetic shift. The program allows for integer shifts only. We would

like to shift left or right six bits at a time. This is, left shift 3 would move the field left eighteen bit positions.

6. There is a decimal to binary conversion routine in SURGE. We would like to use it instead of jumping to a subroutine when we desire a conversion.

7. There is no way to pick out a block of storage in SURGE. That is, if we wanted to work on a particular block of words we cannot refer to an address to pick up this block.

8. There is no internal sort routine in SURGE. There is an external sort, that is, a tape sort. Could we possibly incorporate our our own sort routine into SURGE and use it, or better still, give us the option of using either ?

9. We would like to pick out certain characters within a field without shifting. To do this a character mask is required.

10. FLOP has a very convenient way of skipping lines. As SURGE now stands, we must write a source statement to double skip each time. If there is a lot of spacing required on a report, a large number of source statements is required. Could

SURGE - List of Changes Desired -

something like the FLOP skipper be built to SURGE ?

11. In order to take advantage of the ten-line/record capacity of our 720 Printer and save writing and printing time, could the output of SURGE be made optional for the 720 or 717 ?

12. Decimal points are automatically inserted in numeric fields on the SURGE Format. It would be nice if dashes or slashes could be automatically inserted within alphabetic fields for output.

13. With some source statements you may actually place and use a constant in field A, B. However, you cannot do this with a replace command. We would like to replace with an actual constant in field B.

14. Something like a Fortran "Do" statement would be very handy for a group of SURGE source statements.

Additional comments made by Vernon Walfield, Electronic Data Processing Machines, Lynn Computations Operation (sic):

1. SURGE goes to the trouble of converting dummy fields. This increases machine time considerably on longer jobs. This is no hindrance in programming, but we wish that SURGE would ignore dummy fields.

2. Group suppression will not function for numerical fields. Works O. K. with defined BCD fields.

3. The operation "shift right and truncate" does not operate when it is used alone (that is, not preceded by an arithmetic operation in the same source statement.) this may be due to a bug in our SURGE binary deck.. We haven't tried this with our newest deck.

4. The tab packet prints values of fields in total lines which are current when the line is printed, but not necessarily current for the detail lines being totaled. This occurs only under certain conditions. We are currently working on ways to correct this.

Dorothea S. Clarke
Specialist Automatic Coding
General Electric - Evendale
August 1959

SHARE XIII

Session 14D, 15D. Friday, August 21, 1959

SURGE WORKING COMMITTEE

Present: Evelyn Austin (MB), Lee N. Caplan (GE), George Carroll (NY), Dorothea S. Clarke (GE), Wil Couch (EL), Bert Coudriet (CE), W. B. Dobrusky (DA), Mary Ferguson (MD), R. F. Greer (BA), D. E. Hart (GM), David S. Hoffman (GR), D. Holmes (CALTEX), J. C. Johnson (BA), C. W. Libby (GA), William P. Melcher (UA), Robert E. Pugh (DI), Edna Stevens (MF), J. Strella (RF), Paul Tani (NC), C. E. Wright (UW).

The known bugs in SURGE are to be removed as soon as possible. When this is done, SURGE will be sent to the SDA in both binary and symbolic form. Thereafter, errors will be corrected through established channels.

Requests for modifications were considered. Most of the original creators of SURGE are no longer available for coding these changes so that modifications for the most part will have to be made by new members of the SURGE Working Committee. The installations which indicated that they _may_ assign one or more members to this committee are DA, CE, GA, GE, EM, GR, MB, MD, MF, NC, NY, and RF.

A rough estimate when minor modifications may be sent out for field test is December 31, 1959.

Since there was a great deal of unfinished business, it was decided to hold a two-day working committee meeting in Columbus, Ohio on Thursday and Friday, September 17-18, 1959. This meeting follows a three-day discussion of Commercial Translator, 9PAC, and SURGE and the possible development of another compiler for DP work on the 709-7090.

XIII

REPORT OF THE 9 PAC SUBCOMMITTEE

Prior to SHARE XIII the 9 PAC Subcommittee prepared preliminary manuals. These were the Users' Reference Manual, The Operator's Manual and a Systems Manual. A copy of the Users' Manual has been distributed to all SHARE members attending SHARE XIII. Manuals may be obtained by contacting the SHARE Distribution Agency.

During the coming six months different types of monitors will be reviewed by committee members in hopes that a firm recommendation may be made by SHARE XIV. Also, the File Processing and Reports Generator programs will be studied to determine how to obtain object decks in absolute binary.

As soon as the binary programs and the symbolic programs are in agreement, the 9 PAC programs will be turned over to IB. IB will compile the program using SOS and will distribute SQUOZE decks upon request. In the meantime, copies of the programs may be obtained by sending a tape for each program to IB, attention of William P. Heising. A letter will be put in the SSD when the SQUOZE decks are ready for distribution.

K. R. Wright, Chairman
9 PAC SUB COMMITTEE
DATA PROCESSING COMMITTEE

A few significant conclusions can be drawn from the presentations of users, namely:

1. Experience indicates generally successful results with use of IB9SRT.
2. DC, TR, ML and GH have all used 9PAC productively.
3. Troubles have been encountered by many installations with the 709 tape system and in at least three installations with the card reader.
4. There is a general impression that processing with RG and FM on the 709 is slower than people anticipated. There are no good measurements at the present time to substantiate this impression.

During the last hour of Session 15 the future of variable length sorting was discussed. Ron Pulfer (GH) suggested that a variable length sort as well as several of the longer range objectives of the 9PAC subcommittee could be achieved with approximately the same effort required to write the variable length sort from scratch if an appropriate compiler generator were written. This immediately suggested a tie-in with the efforts of the 704 Data Processing Committee. It was decided that (GH) would prepare a proposal for the generator-compiler to be submitted to the meeting of the 704 Data Processing Committee in September. Charles Bachman (DC) volunteered to represent the Data Processing Committee at Columbus in September.

Respectfully submitted,

R. C. McGee
Chairman, Data Processing Committee

60 FEB 17

SHARE XIV
B.19.1

704 SURGE SUBCOMMITTEE REPORT TO SHARE XIV

I. History since SHARE XIII

A. SURGE has been distributed to at least 31 installations by EC.

3. 704 SURGE was frozen except for corrections.

4. Plans were made for distribution of SURGE, and an accompanying systems manual.

5. Channels for the maintenance of the system were established. (SEVERAL INSTALLATIONS. RESPONSIBLE FOR DIFFERENT PARTS

4. 704 SURGE II:

Many worthwhile alterations and additions have been suggested. Inserting these in the present SURGE would require time as well as additional interested personnel. The present committee does not consider these additions to be essential and, unless a new interest is strongly indicated, they will not be made.

5. The present committee's stand is to learn how to live with the present SURGE. This can best be done by communication among 704 SURGE users. It was suggested that explicit subroutines or the concepts involved that could be of general application be distributed through normal SHARE channels, SSD or SDA.

SHARE XV – 60 SEP 11

The 704 SURGE subcommittee did not meet between the times of SHARE XIV and SHARE XV. It did achieve its primary goal of distributing the 704 SURGE system as SDA 877 in April, 1960. The channels for maintaining the 704 SURGE system were established and the first binary patch to the system was distributed as SDA 906. As a result of discussions held at SHARE XIV, SSD communication concerning 704 SURGE usage was increased for the benifit of all concerned. The goal of a completed system manual with descriptions and flow charts was not achieved. Some portions were completed and these along with other available information were provided in limited numbers at SHARE XV. All future information will be distributed as SSD's.

No 704 SURGE meetings were held during SHARE XV and none are planned for the future. The committee is continuing its maintenance function and any information concerning this should be sent to the committee chairman and to the SHARE Secretary for SSD distribution.

Respectfully submitted,

Ivan L. Bowman, 1/Lt., USAF (EC)
Chairman 704 SURGE Subcommittee

SHARE XV
C.2.1

Report on 709-90 SURGE Subcommittee Meeting
September 12, 1960 - 9:00 A.M.
and
September 15, 1960 - 11:00 A.M.

Since a system checkout session was held the week preceding SHARE XV, it was felt that an open meeting would be more beneficial than a closed committee session. Len Longo (GE), subcommittee chairman, announced that no new target is being set for 709-90 SURGE field test distribution. The original target date had been set for SHARE XV.

The current status of the system was presented. Two system assemblies have been made including all components except the Pass III absolute to SCAT disassembly. Passes I and II of the compiler have been checked out together and are working harmoniously except for a few bugs.

Up-to-date programming manuals and sample forms are available and may be obtained upon request from Len Longo, General Electric Company, Evendale, Ohio, phone extension 704. Those who already have older versions of the manual will receive a copy without requesting it. The old forms are up-to-date except for the I card.

A report on the current status of QD SURGE was also given. The system is ready for field test with only one known bug. This bug concerns the report generator and an entrance to left field if START is pushed on the error comment "Conversion error. Press START to ignore." GE is currently maintaining QD and a binary field test deck and listing may be obtained from Len Longo. The 704 SURGE manuals and coding forms apply to QD SURGE with the exception of tape assignments. These assignments have been made as follows:

	704 SURGE	QD SURGE
for compilation	1	A1
	2	A2
	3	A3
for execution	1	A2
	2	A3

QD read and write buffering seems to be very efficient (the sort is not buffered) as initial tests indicate approximately a 4 - 1 time ratio on the 709 for 704 programs recompiled and run with QD. Under the compatibility program, only about a 3 or 4 - 1 improvement was realized on the 7090.

The September 15 9/90 SURGE parallel session was largely a question and answer period. Since most of the topics discussed were included in the above paragraphs and the specific questions will be answered in the manual, no further report on the parallel session seems necessary.

Mary Ferguson (MD)

Techniques

R. W. BEMER, Editor; F. A. WILLIAMS, Assistant Editor

SURGE: A Recoding of the COBOL Merchandise Control Algorithm

Leonard F. Longo

General Electric Company, Evendale, Ohio

Introduction

Many among us feel that English-type languages are not necessarily the best solution to the problem they try to solve. Some of these objections may be quoted from [1]:

> ". . . computer marketers would have us believe, it is unnecessary for the user of a large-scale computer, with its associated English language compiler, to learn a new programming language—he (the programmer) can write his programs in good old familiar English. In fact, the English 'understood' by computers is not the English spoken by you and me." "Looking at the matter of language from the point of view of the computer programmer, then, computer English fails on two counts: it is difficult to write precisely enough for comprehension by the translator (compiler) program and it takes too much writing to express a simple command." "The great length of the source program increases the probability of transcription error, while the requirement for precision makes such errors costly." ". . . how has English language programming (computer English) reached the position of importance it now occupies in programming technology?" "Sad to say . . . the illusion that by obtaining a computer with an associated English Language Compiler, they (management) can avoid selling their company down the river to a bunch of technicians, has enormous sales appeal. So they decide to go ahead, using Fact or Flowmatic or Comtran or Cobol."

History of SURGE

SURGE (Sort, Update, Report Generate, Etcetera) was created by a Share (the IBM 704, 709 & 7090 users group) subcommittee. The compiler, completed in 1958, is being used in activities such as accounting, inventory control, payroll, etc., on IBM 704 machines. In the second quarter of 1960 the Surge compiler was modified for the IBM 709-90. While this version (known as QD Surge) is in use now, work is nearing completion on a newer more powerful edition of the Surge compiler for the IBM 709-7090. All versions of Surge are complete BOL languages including within themselves automatic Reporting and Sorting.

Surge is a fixed-format language in the form of a check-off sheet which resolves many of the objections to English-statement-type compilers. Using this system, the programmer merely describes his input and output within fixed-formats and checks off the action he desires on another fixed-format sheet. The actions possible are listed at the top of the format sheet.

Personnel are easily and quickly trained in Surge. A two-week (20 hours) Surge familiarization course is adequate to train new people. Programs are easily and quickly written. Messrs. A. Todd and M. Hochdorf of the TVA put it this way, ". . . (Surge is) for the analyst experienced in tabulation work. It has been a thrilling experience to see how girls who previously had only done board wiring could, after a few days' training in Surge, write programs for the. . . (computer)." Special requests for reports dealing with information on a master tape are written in about one hour by people who are familiar with Surge coding but who have never done any machine language programming. The automatic tabulation of accounting totals and formating of complete reports are features especially pleasing to people accustomed to EAM operations.

The Merchandise Control Problem

Upon reading the article on Cobol [2] showing a sample problem, it was decided to see what this problem would look like coded in Surge. The difference was, to our eyes, significant enough in terms of length and simplicity to warrant publication of an article on Surge. This should be of interest to the data processing community in that it will allow comparison of two different philosophies. The Surge source coding is presented here without reprinting the flow charts of the problem to be solved (Figures 1, 2).

In this Surge example, the cards through the set with an L in the first position are comparable with the Cobol set of cards up to 008800. The set of Surge source cards with an S in the first position are the Surge procedural statements and are equivalent to Cobol cards numbered 008800, to and including the last one, 015900.

We would like to emphasize:

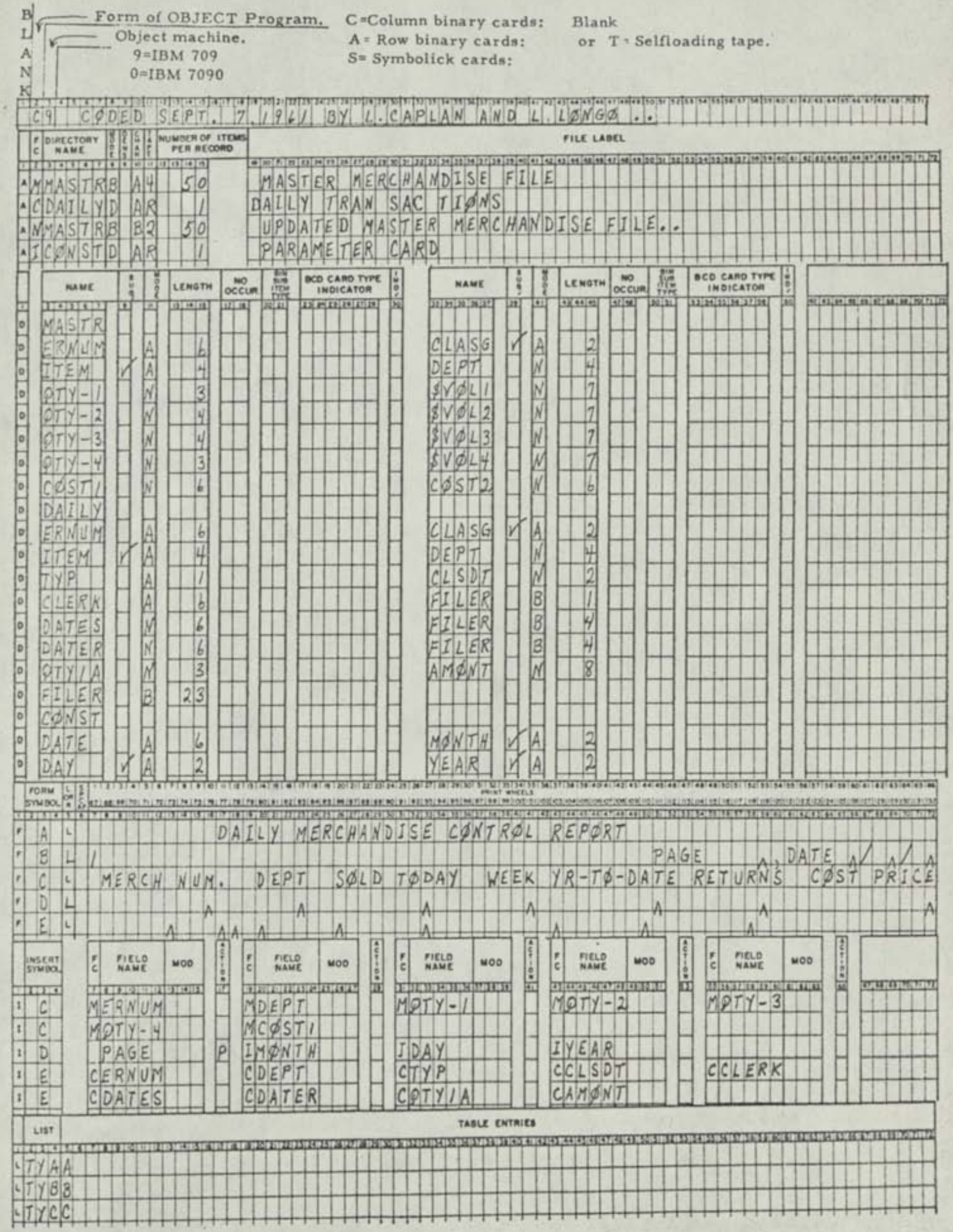
(a) The reduction in the number of cards needed (Cobol's 167 vs. Surge's 83).
(b) The reduction in keypunching per card.
(c) The Surge lexicon is in fact quite readable and of course there is *no ambiguity*.
(d) The coder needs no elaborate set of punctuation rules.
(e) The fixed location of each function makes for much lower compile and re-compile time, as compared to English language statements.

As to readability, card S1 says RI, that is, the source program card identification is 1 and says to Read an item of information from File I. This will result in storing for future use the information from the Date Parameter card. The next two statements say to "Read, and Test for End"

62 FEB

and item from file C, the Daily Transactions file, and M the unupdated Master Merchandise file. On END go to the statement numbered MED or TED respectively.

To define each mnemonic would of course be too space-consuming, but one more description is in order. S9 through the next five cards read: Write an item on 'N', the updated Master Merchandise file; write the Master Merchandise Report as follows: report to be printed on '2' (off-line IBM 717), tape for this to be on channel 'B' unit '4'; write '60' lines to a page; when starting a new page, write the three Heading lines depicted on cards F 'A', F 'B', and F 'C'; write Detail lines on the rest of each page as depicted on card F 'D', including on each detail line the current values of the fields named



B L A N K

Form of OBJECT Program. C=Column binary cards: A= Row binary cards: S= Symbolick cards:

Blank or T = Selfloading tape.

Object machine.
9=IBM 709
0=IBM 7090

C9 CØDED SEPT. 7.1961 BY L.CAPLAN AND L.LØNGØ ..

	FC	DIRECTORY NAME			NUMBER OF ITEMS PER RECORD	FILE LABEL
A	M	MASTR	B	A4	50	MASTER MERCHANDISE FILE
A	C	DAILY	D	AR	1	DAILY TRAN SAC TIØNS
A	N	MASTR	B	B2	50	UPDATED MASTER MERCHANDISE FILE..
A	I	CØNST	D	AR	1	PARAMETER CARD

	NAME			LENGTH	NO OCCUR	BIN SUB ITEM TYPE	BCD CARD TYPE INDICATOR		NAME			LENGTH	NO OCCUR	BIN SUB ITEM TYPE	BCD CARD TYPE INDICATOR	
D	MASTR															
D	ERNUM		A	6					CLASG	✓	A	2				
D	ITEM	✓	A	4					DEPT		N	4				
D	ØTY-1		N	3					$VØL1		N	7				
D	ØTY-2		N	4					$VØL2		N	7				
D	ØTY-3		N	4					$VØL3		N	7				
D	ØTY-4		N	3					$VØL4		N	7				
D	CØST1		N	6					CØST2		N	6				
D	DAILY															
D	ERNUM		A	6					CLASG	✓	A	2				
D	ITEM	✓	A	4					DEPT		N	4				
D	TYP		A	1					CLSDT		N	2				
D	CLERK		A	6					FILER		B	1				
D	DATES		N	6					FILER		B	4				
D	DATER		N	6					FILER		B	4				
D	ØTY/A		N	3					AMØNT		N	8				
D	FILER		B	23												
D	CØNST															
D	DATE		A	6					MØNTH	✓	A	2				
D	DAY	✓	A	2					YEAR	✓	A	2				

FORM SYMBOL — PRINT WHEELS

```
F A L              DAILY MERCHANDISE CONTRØL REPORT
F B L /                                                PAGE     ^, DATE ^/  ^/  ^
F C L  MERCH NUM.  DEPT  SØLD TØDAY  WEEK YR-TØ-DATE RETURNS  CØST PRICE
F D L        ^        ^        ^        ^        ^        ^        ^
F E L     ^     ^^ ^     ^     ^     ^     ^     ^
```

INSERT SYMBOL	FC FIELD NAME	MOD	ACTION	FC FIELD NAME	MOD	ACTION	FC FIELD NAME	MOD	ACTION	FC FIELD NAME	MOD	ACTION	FC FIELD NAME	MOD	ACTION
I C	MERNUM			MDEPT			MØTY-1			MØTY-2			MØTY-3		
I C	MØTY-4			MCØST1											
I D	PAGE		P	IMØNTH			IDAY			IYEAR					
I E	CERNUM			CDEPT			CTYP			CCLSDT			CCLERK		
I E	CDATES			CDATER			CØTY/A			CAMØNT					

LIST	TABLE ENTRIES
L TYAA	
L TYBB	
L TYCC	

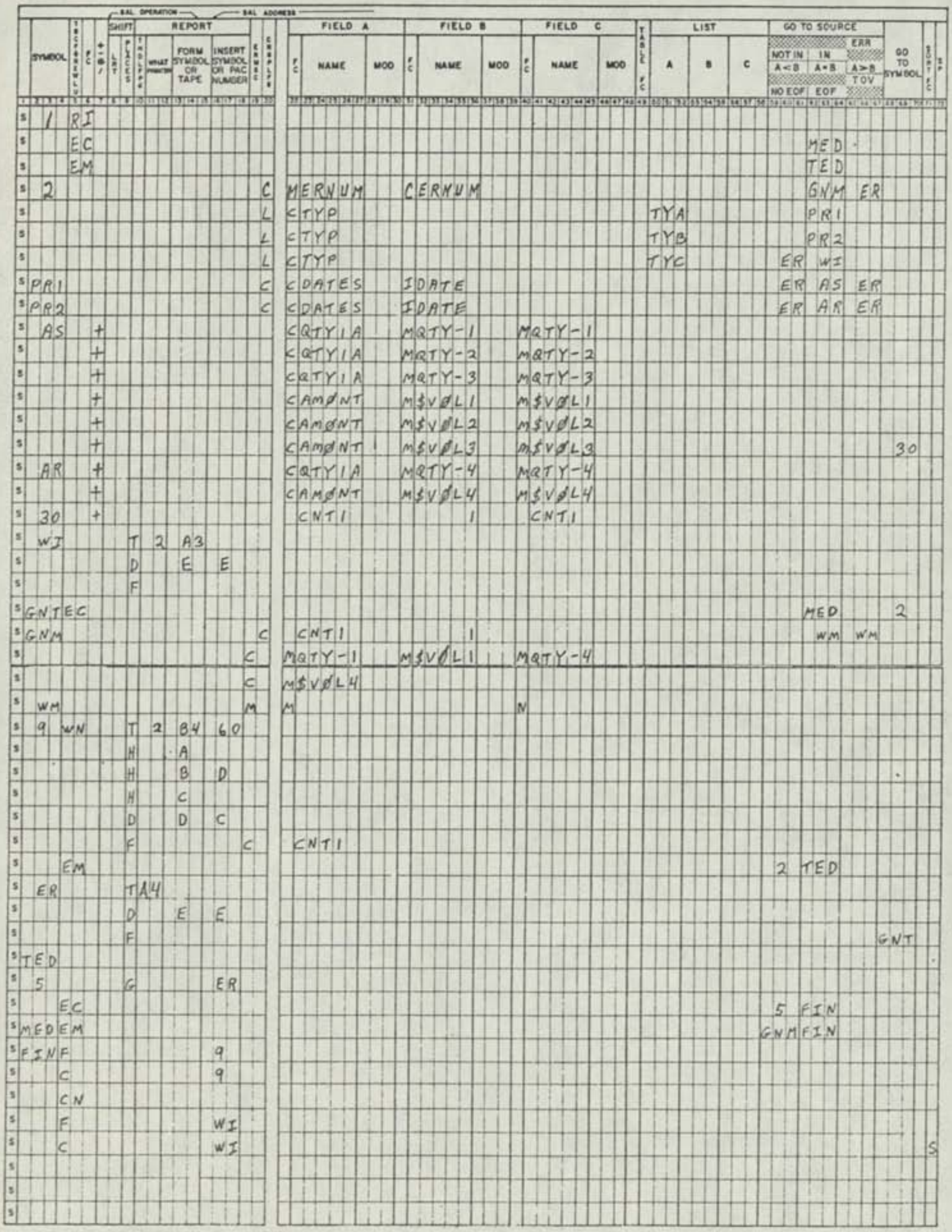
FIG. 1

on card I 'C'; after writing each part of the Report, Clear field 'CNT 1' to zero.

Processors should be primarily concerned with programmer and compile time efficiencies and then object time efficiency. The nonprogrammer readability of a POL lexicon should be of secondary consideration to the three efficiencies mentioned above; after all, who reads a source code—the programmer's boss's boss never will, and has no need to.

REFERENCES

1. Hughes, J. H. The trouble with commercial compilers. *Computers and Automation* (July 1961), 13–14.
2. Makinson, T. N. COBOL: a sample problem. *Comm. ACM* 4 (Aug. 1961), 340-346.



					SAL OPERATION				SAL ADDRESS																						
					SHIFT		REPORT						FIELD A			FIELD B			FIELD C				LIST			GO TO SOURCE					
	SYMBOL	[illegible]	FC	+ − * /	LRT	PLACES	[illegible]	WHAT PRINTER	FORM SYMBOL OR TAPE	INSERT SYMBOL OR PAC NUMBER	[illegible]	[illegible]	FC	NAME	MOD	FC	NAME	MOD	FC	NAME	MOD	TABLE FC	A	B	C	NOT IN A<B NO EOF	IN A=B EOF	ERR A>B TOV	GO TO SYMBOL	SORT FC	SP
s	1	R	I																												
s		E	C																								MED				
s		E	M																								TED				
s	2											C	M	ERNUM		C	ERNUM										GNM	ER			
s												L	C	TYP									TYA				PR1				
s												L	C	TYP									TYB				PR2				
s												L	C	TYP									TYC			ER	WI				
s	PR1											C	C	DATES		I	DATE									ER	AS	ER			
s	PR2											C	C	DATES		I	DATE									ER	AR	ER			
s	AS			+									C	QTYIA		M	QTY-1		M	QTY-1											
s				+									C	QTYIA		M	QTY-2		M	QTY-2											
s				+									C	QTYIA		M	QTY-3		M	QTY-3											
s				+									C	AMØNT		M	$VØL1		M	$VØL1											
s				+									C	AMØNT		M	$VØL2		M	$VØL2											
s				+									C	AMØNT		M	$VØL3		M	$VØL3									30		
s	AR			+									C	QTYIA		M	QTY-4		M	QTY-4											
s				+									C	AMØNT		M	$VØL4		M	$VØL4											
s	30			+										CNT1			1			CNT1											
s	WI						T	2	A3																						
s							D		E	E																					
s							F																								
s	GNT	E	C																								MED		2		
s	GNM											C		CNT1			1										WM	WM			
s											C		M	QTY-1		M	$VØL1		M	QTY-4											
s											C		M	$VØL4																	
s	WM										M		M						N												
s	9	W	N				T	2	B4	60																					
s							H		A																						
s							H		B	D																					
s							H		C																						
s							D		D	C																					
s							F				C			CNT1																	
s		E	M																							2	TED				
s	ER						T	A4																							
s							D		E	E																					
s							F																						GNT		
s	TED																														
s	5						G			ER																					
s		E	C																							5	FIN				
s	MED	E	M																							GNM	FIN				
s	FIN	F								9																					
s		C								9																					
s		C	N																												
s		F								WI																					
s		C								WI																					S
s																															
s																															
s																															

Fig. 2

GENERAL ELECTRIC

Information Systems
Equipment Division

DIAL COMM 8•433 452-2569 DATE 1970 June 22 MAIL ZONE M2

DEPT.•

ADDRESS•

SUBJECT• COPIES•

Bob Bemer - ASD

Looks great

6/29

TO: B. Stroup
T. A. Vanderslice

FROM: R. W. Bemer

Attached for clearance is a paper to be given at the DPMA Conference on 1970 June 26. It's very late; however, it is an adaptation and rewrite of two previously cleared presentations.

Thus I do not believe there will be anything controversial in the oral presentation. Publication will be delayed sufficiently so that changes are no problem.

RBemer

po

Computer
History
Museum

SEPARATION SHEET
Computer History Museum

Category of item(s):

□ Audiocassette	□ Printed materials
□ Electronic records	□ Three-dimensional objects
□ Oversized materials	□ Videotape
☒ Photographs/negatives	□ Other:

Number of items: 18 35mm slides (1 slidesheet)

Provide information about item(s):
Presentation slides found in 1970 part of folder. Slides are numbered 1-20, but slides 5 and 6 are missing.

Originally filed in:

Catalog #: 102785430 Lot #: X3054.2005

Collection Name: Robert (Bob) Bemer papers

Series: Papers Subseries:

Folder title: Speeches and papers, 1955-1982
Folder 3 of 5

Relocated to:

Catalog #: 102785615 Lot #: X3054.2005

Collection Name: Robert (Bob) Bemer papers

Series: Papers Subseries:

Folder title: Presentation slides, 1970-1978

Purpose of removal: To separate slides from text.

Date: 2/28/19

Name: Sydney Olson

OS-DOS

COEXISTENCE - ESC / CHAR FORMATS

! NEED NOT BE HOPELESS!

DEFINITION OF "100% TRANSFERABILITY"– MECHANICAL"

- ALL GE PRODUCT LINES (55, 100, 200, 400, 600) ARE CONSIDERED
- SMALLER MACHINE CAPABILITIES MUST BE SUBSETS OF LARGER MACHINES
- 100% COMPATIBILITY IN OPERATING SYSTEM COMMAND LANGUAGE — ENVIRONMENT
- 100% OF SOURCE LANGUAGE PROGRAMS MUST BE ACCEPTABLE TO COMPILERS OR CAN BE MECHANICALLY CONVERTED TO AN ACCEPTABLE FORM — PROCEDURE
- FILE FORMATS 100% MECHANICALLY CONVERTIBLE — DATA

BUT, PROCEDURE IS A STATEMENT OF A PROCESS. IF THE PROBLEM AND PROCESS ARE ADEQUATELY STATED ORIGINALLY (EITHER IN THE VERBIAGE OF THE PROCEDURE, OR IN AUXILIARY DOCUMENTATION), IT MAY WELL BE EASIER TO RESTATE THE PROCEDURE AND ACCOMMODATE THE DIFFERENT CHARACTERISTICS OF THE NEW SYSTEM.

MOTIVATION - RPT - 24/36

705/ PULAND 702 SLIGHTLY FASTER

DATA- IF CHAR EXCEEDS '5" THEN NOT SM ELEC STOR.
SNDS

REASONS FOR I. P. STANDARDS*

- DATA INTERCHANGE
- MULTIPLE USAGE OF DATA (BANKS)
- TRANSFER OF PROBLEM SOLUTION (PROGRAM AND DOCUMENTATION) TO:

  ADDITIONAL EQUIPMENT
  MULTIPLE "
  BACKUP "
  LINKED "
  NEW/DIFFERENT "
  BROKERAGE

- ECONOMY OF COMPETITIVE ACQUISITION (INTERFACES, MIXED SYSTEMS)
- AVOIDANCE OF REINVENTION
- FLEXIBLE CHANGE IN RESPONSE TO CHANGING REQUIREMENTS
- PERSONNEL TURNOVER AND TRAINING

* "AN ARBITRARY SOLUTION TO A RECURRING PROBLEM"

PUT IN TEXT

WHO'S WHO IN I.P. STANDARDS?

		COMPUTERS & INFO. PROCESSING	OFFICE MACHINES
INTL -	ISO	TC97	TC95
	IEC	x	x
	ECMA	x	x
	CCITT	x	
	ITU	x	
	IATA	x	
	IFIP	x	
NATL -	NBS		
	USASI	X3	X4
	AFNOR	x	x
	BSI	x	x
	⋮ ~100		
PROF -	ACM	x	
	IEEE	x	
BUS -	ABA	x	x
	EIA	x	
	ATA	x	
	NRMA	x	x

TRY TO GET FULL INTERNATIONAL AGREEMENT FROM THESE!

COMPATIBILITY PROPERTIES
REQUIRED OF DATA

- CODE & FIELD FORMAT (αNUMERIC CHARACTERS)
- FORMAT & SEMANTICS OF INTRINSIC DATA TYPES (INTEGER, FLOAT. PT., PACKED NUMERICS)
- VARIABLE-LENGTH DATA CONTROLS
- LOGICAL RECORD FORMAT & CONTROLS
- MESSAGE " "
- PHYSICAL BLOCK " "
- LOGICAL FILE " "
- FILE LABELS
- FILE CATALOG
- SPACE ALLOCATION TECHNIQUE
- VOLUME LABELS
- RECORDING TECHNOLOGY & CONVENTIONS
- PHYSICAL MEDIA

(A.L. ELLISON, 69 MAR 28)

ELEMENTS OF INDIRECT LEGIBILITY IN DATA

PRIMITIVES - ISO (ASCII) CODE

ELEMENTS - ENCODING OF ENTITIES
(PERSONS, LOCATIONS, ETC.)

- STRINGS } BINARY
NUMERALS } NUMERIC (PACKED)
FLOATING POINT
ETC.

STRUCTURES - LISTS, PLEXES

- RECORDS, FILES

- SELF-DESCRIPTIVE STRUCTURES

MUCH OF THIS VITAL STANDARDS
WORK IS NOT IN PROCESS!

NOTABLE CONTROVERSIES & FAILURES

- INCH - METRIC 90% OF WORLD. NEW U.S. STUDY. BRITISH OVER 10 YRS. HSEWIFE. QT-LTR TELEV/RADIO TEMP IN C°/F°
- CURRENCY AUSTRALIA, UK (5¢10)
- STEERING WHEELS SWEDEN

- CARD CODE (DECIMAL ASCII VS. HOLLERITH) IBM INTRODUCED W/O THOROUGH ✓. ECMA STD. HAD TO BACK DOWN
- ASCII FED. STD. VS. EBCDIC VS. RS244 (4 = EOT!)
- 7 VS. 8 BITS IN COMMUNICATIONS (HI-LOW) NOW FILLING ON PLANNED BASIS. GOVT. REGISTRY PRINCIPLES. 1 CHAR BUFFER, SHOW HOW TO BUILD CHEAP TERMINAL
- HANDWRITING CONVENTIONS 1,7 06-0)
- OCR-A VS. OCR-B (IT'S PRINT QUALITY!)
- KEYBOARDS (IBM BALL, 44 LIMIT. SPECIAL TRIP TO LEX. IN 64 "NEVER BE A TERMINAL" DEAD KEY VS. BSP SEQUENCE. BIT PAIRING, 3 KYBDS - MANUAL, OCR, CODED?
- SYSTEM INSTRUMENTATION (FORTRAN 2-INSTRUCTION EXAMPLE)
- PL/I

CHAOS RISES FASTER THAN NOW CONTROLLABLE BY STANDARDS!

WHO'S AT FAULT?

- USERS - DO NOT DEMAND (ASCII, INSTRUMENTING)
- MFRS - PUT OFF FOR PRESENT CONVERSION COSTS, IGNORING LIFE CYCLE COSTS

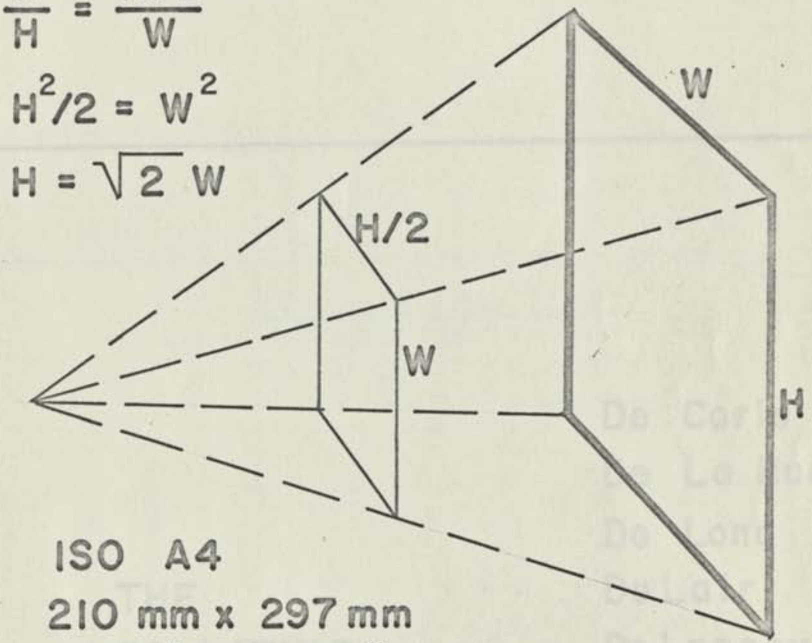
(MINE THE HI-GRADE ORE!)

3

$$\frac{W}{H} = \frac{H/2}{W}$$

$$H^2/2 = W^2$$

$$H = \sqrt{2}\,W$$



ISO A4
210 mm x 297 mm
8.27 " x 11.69 "

4

IF CHARACTER EXCEEDS
'S' THEN NEXT STATEMENT
OTHERWISE STOP

5

THE
COLLATING
SEQUENCE
PROBLEM →

De Carlo
De La Rue
De Lonq
DeLair
DeLancey
DeLaRue
Delancey
de Carlo
de la Rue
de Lancey

6

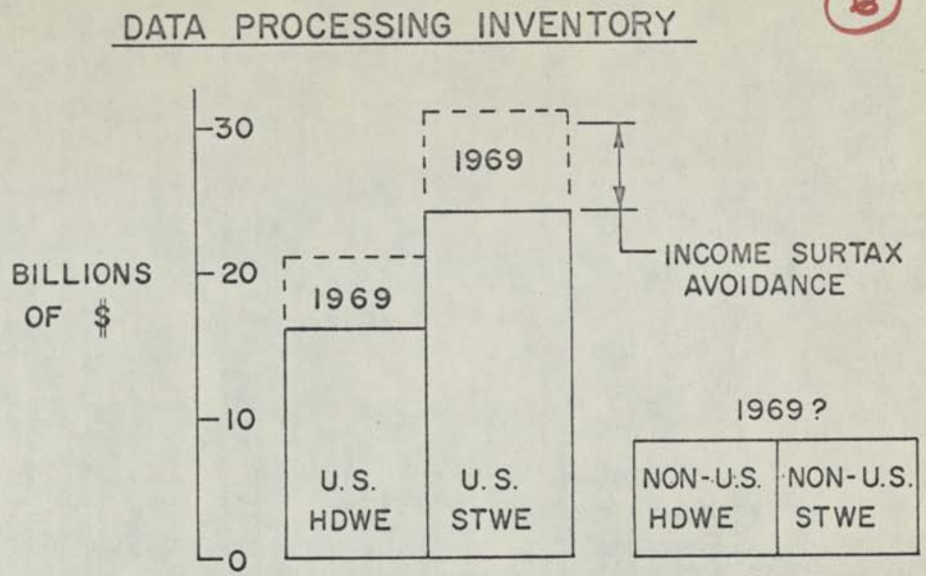


Figure 1.

7

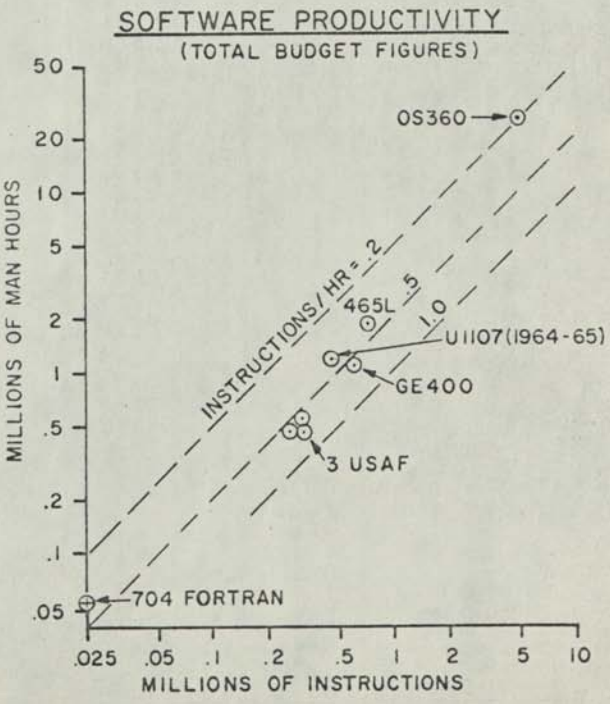
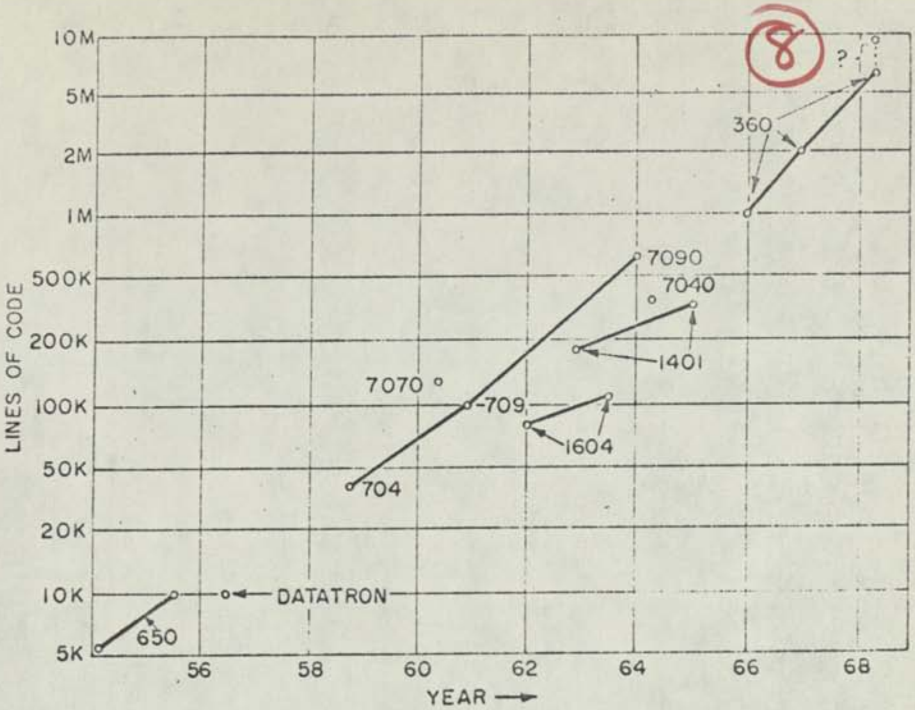


Figure 2.

8



GROWTH IN SOFTWARE REQUIREMENTS

Figure 3.

9

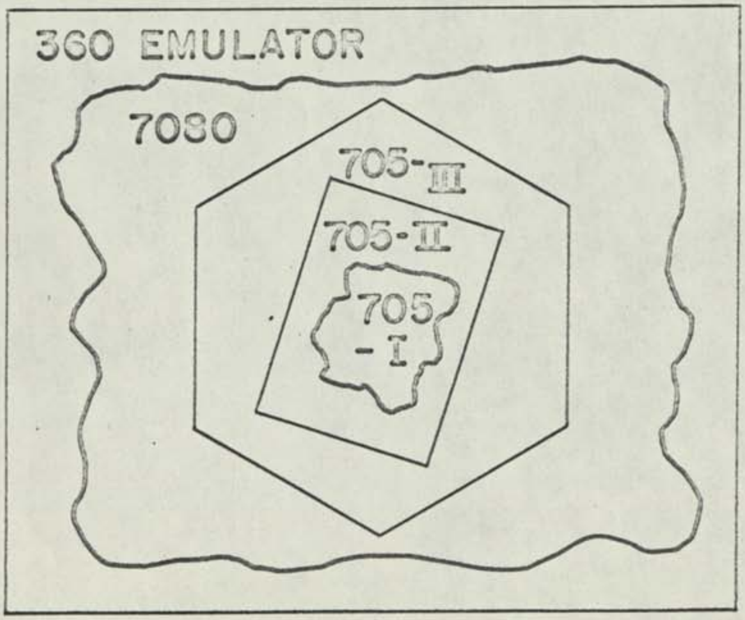


Figure 4. NO STANDARDS FOR TRANSFER

10

IF THE DATA ARE NOT TRANSFERABLE ~

THE PROGRAM CANNOT BE TRANSFERABLE

11

DATA:

A REPRESENTATION OF FACTS OR IDEAS IN A FORMALIZED MANNER CAPABLE OF BEING COMMUNICATED OR MANIPULATED BY SOME PROCESS.

12

INFORMATION:

THE MEANING THAT A HUMAN ASSIGNS TO DATA BY MEANS OF THE KNOWN CONVENTIONS USED IN ITS REPRESENTATION.

13

HOW TO RECOGNIZE DATA

IF YOU CAN:

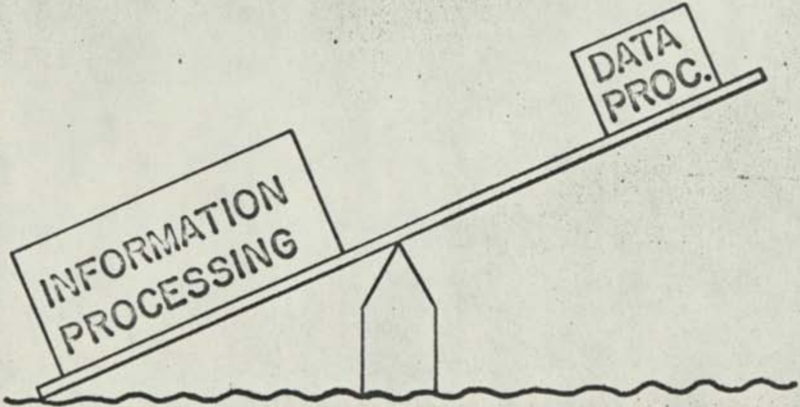
1) MOVE IT
2) PUT IT AWAY
3) FIND IT AGAIN
4) TRANSFORM & (UN) TRANSFORM

WITHOUT KNOWING WHAT IT MEANT -

IT'S DATA !

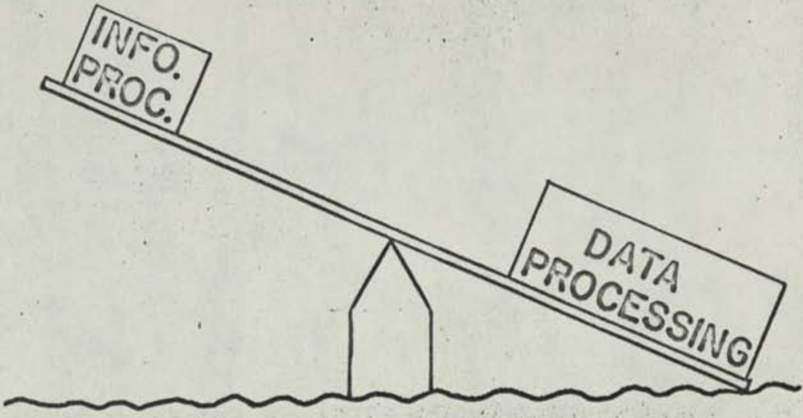
14

HOW IT WAS IN 196X



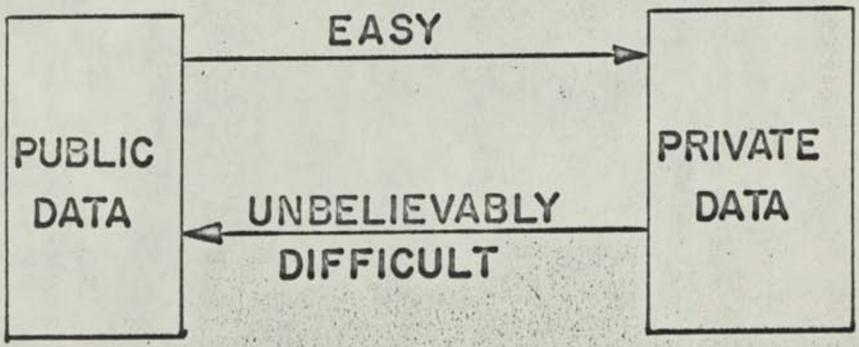
15

HOW IT WILL BE IN 197X



16

CHANGING THE RULES



PUBLIC DATA

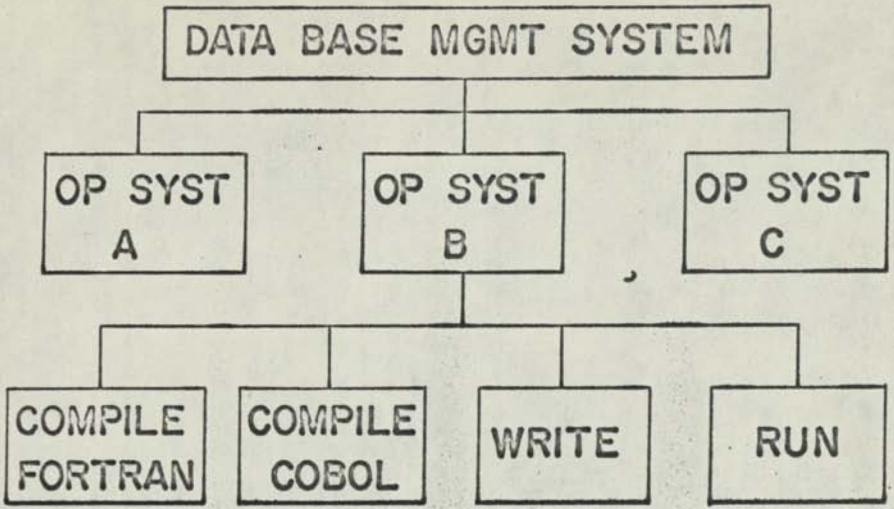
EASY

UNBELIEVABLY DIFFICULT

PRIVATE DATA

17

THE HIERARCHY OF SUBSERVIENCE



DATA BASE MGMT SYSTEM

OP SYST A

OP SYST B

OP SYST C

COMPILE FORTRAN

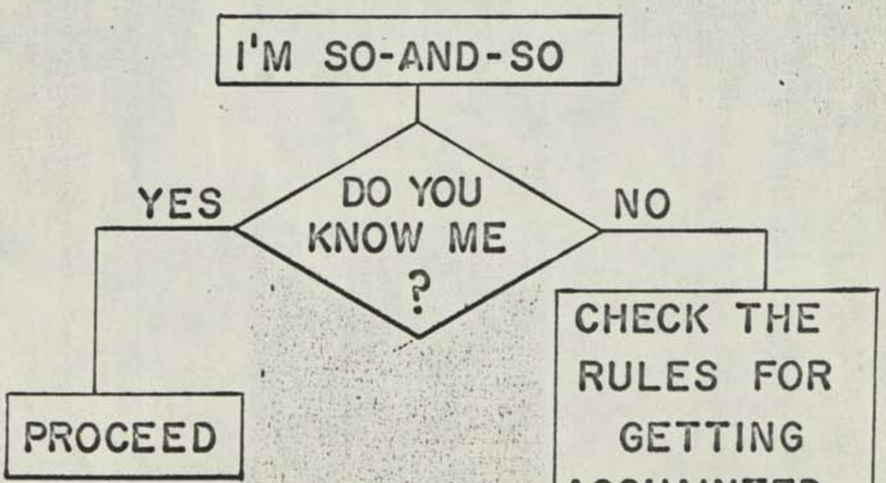
COMPILE COBOL

WRITE

RUN

18

THE RECOGNITION PROCESS



I'M SO-AND-SO

DO YOU KNOW ME ?

YES

NO

PROCEED

CHECK THE RULES FOR GETTING ACQUAINTED

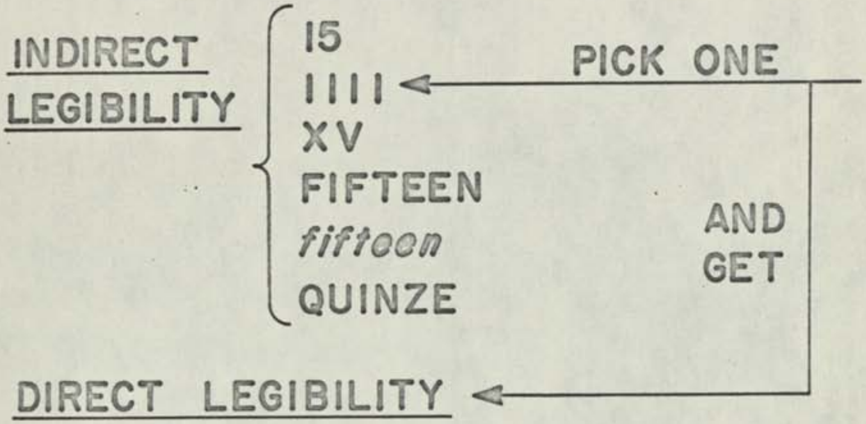
USAGE OF "ESCAPE"

TO CHANGE TO ANOTHER

1) CHARACTER SET
2) MEDIA LABEL
3) DATA FORMAT
4) DATA COMMUNICATION CONTROL PROCEDURE
5) ESCAPE DOMAIN

20

(SPARE)



INDIRECT LEGIBILITY

15

IIII

XV

FIFTEEN

fifteen

QUINZE

PICK ONE

AND GET

DIRECT LEGIBILITY

CONTROL ELECTRICAL SIGNAL LEVELS, TIMING, TRANSMISSION RATES, AND MEDIA REPRESENTATIONS - TO GET

PROCESSIBILITY

As they exist, governments have the main functions of:

- Embodiment of the state
- Decision-making

If there were no decisions to be made, the congress and the courts could go home, and the executive branch could reduce itself to the state department.

The decisions made by the government are primarily national in nature. They require national data. As we are constituted there is only one feasible way to get national data of sufficient completeness, accuracy, and homogeneity--the computer. This is a major goal for the proposed National Computer Year.

National data consists of the composite of local data, of which we suffer no lack. But the problem with local data is that it is essentially private data, hampered by information losses that prevent it from going public. As for fears of private data going public, protection is a matter of legislation and controls. The technical requirements are known or easily obtained, such that public data can be returned to privacy.

What really prevents data from going public is that there is no technical method for doing so (this is the contrary situation to making data private), even should the requirement be legislated.

Are not the mechanics of achieving this a very proper action for a standards body, even if we do not yet perceive a consensus in method? We have undertaken a development effort in programming languages, a field we understand somewhat, and for which there are some reasonably adequate standards already in usage. ASCII (ISO R646/GOST) is another example.

The country can survive without PL/I. It cannot survive perhaps without some method of focusing the attention of our decision-making bodies upon timely and sufficient data, in which the apples and the oranges have had their relative values quantized, and which comes from a~~n~~ system impersonal enough to counteract the effects of specialized bodies putting forth their grasping interests against the common and personal good.

And as we start data language and structuring development under the aegis of standardization, which is quite proper, let us go to the universities and make our apologies. Let us say that we overemphasized programming languages to such an extent that they are now busy turning out PhDs in syntax analysis that don't know anything about the structures and characteristics of the data upon which decisions affecting their lives are made.

4. If data conversion is required?

Several types of conversion may be required, such as:

- Graphic set content, encoding and character size
- Precision and range of numerals
- Data formats
- File content (added, changed or deleted)
- File structure
- Media labeling
- Physical media formats

Once-and-for-all conversion is the exception, and it may be advisa have it done on a service basis, particularly by an outside suppli commonly the new and old products must coexist until the new one : sufficiently. In this case it may be useful to have separate fil the old and the new product, with a bidirectional conversion prog verify identity between the two versions at each stage.

and standardizing to make user-developed software reusable and to reduce needless variety.

Is it for use within your own company, or can some generalization or modification in design enable it to be sold to others for the same purpose? Or can the algorithm be compartmentalized from the application so it may be used for different purposes by you and others?[2]

ARE DIFFERENT ANSWERS BENEFICIAL?

YOUR U.S. ARMY 7090 HAS RUN A FORTRAN OBJECT PROGRAM FOR THREE YEARS. UNIVAC COMPILES THE SAME SOURCE PROGRAM IN TRYING TO SELL YOU A 1107, AND A DIAGNOSTIC MESSAGE TELLS YOU THAT THERE IS AN ENTRY IN THE MIDDLE OF A DO LOOP, NOT REALIZED FOR THOSE THREE YEARS OF WRONG ANSWERS.

(TRUE CASE #1)

ARE DIFFERENT ANSWERS BENEFICIAL?

YOU HAVE INVERTED A LARGE MATRIX IN SHORT PRECISION ON YOUR 360. YOU MOVE THE PROGRAM TO A 48-BIT WORD MACHINE. YOU USED TO THINK THAT YOU HAD 5 DECIMAL DIGIT ACCURACY, AND YOU SUDDENLY REALIZE THAT IT IS ABOUT 1 DIGIT, AND YOU WERE MAKING DECISIONS BASED UPON AT LEAST 3.

(TRUE CASE #2)

ARE DIFFERENT ANSWERS BENEFICIAL?

YOU ARE UNIVAC, AND TRY A 7090 FORTRAN PROGRAM WHICH GIVES ANSWERS OVER THE FULL RANGE OF FLOATING POINT NUMBERS. YOU HIT A LOW BARRIER OF 10^{-22} AFTER WHICH THE ANSWERS ARE ALL ZERO. THE IBM PROGRAM RUNS TO 10^{-38}. THUS YOU REALIZE THERE IS A LOGIC FLAW IN THE CPU, AN INTERRUPT ON OVERFLOW IN THE LESS SIGNIFICANT PART OF THE PRODUCT, AND THERE ARE ALREADY SEVEN MACHINES IN THE FIELD.

(TRUE CASE #3)

COLLEGE
OF
ENGINEERING

UNIVERSITY OF FLORIDA

GAINESVILLE, FLORIDA 32601
AREA CODE 904 PHONE 392-0911

DEPARTMENT OF ELECTRICAL ENGINEERING

January 14, 1970

Mr. R. W. Bemer
E M I O Department
General Electric
13430 North Black Canyon Highway
Phoenix, Arizona 85029

Dear Mr. Bemer:

I would like to take this opportunity to thank you for your participation at the Third International Symposium on Computer and Information Science. From all the reports we have received, we may say that this international meeting was a memorable success. Your participation as an author has contributed significantly to the success of this Symposium.

We are making plans to organize the COINS-71. We look forward to your active participation at the next COINS Symposium.

Sincerely yours,

Julius T. Tou, Chairman
COINS-69 Symposium Committee

JTT:gs