R. W. Bemer

Speeches and Papers

1979 On

qannotator/field 07/27/82 14.252 MEDIA CODE 6

To: George Field, MS39-440
98*248-7238

1982 ~~Nov~~ Sep 27

From: Bob Bemer

cc: Steve Kirk

Subj: Papers

Here is a list of my papers for qualification for the honoraria authorized by the Technical Articles Program. Reprints or other establishing documents accompany this memo.

1. -, "Information acquisition, storage, and retrieval in the office environment", in State-of-the-Art Report 66, "Information Technology", Infotech International, Maidenhead, Berks., UK, 1979 Nov 26-28, 7/1-7/13.
2. -, "Some history of text processing", AFIPS Office Automation Conf., Atlanta, 1980 May 3-5.
3. -, "Office automation and invisible files", Automatizione e Instrumentazione ?, No. ?, 1980 ?, ???, Italy.
   (do not have publication for reprint yet, but know it is published. Sci. Honeyweller has asked for an abstract.)
4. -, Presentation, Keynote Panel, 1980 NCC Personal Computing Festival, 1980 May 19.
5. -, Presentation, 1980 NCC Pioneer Day Program (SHARE).
   (extensive summary published in Annals of the History of Computing, Vol 3, No. 1, 1981.)
6. -, "Incorrect data and social harm", DATA Magazine, Copenhagen, 10, No. 9, 1980 Sep, 43-46.
7. -, "File management for office automation", Infotech State-of-the-Art Conference, London, 1980 November 26-28, Vol 2, 15-26.
8. -, "New dimensions in text processing", Interface '81, Las Vegas, 1981 Apr 01-02.
9. -, "Problems and Solutions for Electronic Files in the Office", Proc. A.I.C.A. Annual Conf., Pavia, Italy, 1981 Sep 23-25, 131-133.
10. -, "Incorrect data and social harm", Computer Security J. I, No. 2, 51-56.

(In process, for the record)

11. -, Standards for Keyboards, Computers & Standards (accepted)
12. -, "Coordinated Text and Transparencies", submitted to Prac-Section, Commun. ACM.

FRI DEC 3 1030-1100

COMPUTER SECURITY JOURNAL
100 Tower Office Park • Woburn MA 01801
617/935-9200

Robert Bigelow, Editor

January 30, 1981

Mr. Robert W. Bemer
Honeywell, Inc.
2 Moon Mountain Trail
Phoenix, Arizona 85023

Dear Bob:

Many thanks for sending me your article from the September issue of Data. It's very good and I wonder whether I might reprint it, or perhaps a longer version if you have one, in the Computer Security Journal (assuming that it hasn't been published in the United States -- I could have missed it).

The Computer Security Journal is a semiannual periodical designed for the computer security professional. Articles cover the entire spectrum of problems faced by individuals concerned with computer security, from management to physical security, data security, communications security, computer crime and other relevant topics. The Journal's purpose is to furnish the computer security professional practical and innovative assistance in a readable fashion.

When you respond, would you send me another copy of page 46; the photocopier chopped off the top two or three lines of that page. Also, if you happen to have corrected the typos in a copy, I'd certainly appreciate that one to make sure I haven't missed any.

Sincerely,

ISSUE 2/3

Bob

Robert Bigelow

RB/ocsj

P.S. Did you reserve copyright, or should I clear with Eric? If so, could you send me his current address. The one I have is from quite some time ago.

A publication of the Computer Security Institute

E. Drake Lundell Jr.
Ronald A. Frank
Nancy French
Donald Leavitt
Frank Vaughan

Edith Holmes
Catherine Arnst
Jeffry Beeler
Marcia Blumenthal
Howard Karten
Timothy J. Scannell
Brad Schultz
Marguerite Zientara
Cheryl M. Gelb
Barbara T. VanScoyoc
Susan Gerrard
Joe Hunn
Ann Dooley
Denise Petski
Jim Orton

J. Daniel Couger
Roy N. Freed
Alan Taylor
Jack Stone

Roy Einreinhofer

Terry Williams
Mary Anne Furlong
Kathy Steinberg

Jack Edmonston
Kathryn V. Dinneen

Margaret Phelan
Barbara Jeannetti
Peggy Kelly
Eleanor Smith

Lee Vidmer
Henry Fling
Peter Holm
Donna Turnbull

espondence to the appro-
797 Washington Street,
. Phone: (617) 965-5800.

OFFICES: Washington,
Press Building, 529 14th
ton, D.C. 20045. Phone:

78 MAY 8

Blamed in Collapse of Civic Center Roof

By Howard A. Karten
CW Staff

HARTFORD, Conn. — The roof of this city's $31 million civic center collapsed under a load of wet snow last January because of basic errors in the original design, according to a computer analysis conducted by an academic committee investigating the incident.

This finding was part of a preliminary report of the Mayor's Academic Task Force, whose members blamed the problem in part on too much reliance on computers and not enough on common sense.

"Engineers tend to rely on computers too much," according to task force member Howard Epstein, an associate professor of civil engineering at the University of Connecticut at Storrs.

"There should be a lot of hand checking done in structures like this [space frame] because they're not the common ones. People tend to accept anything the computer tells them, which is a very dangerous practice," he said.

Like many engineers, the building's designers used a software package called Structural Design Language (Strudl) to assist in basic engineering calculations. The mayor's committee used the same package to investigate the design.

But, according to committee chairman Hayrettin Kardestuncer, also a professor of civil engineering at the university, "all the designers had to do was to look at the data submitted to the computer and they could see the data was not right."

They put garbage in and "they did get garbage out," he added.

Kardestuncer acknowledged that users occasionally find bugs in Strudl, which is updated annually by a user's group, but he said the bugs were not serious enough to have affected the calculations much.

Faults Itemized

The preliminary report, which will be followed by a more detailed version expected from the committee soon and another by a New York consulting firm hired by the city, made the following points:

- Both dead loads (the weight of materials used in construction) and live loads (e.g., snow) seem to have been underestimated.
- Buckling of the space frame's compression members appears to have been one of the major modes of failure. Bracing and the connection of interior members were inadequate.
- Struts used in adjusting the slope of the roof on the space frame appear to have contributed to the buckling of the structure.
- Building code provisions were not followed in the design of some of the tension members.

Kardestuncer said his group reached these conclusions after reanalyzing the original data with a system known as Finite Elements Analysis (Finel) in addition to Strudl. The runs were made on the university's IBM 370/155.

Another area of concern mentioned in the report was the need for a double check on engineering calculations.

At present, standard engineering and architectural practice is that a licensed professional engineer must certify the calculations; a provision for an independent check on such calculations is not a part of many construction codes.

Epstein said he plans to mention this need before the appropriate professional groups. Commenting on double checking by the city of Hartford, Kardestuncer said the additional cost of engaging an independent consultant would have been insignificant. "After all," he said, "this was not a coop that was going to house 10,000 chickens."

Babson Honors DEC's Olsen

WELLESLEY, Mass. — Babson College's newly created Academy of Distinguished Entrepreneurs honored five entrepreneurs including Kenneth H. Olsen, president of Digital Equipment Corp., as part of the college's Founder's Day ceremonies held here recently.

According to Babson's president, Ralph Z. Sorenson, the academy was established "to pay public honor to a representative group of entrepreneurs who, through their ability to create, take risks and build, have contributed significantly to the well-being of the economy and society."

Clarification

"Shop Floor Terminals Ease Production for Truck Maker" [CW, April 24] was not meant to give the impression that terminals made by IBM and Panasonic Corp. were incompatible with the Xerox Sigma 7 mainframe.

by Ann Dooley
CW Staff

TOWE, Vt. — ice here stopped arrested a driver and four passen- s recently after a check of the Na- al Crime Information Center CIC) system indicated the car was en. That information, however, ned out to be inaccurate and out of e.

he occupants of the Mercedes Benz e members of an all-black band ed "Hawkeye" who were enroute to usical engagement when they were pped.

STULL 129

ate police said they spotted the rcedes on a major interstate high- and ran a "routine check." Upon ning the car was stolen, they in- ned local authorities who reportedly stopped the group with "guns drawn."

Second Check

When Robert Sumler, owner of the car, produced his license and registration, Stowe police made a second check with the police in Providence, R.I., where the car was registered. The second check showed the car had been stolen last summer in Providence but was recovered in Boston some time later and returned to Sumler.

But Providence police apparently never purged the registration number and theft report from the NCIC file when the car was recovered, Vermont state police claimed.

"The system is no better than the people feeding information into it," according to Lt. Edward Fish, head of that state's police unit.

Providence police, however, claimed that since the car was recovered out of state, they had no way of knowing it had been returned to its owner. Boston police never informed the Providence department of the vehicle's recovery, and it's difficult to tell what happened since Boston has so many stolen car reports that it doesn't keep records very long, a Providence police spokesman claimed.

Hawkeye members could not be reached for comment.

'Number of Reasons'

All of the law enforcement agencies agreed there could have been any number of reasons why the theft report didn't get purged from the tem. The large ume of paperw and the lack of communication am all the agencies makes these incid possible, one officer said.

In fact, a Stowe police officer m tioned he had recently noticed a v cle on the stolen vehicle list that been returned to its owner. The off noticed the error only because St has only "about half a dozen" st vehicles to record, he said.

Vans With Min
Aid Debris Hu

(Continued from Page 11)

two such drives from Data Gen and Cipher Data Products, Inc. to r them, Story said.

"A recently-designed micropro sor-based data acquisition system n mally writes on DC300A-type c tridges, [so] two drives were provi in one of the vans to handle those c tridges," he explained.

Results Plotted

The results of the data reduct were presented as plots from T tronix, Inc. 4010 interactive grap terminals with Tektronix 4610 ha copy units and from California Co puter Products, Inc. 30-in. Model and 10-in. Model 565 drum plotter each van, Story noted.

"A variety of in-house Assembly l guage programs could be loaded fr the 9-track drives, from the dual D General 'Philips' cassette decks from single 2.5M-byte [Diablo S tems, Inc.] disks.

"Batching of jobs was achieved vi Data General high-speed paper t reader, and paper tape punch, and

78 APR 10

1980 NCC PIONEER DAY PROGRAM

21 May 1980   1:30-4:45 p.m.

Anaheim Convention Center--Santa Ana II

Co-chairmen--Mort Bernstein, Hank Tropp

1:30 INTRODUCTIONS

Topic: The Computing Environment of the '50s

1:40 "Piercing the Unknown" (5 min. version), an IBM film

1:50 The Vendor's Environment

Panelists: John Backus, IBM
John Greenstadt, IBM
Cuthbert Hurd

2:25 The User's Environment

Panelists: Lee Amaya, SIAC
Bob Bemer, Honeywell
Ed Jacks, GM

3:00 BREAK

3:15 SHARE: Getting Started

Panelists: Paul Armer, CBI
Jack Strong
Frank Wagner, Informatics

3:45 SHARE: The Technical Effort

Panelists: Irwin Greenwald, Honeywell
Owen Mock, CSC
Roy Nutt, CSC

4:15 The View from Today

Panelists: Frank Wagner, Informatics
Jerry Feinman, IBM

1980 OFFICE AUTOMATION CONFERENCE

1980 Office Automation Conference Digest

March 3-5, 1980
Atlanta, Georgia

Conference Chairman: Floyd O. Harris
Program Chairman: Amy D. Wohl
Digest Editor: Ira W. Cotton

afips PRESS

Sponsored by
American Federation
of Information
Processing Societies

The ideas and opinions expressed herein are solely those of the authors and are not necessarily representative of or endorsed by the American Federation of Information Processing Societies.

Library of Congress Catalog no. 80-80155

Printed in the United States of America

Session 34

Text Editing Software

Chairman: Robert Bemer
Honeywell

A tutorial on the design and development of text editing software. The speaker will place some emphasis on the tradeoffs between different types of software.

1979

COMPUTER-ASSISTED LEARNING FOR THE METRIC SYSTEM

R. W. Bemer

HIS-LISD, Phoenix, AZ

ABSTRACT

A program for Honeywell L66 computers provides all relationships and conversions in and between the SI (new metric system) and the Old English system of measure. The TEX language was used to provide ease-of-use and the capability to provide some response for any input, an important requirement for computer-assisted learning.

Keywords - metric-SI-customary-English-measure-conversion-TEX-computer-assisted-self-learning

Why SI?

The United States has had two measurement systems for the last 100 years -- Old English (customarily called "customary") and metric. Historically we have, as a nation, looked upon the metric system as useful only to scientists, and to people in the weights and measures business, who need it as a rational way (and the <u>only</u> legal way) to define the Old English units!

But both the metric system and the world have changed. The first has been transformed and redefined into the SI (Système International d'Unitès) (International System of Units), and the world has changed to its general use (except for the United States, in any legally compulsory way). The laws of most nations demand its exclusive use; more importantly to us, selling goods abroad demand it. Thanks to the oil problem, more U. S. citizens now understand what a "balance of payments" problem is, and how a "negative balance" erodes the value of our dollars. It's no accident that the switchover to the SI is led by our major exporters -- General Motors, Caterpillar Tractor, Rockwell International, IBM, and Honeywell!

A Matter of Conversion

Those who have made good adjustment to using the SI (the new metric system) are agreed that it's done by thinking metric, not by continual back and forth conversion, either mental or using your hand calculator. For example, the relation between Fahrenheit and Celsius temperature scales is just too complicated. As usual, we fare well by thinking decimal, or dollars. Water boils at a dollar (100 oC), freezes at lack of a dollar (0 oC). We're physically comfortable at 18 oC, even though our standard body temperature is 37 oC.

By knowing these four points of reference we get a good feel for other points on the scale. Living in Phoenix, I know a fifth point -- it gets to 46 oC in the summer, without fail.

Let's not be turned off by those who may say that the SI is a foreign, or even unAmerican, system -- just because the French were the leaders in defining and using the SI. Don't forget that your everyday alphabet is Roman, and your everyday numbers are Arabic.

A Computer can Convert

Even though total immersion and thinking in the SI is best, there are still two cases where conversion is useful -- 1) for places where human thinking can't do it, as in a computerized database, and 2) when your head doesn't convert, but only sees the results when a computer converts.

For these reasons I've constructed a computer program that coexists between our two measurement systems. Other conversion programs have been written, but this (to my knowledge) is the only one in existence with all of these properties:

1. It works within either system, or between systems in either direction.

2. It is interactive (one uses a terminal connected to a computer).

3. It is both conversational and tutorial. Invalid requests are detected and handled.

4. Variation in human input is accommodated, as are abbreviations, misspellings, and symbols.

5. It handles all measurement units in both systems without altering the program when another unit is added to the repertoire.

Before we look at each of these aspects in detail, remember that the purpose of doing so is not to gain insight into the SI. One does that by using the program itself. In this paper, I use the SI program equally to show good programming practice for those hot buttons of today -- ease-of-use, friendly interface, flexible design, minimum maintenance, and avoidance of software traps.

Figure 1 shows the entire SI program except for the subprogram that converts spelled-out constants to digits. Figure 2 shows the unit terms (and variations) that the program recognizes.

In, and Between, Either System

Oddly enough, because the Old English system is so much more complex than the SI, most personal usage of this program turns out to be within it! E.g.:

sections/township	(there are 36)
barleycorns per foot	(there are 36)
teaspoons per gallon	(there are 768)
chains per furlong	(there are 10)
fathoms/league	(there are 2640)
leas per 8 hanks	(there are 56)

You might not, in particular, know the last relationship unless you are in the textile industry, where a hank of cotton yarn is 840 yards long, and a lea is 120 yards. However, a hank of worsted yarn is not 840 yards, nor is a hank of linen yarn, and they differ in length. In the SI it's all in metres.

It's equally useful to find relationships entirely within the SI, such as:

sq metres/hectare	(there are 10000)
cubic metres per stere	(there is 1)

Then we may wish to go back and forth. Particularly in computerized databases, for a long period of changeover. Some examples:

10 pints per 8 hrs	(0.164 millilitres per second, quantity of flow, or consumption)
10 pints	(4.732 litres, quantity of liquid volume)
hectares/40 acres	(there are 16.188)
quarts per litre	(there are 1.057)
grams/lb.	(there are 453.592)

```
!si _ 1979-03-26 author:RWBemer, 602-942-1360
 _ Copyright 1979, Honeywell Information Systems, Phoenix, AZ
 filename="texlib/l/wordfile,q" erca !first,b=b
 b if *cl']9:nes:"|wordfile" call texlib/f/old

!again subs $ case
 narg=1 nn="00" npower=-13 nconst=10000000000000 darg=1 nd="00"
 out:b out:b in:"  Term is?  " input=*in if *lin:eq:0 $*svmd$ return
 scan:input:" per " if *lm:eq:0 scan:input:"/"
   *lm:ne:0 darg=*r>'"b
   g=*l'<"b if narg:nes:"1" if narg:nes:"one" call !numer $retry$
 if darg:eqs:"1" power=npower const=nconst goto !unity
 call !denom $retry$
 if du:eqs:"g" if nu:nes:"g" dpower=dpower+3 du="kg"
 power=npower-dpower const=nconst/dconst
 if nn:eqs:nd call !ratio goto !again

!unity b f:(nn,"/",nd) if *eof out:" - Has no meaning in the SI" goto !again
 scan:*cl:"|" use=*r k=(setk'](*l']2+1))['1 if *l']5:eqs:"26/25" k=1
 lc=rnnst>*null posit=power+lc+18*k div=3
 if L.ce:1 div=3*k if posit:ge:(15*k) if posit:le:(21*k) div=k
 q=posit/div*div rem=*rmdr
 if q:gt:(36*k) $exceed$
 if rem:lt:0  $exceed$
 sym=(power$k$]'posit)']1 if sym:eqs:b sym=*null
 if sym:eqs:"?" sym="da"
 split:(const+(50000000000)'](lc-3-rem)):rem call !adju_con
 pre=const,b if pre:eqs:"1 " pre=*null
 scan:use:"|" out:"  output =  ",pre,sym,*l scan:*r:"|"
 b if *l:nes:*null out:"quantity of ",*r
goto !again

!numer arg=narg s=94 k=1 can="yes" power=0 words=1 call !sqcu
 call !constant $cant$
 call !get_it $cant$
 nn=numb nu=u lnc=const>*null
 npower=power+k*del1+del2+lnc-precis nconst=(const,zero)']precis return

!denom arg=darg s=94 k=1 can="yes" power=0 words=1 call !sqcu
 call !constant $cant$
 call !get_it $cant$
 nd=numb du=u dpower=power+k*del1+del2
 dconst=const return

!get_it i=*random/2 larg=arg>*null
 if larg:gt:1 if legit>(".",arg,"."):gt:12 call !prune
 if larg:ge:5 call !meter
 call !find $no$ out:\"\,arg,m$i$ return
 call !adjust const=const*mult return

!find ps="no" can="yes" s=94 del1=0 if larg:eq:2 nocase call !poss_ambig
   ps:eqs:"yes" goto !pp
   e b f:(arg,"|") if "*eof return
 if larg:eq:1 can="no" return
 if *eof if words:eq:2 scan:arg:b b f:(*l,*r,"|") if "*eof return
!pp call !poss_pref $cant$
 clue=arg,"|" b f:clue if *eof can="no" return
 del1=f$s$ return

!adjust numb=*cl<'"|" if numb:eqs:"f;1" f;1 goto !adjust
 if numb:eqs:"01" numb=(length'](2*k))['2
 if numb:eqs:"03" numb=(time'](2*k))['2
 scan:*cl:"[" if *lr:ne:0 call !bump return
 scan:*cl:"|" scan:*r:"|" u=*l mult=1 del2=0
 if numb:eqs:"25" k=2
 if numb:eqs:"26" if u:nes:"l" k=3
 return

!bump scan:*r:"]" u=*r'>"|" scan:*l:"p" mult=*l del2=*r*k
 if k:eq:2 mult=mult*mult
 if k:eq:3 mult=mult*mult*mult
 if mult>*null:gt:7 split:mult:7 mult=*l del2=del2+*lr
 return

!sqcu scan:arg:"sq " if *lm:eq:0 scan:arg:"square "
 if *lm:eq:0 scan:arg:"cu " if *lm:eq:0 scan:arg:"cubic "
 if *lm:eq:0 return
 if *m']2:eqs:"sq" k=2
 if *m']2:eqs:"cu" k=3
 arg=*l,*r return

!poss_pref dotarg=".",arg case
 if larg:gt:5 split:dotarg:6 s=prefixes>*l
 if larg:gt:4 if s:ge:94 split:dotarg:5 s=prefixes>*l
 if larg:gt:3 if s:ge:94 split:dotarg:4 s=prefixes>*l
 if larg:gt:2 if s:ge:94 split:dotarg:3 s=prefixes>*l
 if larg:gt:1 if s:ge:94 split:dotarg:2 s=prefixes>*l
 if s:ge:94 can="no" return
 nocase if *l:eqs:".m" s=16
 if *l:eqs:".P" s=53
 case
 arg=*r if arg:eqs:"hm" arg="ohm"
return

   ne if arg['2:eqs:".s" return
 if arg['1:eqs:"." arg=arg'[1 larg=larg-1 return
 scan:arg:".*" if *lm:ne:0 arg=*l,"*",*r larg=larg-1
 scan:arg:". " if *lm:ne:0 arg=*l,b,*r larg=larg-1
 if larg:eq:1 return
 if arg['1:eqs:"s" if larg:gt:2 arg=arg'[1 larg=larg-1
return

!meter scan:arg:"meter" if *lm:eq:0 goto !liter
 arg=*l,"metre",*r out:"A 'meter' is an instrument."
 out:"The unit of length is the 'metre'." return

!liter scan:arg:"liter" if *lm:eq:0 return
 arg=*l,"litre",*r out:"'litre', if you please!" return

!constant t=arg lt=arg>*null if lt:ge:2 t=t'[1
 scanr:t:*n t=*ll if t:ne:0 call !numeric return
 if t:eq:0 call !words return

!words scanr:arg:b if *lm:eq:0 const=1 power=0 return
 rhold=*r count=tails>(".",*l['2,".")
 if count:eq:9 if *l['5:nes:"eight" call !twowords
 if count:eq:21 if *l['5:nes:"llion" call !twowords
 if count:gt:33 call !twowords
 if *l:eqs:*null return
 count=tails>(".",*l['2,".") if count:gt:33 can="no" return
 arg=rhold var=*l call texlib/l/spell const=numvar return

!twowords scanr:*l:b if *lm:eq:0 const=1 power=0
 rhold=*r,b,rhold words=2 return

!numeric const=*l,*m scan:arg:const arg=*r>'"b
!cloop scan:const:"," const=*l,*r if *lm:ne:0 goto !cloop
!bloop scan:const:b const=*l,*r if *lm:ne:0 goto !bloop
 scan:const:"." const=1*(*l,*r) power=0-*lr return

!ratio if nu:nes:du call !scale
 power=sd+power const=(1,zero']sd)/const diff=power-(const>*null)
 if diff:gt:0 call !small return
 splitr:(const+(5,zero']sd)'](power-3)):power
 if *ll:gt:3 out:"            = ",*l return
 call !adju_con out:"            = ",const return

!small one=("1",zero)'](power+1) const=one+const
 splitr:(const+(5,zero']sd)'](power-3-diff)):power
 out:("            = 0.",*r'](diff+3))'<"0" return

!scale if nu:eqs:"m**3" if du:eqs:"l" power=power+3 return
 if nu:eqs:"l" if du:eqs:"m**3" power=power-3
 return

!adju_con const=*l,".",*r']3
 if *ll:eq:0 const="0",const
 const=const'<"0" if const['1:eqs:"." const=const'[1
 return

!poss_ambig x=amb>(".",arg,".") if x:gt:62 return
 if x:gt:44 ps="yes" return
 if x:le:3  S1="ft" S2="fT" T1="feet"      T2="femtotesla"  goto !ask
 if x:le:9  S1="g"  S2="Gm" T1="gram"      T2="gigametre"   goto !ask
 if x:le:18 S1="cubic cm" S2="cC" T1="cm**3" T2="centicoulomb" goto !ask
 if x:le:23 S1="t"  S2="T"  T1="tonne"     T2="tesla"       goto !ask
 if x:le:28 S1="Pa" S2="pA" T1="pascal"    T2="picoampere"  goto !ask
 if x:le:37 S1="ha" S2="hA" T1="hectare"   T2="hectoampere" goto !ask
 if x:le:43 S1="pH" S2="ph" T1="picohenry" T2="phot"
!ask out:\Enter "\,S1,\" if you mean "\,T1,\"\
 in:\      "\,S2,\" if you mean "\,T2,\"  ? \
 if *in:eqs:S1 arg=T1 larg=arg>*null
 if *in:eqs:S2 arg=T2 larg=arg>*null
 x=amb>(".",*in,".") if x:gt:62 return
 if x:gt:44 ps="yes" return
 goto !ask

!first b=" " precis=14 sd=30 f0="6" f5="3" f10="-2" f16="-3" f22="-6"
 f28="-9" f33="-12" f38="-15" f44="-18" f49="18" f53="15" f58="12"
 f63="9" f68="2" f74="-1" f79="1" f82="1" f87="1" f92="-6"
 legit=".das.tbs.mph." length="012526" time="0327"
 retry=/ out:"Retry" goto !again/ exceed=/out:"Wow! You've exceeded SI limits!"/
 exceed=exceed,retry no=/if can:eqs:"no"/ retry=no,retry cant=no," return"
 prefixes=".mega.kilo.centi.milli.micro.nano.pico.femto.atto.exa."
 prefixes=prefixes,"peta.tera.giga.hecto.deci.da.deka.deca.u"
 power1="aaafffpppnnnuuummm   kkkMMMGGGTTTPPPEEE"
 power2="aaaaaafffffppppppnnnnnnuuuuuummccdd  ??hhkkkkkk"
 power2=power2,"MMMMMMGGGGGGTTTTTTPPPPPPEEEEEE"
 power3="aaaaaaaaafffffffffpppppppppnnnnnnnnnuuuuuuuuummmcccddd  "
 power3=power3," ???hhhkkkkkkkkkMMMMMMMMMGGGGGGGGGTTTTTTTTTPPPPPPPPPEEE"
 setk="111111111111111111111112321111111111111"
 zero="00000000000000000000000000000000000000000000000"
 amb=".FT.Ft.gm.GM.CC.cc.Cc.t.T.pa.PA.ha.HA.Ha.PH.Ph.fT.Gm.cC.pA.hA.pH."
 big="######milbiltriquaquisexsepoctnondec" trans="##ontwthfofisiseeiniteel"
 tails=".ed.ee.en.ht.ix.nd.ne.on.ty.ur.ve.wo." u=*null
 m0=\" isn't in our vocabulary\ m1=\" not in the repertoire\
 m2=\" stumps me\ m3=\" is a problem\ m4=\" isn't findable\
 call !explain return

!explain out:" "
 out:"TEXLIB/L/SI converts just about anything to the SI,"
 out:"or International System of Units (metric system)."
 out:"It's useful for teaching, and extracts may be used"
 out:"for conversion subroutines in production programs."
 out:" " out:"Examples:  Term is? furlongs per fortnight"
 out:"             Term is? teaspoons/gallon"
 out:"             Term is? tesla"
 out:"             Term is? 55 miles per hour"
!end_explain return
```

```
"  (inch)
#  (pound)
'  (foot)
A
A.s
A.s.m
BTU
Bq
C
C.m
Ci
F
Gy
H
HP
Hz
J
K
Mx
N
N.m
P
Pa
Pa.s
Pa.sec
R
S
Sv
T
V
V.s
[illegible]
W.h
W.s
Wb
Wh
abampere
abcoulomb
abfarad
abhenry
abmho
abohm
abvolt
acre
acre feet
acre foot
acre ft
acre in
acre inch
amp
ampere
ampere hour
ampere second
ampere second metre
ampere-second
angstrom
are
astronomical unit
atmosphere
atto
b
bar
barleycorn
barn
barrel
basket
bd ft
becquerel
biot
board ft
boardfeet
boardfoot
bolt
bushel
butt
cc
c.c.
cab
caliber
calorie
candela
candela steradian
candela-steradian
candle
candlepower
carat
cd
cd.sr
cental
centi
century
chain
chaldron
clove
cord
coulomb
coulomb metre
cubit
cup
curie
cut
cycle
day
deca
decade
deci
deka
denier
drop
dyn
dyne
electronvolt
ell
erg
exa
fall
farad
faraday
fathom
feet
femto
fermi
fifth
firkin
fl ounce
fl oz
fluid dram
fluid oz
fluidounce
fluidrachm
fluidram
foot
footcandle
footlambert
fortnight
franklin
ft
furlong
g
gal
gallon
gamma
gauss
giga
gilbert
gill
gm
gr
grade
grain
gram
gramme
gray
ha
hand
hank
hectare
hecto
heer
henries
henry
hertz
hogshead
horsepower
hour
hr
hundredweight
in
in**2
inch
inches
iron
jeroboam
joule
kW.h
kWh
kayser
kelvin
kg.K
kg.m
kilderkin
kilo
kilogram kelvin
kilogram metre
kilogram-kelvin
kilogram-metre
kip
knot
l
lambert
langley
last
lb
lea
league
lightyear
line
link
litre
lm
lumen
lux
lx
m
m**2
m**2.sr
m**3
m.K
magnum
marathon
maxwell
mega
methuselah
metre
metre kelvin
metre newton
metre-kelvin
metre-newton
metric ton
mho
mi
micro
micron
mil
mile
milli
millibar
min
minim
minute
mol
mol.K
mole
mole kelvin
mole-kelvin
nail
nano
naut mi
nautical mile
nebuchadnezzar
newton
newton metre
newton-metre
nmi
noggin
oersted
ohm
ounce
oz
pace
palm
parsec
pascal
pascal second
pascal-second
peck
pennyweight
perch
peta
ph
phot
pica
pico
pint
point
poise
pole
pond
pottle
pound
poundal
psi
pt
puncheon
qt
quart
quarter
quintal
rad
radian
rd
rem
rhe
rod
roentgen
roll
s
s**2
sack
seam
seamile
sec
sec**2
second
second**2
seconds**2
section
shake
shot
siemens
sievert
skein
slug
span
spindle
sr
standard
statampere
statcoulomb
statfarad
stathenry
statmho
statohm
statvolt
steradian
stere
stilb
stokes
stone
strike
survey feet
survey foot
survey ft
t
tablespoon
tbs
teaspoon
tenthmetre
tera
tesla
tex
therm
thread
ton
tonne
torr
township
tsp
tun
unit pole
volt
volt second
volt-second
watt
watt-hour
watthour
wave number
weber
week
wey
wineglass
wk
yard
yd
year
yr
```

Interactive Characteristics

The program itself is a file named TEXLIB/L/SI, where:

TEXLIB	is a library of service programs, usually written in the TEX language, organized for retrieval by keywords.
L	is the "LEARN" subdivision of TEXLIB.
SI	is the program itself.

In practice, one dials up to connect a terminal to a Honeywell Level 66 computer (the only computer that understands the TEX programming language [1,2,3,4] at this writing). After invoking the TEX system, the program TEXLIB/L/SI is called, with these results:

```
-call texlib/l/si

TEXLIB/L/SI converts just about anything to the SI,
or International System of Units  (metric system).
It's useful for teaching, and extracts may be used
for conversion subroutines in production programs.

Examples:  Term is? furlongs per fortnight
           Term is? teaspoons/gallon
           Term is? tesla
           Term is? 55 miles per hour

Want to see the permissible terms? (80 lines) no

Term is?
```

Tutorial and Conversational Aspects

The user may pace the learning process, for no other person is involved. The user may give any reply to the question:

```
Term is?
```

The program then takes some action for every input, each action being in one of these classes:

- Input correct

-- A response is output; a conversion, a ratio, or a definition (as for a prefix or unit).

```
  Term is?  lightyears per year
  output =  0.3 Gm/s
quantity of velocity

  Term is?  firkins per hogshead
          = 5.829

  Term is?  peta
peta means 10 to the power 15

  Term is?  tesla
quantity of magnetic flux density
```

In the first example we note that the metric speed of light is 0.3 gigametres, or 300 megametres, per second, quite exactly. An easy figure to remember.

- Unit misspelled

-- The most common responses are:

```
A "meter" is an instrument.
The unit of length is the "metre"

"Litre", if you please!
```

- Unit is ambiguous

-- The SI is a precise system. Its units never vary. When imprecise Old English terms are given, they are recognized, and a message indicates why they're not convertible. For example, the wine industry has no standard for champagne bottles, or fills. The magnum, jeroboam, methuselah, and nebuchadnezzar vary with the individual producer.

```
  Term is?  calorie
No data on "calorie"
Reason = (there are many)
Retry
```

The same is true for British Thermal Units. There are six different values, see [5], pp. 24-25, the most authoritative US document on the SI and conversion. And for basket, pinch of salt, and many others.

- Result out of range

-- Although the SI can handle quantities from the near infinitesimal to the superastronomical, it's still possible to fall outside the permissible range.

```
  Term is?  gigacycles per picosecond
Wow! You've exceeded SI limits!
Retry
```

- Compound unit makes no physical sense

```
  Term is?  volt per second
 - Has no meaning in the SI

  Term is?  kilohms per volt
 - Has no meaning in the SI
```

- Unit isn't known to the program

-- This is not usually true, as the driving table contains some 350 entries, of which many are variants. However, if the user miskeys, or enters some term not in the table, a random digit is generated and used to subscript the selection of a message of condolence.

```
  Term is?  2 biots
No data on "biot"
Reason = (cannot locate conversion factors)
Retry

  Term is? motherinlaw
"otherinlaw" isn't in our vocabulary
Retry
```

The latter appears rather stupid at first, until you realize how the program must work. First it checks for the full prefix from among:

10 to +i		i	10 to -i	
exa	(E)	18	atto	(a)
peta	(P)	15	femto	(f)
tera	(T)	12	pico	(p)
giga	(G)	9	nano	(n)
mega	(M)	6	micro	(μ)
kilo	(k)	3	milli	(m)
hecto	(h)	2	centi	(c)
deka	(da)	1	deci	(d)

In this case, failing to find a full prefix, it attempts to find a prefix symbol. It found one, "m" for "milli", which was then removed to find a unit of measure. "otherinlaw" wasn't in the table, with the above result.

- Interpretation is case-dependent

-- The SI really demands an alphabet with both capitals and small letters. Many interpretations depend upon case distinction, although there is a standard for single-case and limited alphabets [6]. Examples:

```
  Term is?  15 FT/sec.
Enter "ft" if you mean "feet"
   "fT" if you mean "femtotesla"  ? ft
  output = 4.572 m/s
quantity of velocity

  Term is?  gm/cubic m
Enter "g" if you mean "gram"
   "Gm" if you mean "gigametre"  ? g
  output = g/m**3
quantity of density

  Term is?  cc/ml
Enter "cubic cm" if you mean "cm**3"
   "cC" if you mean "centicoulomb"  ? cubic cm
     = 1
```

Variation in Human Input

Programs of this kind are a natural target for the perverse (and not so perverse) user. "Here is this program", the user is inclined to feel, "pretending to act human -- so why not outsmart it?" Several defensive devices are employed:

- A terminal "s" is first assumed to be a plural, and the search is made without it. If no hit is found, the "s" is restored for another search (the unit may have been "siemens" or "gauss"). This means that the table must have certain variant entries, such as "inche" or "henrie" or "hertze", even though "hertzes" is incorrect, and we'll tell the user that later.

- Abbreviation by commonly-used symbols must be recognized in the table -- the double quote for inches, the single quote for feet, and the "#" for pounds. Letter abbreviations must exist, and so must plural variants -- foot, feet, ft, inch, in.

- Then there is the common malpractice of the medical profession, not subject to lawsuit, of using "gm" for gram, instead of "g", and "cc" instead of "ml", for millilitre.

- There may be genuine Englishmen among us, spelling it "gramme", who may use the abbreviation "gr", which should really represent "grain".

- Variations of certain individual or compound terms exist as entries:

fluid dram	fluidram	fluidrachm
nmi	naut mi	nautical mile
Wh	W.h	
watt hour	watt-hour	watt-hr

- Some are inclined to also spell out the numeric constants:

  a hundred six thousand and fiftytwo metres per day

- The program is designed so that the metric (SI) prefixes) may be applied even to English units, with often curious result. Examples:

kilofathoms per centifortnight	(1.512 mm/s)
microfirkins per hectoteaspoon	(there are 12046)

- It is also prepared to accept large constants:

```
  Term is?  0.0000000000000000000000000000000000000001 cu metres
  output =  10000 fm**3
quantity of volume

  Term is?  10000000000000000000000000000000000000000000000 m**3
  output =  10 Pm**3
quantity of volume
```

- It is prepared to accept commas marking off triplets of digits, and may be modified to accept the comma as the radix point.

The Program is Independent of Vocabulary

Conversion is table-driven. The standard entry is in this form, separated by vertical bars for scanning:

```
UN|[aPb]SIU|T|Q|SN
```

where:

```
UN  = unit name (e.g., fathom)
aPb = conversion to the proper Si unit, signifying
      a times 10 to the power b (may be null)
SIU = the proper SI unit symbol (e.g., m)
T   = type code (0, if an SI unit)
Q   = quantity clause, e.g. "angular velocity"
SN  = Serial number for the SI unit
```

The general procedure is:

1. Split input on "per" or "/"
2. Find numerator
3. Evaluate its constant, possibly modified by [aPb]
4. Pick up unit number for numerator
5. If denominator exists, repeat 2
6. If denominator exists, repeat 3
7. If denominator exists, repeat 4
8. Put both constants together for final
9. Search for ratio of serial numbers

Let's follow that for an input of "25 poles per 1.66 minutes". The entries in the table are:

```
pole|[5029159p-6]m|6|length, in metres, not poles|01

minute|[6p1]s|1|time|03
```

A pole (type 6 denotes Old English unit) is 5.029159 metres. A minute (type 1 denotes nonmimimal multiple of a SI unit) is 60 seconds. The combined constant is

$$(25 \times 5.029159)/(1.66 \times 60) = 1.26233 .$$

Now the table is searched for the ratio "01/03", and it yields "m/s", so the answer is:

1.262 m/s, a quantity of velocity

In some cases the ratios will reduce to single SI units. For example, kg.m per second per second (30/27) reduces to newtons of force (16); newtons per square metre (16/25) reduces to pascals of pressure.

Acknowledgements and Permissions

The SI program was first demonstrated publicly at the 4th Annual Conference of the American National Metric Council, in Washington, DC, on 1979 April 02-04. Two terminals were in operation, one connected to FSO in McLean, the other to System X in Phoenix. Most attendees were surprised to see how much intelligence could be incorporated in a computer program, and all that played enjoyed it.

No decision has been made about whether to market the program for revenue, or to donate it to HLSUA and schools throughout the country. If provided free, however, there must be a stipulation that the program must not be altered to accept the provincial spellings "meter" and "liter". They cause too much confusion in usage and pronunciation.

If your favorite Old English term is missing from Figure 2, call 8*357-2569 to have it added.

I thank Louis Sokol, President of the U. S. Metric Association, for support and suggestions during my metric training. Most thanks go to Eric Clamons and Rick Keys, who created and built the TEX language and processor, without which I do not believe such a program could have been constructed!

REFERENCES

1. The TEX Subsystem of the Timesharing System, Series 60 Level 66, Honeywell Information Systems, 200 Smith Street, Waltham, MA 02154, Order DF72.

2. R.W. Bemer, "Introduction to the TEX language - Part I", Interface Age Magazine 3, No. 8, 144-177 (1978 Aug).

3. -, "Introduction to the TEX language - Part II", Interface Age Magazine 3, No. 9, 124-127 (1978 Sep).

4. -, "Introduction to the TEX language - Part III", Interface Age Magazine 3, No. 10, 126-131 (1978 Oct).

5. ASTM Standard for Metric Practice, E 380-76, American Society for Testing and Materials, 1916 Race St., Philadelphia, PA 19103.

6. International Standard 2955, Information processing -- Representation of SI and other units for use in systems with limited character sets.

printout

Vol. 9 No. 23 Large Information Systems Division ■ Honeywell ■ June 14, 1979

How many barleycorns in a foot?

Honeywell has a computer program that will tell you not only how many barleycorns there are in a foot (36), but also how many teaspoons to a tablespoon (3), and how many firkins to a hogshead (5.829).

If you want to convert a customary term like the familiar foot to the metric system the computer program will tell you it's 0.305 metres. If you want to go the other way and discover how many inches, say, are in .375 of a metre, it will respond 14.764.

If you don't know what the metric prefixes mean you can enter them into the computer and it will tell you that the term peta, for example, means 10 to the power of 15. Or, if you don't know your units of measurement, you can discover that the term tesla is a quantity of magnetic flux density.

It will do stones per angstrom (63.503 Tg/m), statvolts per abampere (29.979 ohm), yards per bolt (40), cups per gallon (16), megafathoms per milliday (21.167 km/s), or seconds per century (3153600008).

The computer will convert light-years per year to 0.3 Gm/s. In other words it tells you that 0.3 gigametres, or 300 megametres, per second is the speed of light.

The program was demonstrated recently by its author, Honeywell's Bob Bemer, widely known as the father of ASCII, at the American National Metric Council Confrence in Washington. Bemer believes it could have wide use as a learning tool for metrication since it simply asks what you wish to convert and does it. "You can sit at a terminal all by yourself and learn at your own pace. No embarrassment, no teachers-nothing is required but the time to play and experiment," he said.

The program is driven by a table of more than 300 units and their conversion factors. It is written in TEX, which is a full programming language like many others, except that, according to Bemer, it possesses "exceptional string-handling ability and subsumes a text editor and local file." TEX is available on Honeywell's large-scale Level 66 DPS computers and their predecessors using the GCOS operating system.

American Motors changeover made in one weekend

American Motors Corp. replaced a 13-year-old Honeywell mainframe computer with a larger, more modern Series 60 Level 66/60 in one weekend. Programs that ran on the old one were started up on the new system Monday morning without requiring any new programming.

"We bridged a hardware technology gap of 13 years without changing a single line of code," said Martin Mutz, AMC's corporate director of information systems. "And we were able to do that because the Honeywell GCOS operating system works for all models in the Series 60 line.

"Application programs are transferrable among the entire span of the Level 66 mainframe and minicomputer families. Applications developed on a mainframe can be run on a mini. Also, the central processors are field upgradable and the next larger or smaller processor can be substituted to balance computer power with need."

Commonality of operating systems and transportability of applications are among the main reasons American Motors standardized its widespread data processing operations on Honeywell mainframe and minicomputer systems.

"Everybody makes good hardware," Mutz said. But he cautioned that the operating system, software and conversion-free growth are the most important factors. His confidence in Honeywell's operating system software is based in part on his experience in replacing the old Honeywell mainframe and partly on the demonstrated compatibilities between Level 66 mainframes and Level 6 minis. "This provides an environment in which user application programs can be developed and enhanced without expensive and time-consuming conversions," he said.

AMC's data processing activities grew like those of many other large organizations--separate facilities acquiring computers from different vendors, and with software programs and general operating know-how not interchangeable between facilities. AMC thus found itself with both Honeywell and IBM computers handling similar applications at its largest data processing site, the Corporate Information Services Center in Detroit.

American National Metric Council

1625 Massachusetts Avenue, NW, Washington, D.C. 20036 (202) 232-4545

April 24, 1979

Mr. Robert W. Bemer
Honeywell Information Systems, Inc.
P.O. Box 6000
Phoenix, Arizona 85005

Dear Bob:

On behalf of the American National Metric Council I wish to extend our sincere thanks and appreciation to you for your participation in our Fifth Annual Conference. We are extremely grateful to you for taking time away from your busy schedule and helping make the "Metric Conversion of Computer Programs and Data Bases at Least Cost" Session of our Conference a success.

With best wishes and sincere thanks, I remain

Very truly yours,

Stephen

Stephen A. Vastagh
Program Manager

SAV:jw

INTERFACE Data Communications
INTERFACE '79

The Seventh Annual Presentation
of the Original Data Communications
Conference and Exposition.

Co-Sponsored by
DATAMATION
Magazine

160 Speen Street
Framingham, MA 01701
(617) 879-4502
Outside of Massachusetts, call
Toll Free (800) 225-4620

February 15, 1979

SURROGATE

Mr. Robert Bemer, Consultant
Honeywell Information Systems, Inc.
P.O. Box 6000
Phoenix, Arizona 85005

Dear Bob:

Our phone conversation of the other day resulted from my appreciation of your reputation as a recognized authority and speaker on the subject of code and documentation standards. This is a critically important concern for managers and users of data communications, and therefore constitutes a major part of the conference program of INTERFACE '79, the world's largest annual data communications conference and exposition.

Over 200 speakers will participate in this year's INTERFACE -- our seventh -- at Chicago's McCormick Place, April 9-12, 1979. Attendance is expected to exceed 15,000, and hundreds of major vendors of data communications products, services and supplies will be exhibiting their wares in an area larger than five football fields.

My assignment is to develop the conference program, and I have been instructed to invite as speakers only the most knowledgeable and capable in their fields. I am, therefore, pleased to confirm your participation as chairman of the INTERFACE '79 session on "The Communicating Word Processor" (WPD-1), scheduled for Monday, April 9, from 2:30-4:00 in the Mayer Room; and speaker at "Controlling File Access and Physical Security" (SEC-1), scheduled for Monday, April 9, from 4:00-5:30 in the Lawless Room. Your SEC-1 session chairman, Mr. Harold Uhrbach, will receive a copy of this letter.

INTERFACE '79 is co-sponsored by Datamation magazine and actively supported by more than a dozen other major industry publications, including Data Communications, Infosystems, Telecommunications, Communications News, Mini/Micro Systems, Minicomputer News and Computer Data. It will be extensively promoted by these publications as well as by the general media, and will therefore afford yourself and your organization considerable exposure as leaders in your field.

The enclosed includes an overview of INTERFACE '79 and an outline of the conference program. As a conference participant, you will, of course, be automatically registered (normally a $95 fee) so that you may attend any other sessions you wish.

102 Safe and Secure Systems (SEC-1)

SEC-1 Controlling File Access and Physical Security

Mon., April 9, 1979
4:00-5:30; Lawless Room

An ingenious young student in a major northeastern city managed an unauthorized access to the local police department's data base. He used that unauthorized access to issue warrants for the arrest of his high school principal and an unfriendly teacher. The potential for such mischief (and much worse) is omnipresent, and must be guarded against with techniques that stay one or two steps ahead of the mischief-makers. Security of data is a vital consideration for any installation, but so also is protection against such natural disasters as fire, flood and earthquake. Attendees at this session will get a good grounding in these basic security concerns.

Chairman: **Harold Uhrbach,** President, DBD Systems, Inc., Rockville Centre, NY. Mr. Uhrbach is an internationally recognized consultant and lecturer with 25 years experience with management information systems and data base development. He has designed and implemented systems for government agencies, as well as banks, insurance companies and other industries. Previously, Mr. Uhrbach was involved in informational systems development, EDP review and audit, and hardware/software evaluation and selection for Auerbach Publishers, Inc. in New Jersey. He holds a BS degree in accounting from New York University and is a member of ACM.

Robert William Bemer, Consultant, Honeywell Information Systems, Inc., Phoenix, AZ. Since 1974, Mr. Bemer has been Editor of the Honeywell Computer Journal and Staff Consultant to V. P. ASTO. Previously, he directed various projects at IBM, Sperry Univac and GE. He is credited with writing the first paper on Timesharing and developing COMTRAN, one of three major inputs to COBOL. Mr. Bemer has authored over 80 articles and has been published in German, Dutch, and British magazines. He holds an AB in mathematics at Albion College and a Certificate in Aeronautical Engineering at Curtiss-Wright Tech.

James Booth, Marketing Manager, Info-guard Data Security Products, Motorola, Phoenix, AZ. Mr. Booth has been in his present position for the past three years. Previously, he worked for Motorola's Semiconductor Division as Product Marketing Engineer for their M6800 microprocessor family of products. He is a member of ANSI X9. A. 3 (Security Committee) and ANSI ADHOC (Encryption Committee). Mr. Booth received a BS degree in electrical engineering from Arizona State University.

Jacob Sternberg, President, Conversational Systems Corp., New York, NY, a firm specializing in the research and development of computer systems and computer technology. Mr. Sternberg, a former assistant professor of operations research at the Polytechnic Institute of New York, Graduate School of Engineering, has invented and patented several automated handwriting verification systems. He holds a BS degree in mathematics from City College in New York and an MS degree in operations research from New York University.

120 Word Processing and Datacomm (WPD-

WPD-1 The Communicating Word Processor

Mon., April 9, 1979
2:15-3:45; Mayer Room

Word processors are nice. *Communicating* word processors are better! That seems to be the rationale underlying an explosive increase this past year in the sale of communicating W-P systems. This systems-oriented session will examine what's currently available at varying price levels. Attendees will learn that communicating word processors are more than just word processors with a telephone interface. Would that it were that simple. INTERFACE's independent experts, however, will provide you with the necessary guidance.

Chairman: **Robert William Bemer,** Consultant, Honeywell Information Systems, Inc., Phoenix, AZ. Since 1974, Mr. Bemer has been Editor of the Honeywell Computer Journal and Staff Consultant to V. P. ASTO. Previously, he directed various projects at IBM, Sperry Univac and GE. He is credited with writing the first paper on Timesharing and developing COMTRAN, one of three major inputs to COBOL. Mr. Bemer has authored over 80 articles and has been published in German, Dutch, and British magazines. He holds an AB in mathematics at Albion College and a Certificate in Aeronautical Engineering at Curtiss-Wright Tech.

Dale Kutnick, Director of Research, The Yankee Group, Cambridge, MA, where he edits the Yankee Group's newsletter, "The Technical Office—Research and Analysis." Previously, he was an editor/analyst at International Data Corp., where he was responsible for IDC's *EDP Industry Report*, and where he wrote the "white paper" on "Information Processing in the Office of Tomorrow" that appeared in the October '77 issue of *Fortune*.

John Peers, President and Chief Executive Officer, Logical Machines Corporation, Sunnyvale, CA, where he is involved in manufacturing and marketing the ADAM minicomputer, TINA small business computer and ABLE multi-terminal system. Mr. Peers is the former chairman and chief executive officer of Allied Business Systems Ltd. in England, sales director of Interscan of Britain, a firm selling OCR equipment throughout Europe, and he has had various assignments in programming, systems analysis and sales at International Computers, also in England.

Word Processing and Datacomm 119

These sessions examine the interrelationship between data processing, data communications and word processing. The trend toward integration of these technologies into sophisticated systems is examined with an eye to the advantages compared with stand-alone systems.

Four sessions cover the popular modes of implementing word processing in a datacomm environment. Updates on equipment and systems as well as management considerations for selecting, installing and applying those products will be covered.

Safe and Secure Systems 101

Always an important issue, security attains supercritical status in a distributed data processing environment. A multifaceted subject, DDP security encompasses security of the data (both stored and in transit); security of the site (physical security); security of the network (system security); and security of supporting subsystems (reliability), most notably the power links. These sessions address these areas as well as procedures for backup and recovery.

Luncheon Talk,
  20th Anniversary of CODASYL
  1979 May 21
R. W. Bemer

(Introduction)

The opportunity to talk to you all here has of course triggered a lot of recollections. For example, I remember that the Short Range Committee that gave us the first cut at COBOL was just that. They were supposed to get it on within 3 months. Grace Hopper and I snickered about the impossibly short time, and it appears we were justified. COBOL isn't finished yet, as we celebrate 20 years of work!

Surely you remember COBOL 60. And COBOL 61. And COBOL 68. And COBOL 74. And COBOL 80. And look forward to COBOL 2000. No wonder the inhabitants of Battlestar Galactica worshipped Lord COBOL. How immortal can you get?

COBOL must have been a significant contribution, measured by how the early proponents have been honored. Of course, Charlie Phillips and Joe Cunningham were very distinguished when they moved the project. For some others --

> Do any other countries in the world have high-ranking women officers that are so important they are not allowed to retire, like our own Navy Captain Grace Murray Hopper? But I'm not sure what vessels she may board besides the USS Constitution.
>
> And who was the first woman elected president of the Association for Computing Machinery? Jean Sammet.
>
> Would the Honorable Jack Jones have kept a Vice Presidency so long in a large company if he hadn't been so closely associated with CODASYL and COBOL? Well, yes, he would. But is that any way to run a railroad?
>
> And then there are those who prefer being their own boss -- like Howard Bromberg, who took off for San Francisco as soon as he heard about North Beach.

A Flash of History

Due to the foresight of the founders, and their cleverness in keeping the history in print via hundreds of thousands of copies of the COBOL specification, I need go into very little history here. We are, of course, celebrating that first meeting at the Pentagon, on 1959 May 28 and 29.

It was a noble venture. Jean Sammet's history quotes a motive as articulated by Charlie Phillips -- "To broaden the base of those who can state problems to computers". Possibly he actually said it; I have a great admiration for Charlie. Particularly when you consider that he has been active in CODASYL eight years longer than the formal reign of the Shah of Iran! And with every prospect of running up the total!

Jean also allowed as how the Short Range Committee was somewhat deluded, thinking that their first spec was not "something intended for longevity". But longevity it has, of course, due to the renewability permitted by the CODASYL framework. Which is the principle of my home town -- Phoenix. It would be interesting to compare today's grown-up COBOL with that upstart FACT language of Honeywell, that caused such a stir then!

The name of Jean Sammet also reminds me of a time when my wife Marion and I were both working at IBM. Seems that I spoke of Jean Sammet and COBOL quite often, to the point where my wife became so curious that it could change to jealous. But it blew over when I explained that COBOL wasn't a perfume!

Passing quickly over history, you recall that the Short Range Committee did comply very well, and their report was accepted on 1960 January 7. This led to submission for printing in April, and actual publication via the US GPO in June.

Of course all these activities caught the public eye. Business Week had been on top of the situation since June of 1959, and in April of 1960 the effort was exposed in Computing News, Issue 171. The insight of its editor, Jackson Granholm, was so penetrating that I would like to recall it to you (mainly because the New York Public Library has no back issues -- it's rumored that they were burned):

In a masterful piece of reporting, entitled "POOBLE-ORIENTED LANGUAGES", we find -

> "That the eminent Dr. Rupert B. Pooble should concern himself with the subject of programming languages was to be expected. After all, in his position as Director of Mathematical Action for the Inscrutable Atomic Corporation, Pooble swung a big mass ...
>
> Therefore, as fate would have it, it was on a well-known Tuesday during November past that Pooble called a meeting in his large, oak-paneled office. To this meeting he rather arbitrarily summoned practically everyone in the manufacturing end of the industry who could see lightning and hear thunder.
>
> It was apparent early in the gathering that the attendees tended to break pretty well into two camps. These two camps, to coin some cliches, might well be described as the "Know-Nothings" and the "Green-back Party".
>
> The Know-Nothings lined up solidly behind their idol, Horton Dreamer, Associate Director of Programmercraft for the Suffix-Specific Division of Quantum-Occluded-Domineer. The Green-back party, on the other hand, were solidly behind their eminent spokesman, Dr. Mary Margaret Groper of the Competing Equipment Corporation of America.
>
> Pooble was quick to get to the point.
>
> "It is manifest", he said, in his resonant, cultured voice, "that Inscrutable Atomic is the biggest computing machine customer in the world. We have at this very moment in the back room a total of 43 electronic computers of various sizes. These machines are on rental from 17 different manufacturers ... However", Pooble continued, "I am sorry to note that these 43 machines are programmed in no less than 678 systems of pseudocoding, not to mention their own unique machine codes".
>
> "My!" said Dr. Groper.
>
> "I have asked you here to see what you intend to do about it", Pooble said.
>
> "Not a damn thing", said Dreamer, "If you'd stuck with our equipment like any sensible person you wouldn't be in this mess".

> "Now just a dog-boned minute, Horton", Pooble said, his face growing crimson, "if you don't want your rent cut off you better shape up better than that".
>
> "No need to get hot under the collar, Rupert", said Dreamer, "but they warned me at headquarters that you were apt to pull some glitch like this".
>
> "Never mind that", Pooble said. "What I expect you people to do is to form a committee to produce, at no cost to Inscrutable, the ultimate programming language".
>
> "You're daft", Dr. Groper observed.
>
> "Nonetheless, you have two weeks to get started, or we go back to desk calculators".
>
> Manifestly the heat was on, and the attendees, knowing on which side their bread was peanut-buttered, got with it with dispatch.
>
> By 2:30 in the afternoon a name had been selected. It was decided that the ultimate pseudocode would be called "POOGOL", for Popular Operational Ordinary Glitch-Oriented Language.
>
> By the time the meeting broke up, a modest little working group of 310 members had been set up to implement POOGOL in three months (sic). Dr. Groper and Horton Dreamer were appointed co-chairmen. Since they didn't speak to each other this made for rather difficult coordination, but at least the job was under way.

There's more, the story going that the project lost it's driving force. Pooble resigned and went to work for Dreamer at twice the salary. Etc.

There may have been fallout from this reportage. When Charlie Phillips was first at BEMA, it was in New York, whose citizens drop some R's. A number of people had the misapprehension that he worked for me. This problem has now been rectified by adding a "C", to get CBEMA, whereas I'm RBEMA.

And one wonders whether Bromberg read the story, having it still on his mind as he passed a certain stoneworks holding a sale on animal tombstones!

Original Sucess of CODASYL

Although the activities are now more varied, the original success of CODASYL was the COBOL language. And it's worth reminding ourselves why that should have been so.

It wasn't that it offered a substantial set of capabilities not previously available (B.C.). The official history of COBOL recounts the proprietary languages that were absorbed, melted, and recast into COBOL. The features of each could be found, in a form that, when altered, usually was free of the prejudice and provincialism of the originator (who probably found it somewhere else, anyway).

It wasn't that the computer system suppliers of the time, overcome by user-inspired altruism and conviction that their customers knew more than they did, decided to open the halls for a camel-building party. So they could laugh at it later and hawk their own consistent product, not made by committee. Even if so, the Short-Range Committee fooled them by eventually releasing a remarkably wellmade specification, quite unlike a camel. But then when we had it, not all suppliers rushed to make COBOL their product. In at least one case, it took much arm-twisting by users, and prospective users, to get COBOL compilers in the catalog. COMTRAN isn't much remembered as one of the acronyms I devised. Fortunately CODASYL is.

Incidentally, I usually pronounce it like "codicil" in a will, because that's how it came to mind, rather than like a musical "coda". But this is one place I don't worry about agreeing upon a standard.

COBOL was successful because of the "CO" in it's name. COMMON. Not common in the vulgar sense, but common because the programs using it were fairly portable to other computers. With business data that's a lot tougher job than with floating point numbers!

COBOL was the seventh language to work on more than one model of computer, (Fortran, Algol, APT, IT, Mystic, IPL) and the third (APT, Algol) to work on models of more than one manufacturer. But it was the first business language on both counts, proved on December 6 of 1960.

Unique Role of CODASYL

The significance of CODASYL is greater than ever now. Our economic struggle is no longer within our own country. It's with other countries. And we're not doing so well, or haven't you noticed? Computers and electronic gear account for a substantial plus in our balance of payments act. The trade figures for March were released last week. The total US deficit was $821 million. Computer exports were $496 million, imports $68 million, for a net surplus of $428 million. Without computers as a viable business, then, the US deficit would have been 52% worse! So we should be careful to maintain this advantage. Do you think we can?

Last year a Frenchman told me about a business dinner he had attended in Japan. The Japanese executive next to him had become convivial enough to say:

> "Do you remember the German cameras?
>
> All Japanese now ...
>
> Remember the Swiss watches?
>
> All Japanese now ...
>
> Remember the American computers? ..."

If you're thinking "What does this have to do with CODASYL?", let me remind you of this country's antitrust laws. They may have had ample justification when passed. In each company I have worked for, I have been instructed firmly in the limits of my participation in standards activities of all kinds. Do nothing to contravene the antitrust laws, they say.

But the Japanese are not hampered that way in cooperative ventures. Just the opposite. They have government-controlled joint research, shared between companies. Free enterprise and traditional American competition are penalized in that game. Antitrust limits such cooperation between companies (although last week the government permitted Chrsyler to buy some research results from GM, so maybe they're wising up).

So who can help the U. S. to keep a competitive edge via cooperative development efforts?

- o Not ANSI. Their charter is to register standards, not develop or legislate them. They cater to more fields than just computers.
- o Professional societies can attract volunteers, but the work would have to be funded by dues. These twenty years have shown how inadequately that works. Example: ACM took 6 months after CODASYL started to even acknowledge the work.
- o The several user groups can't be expected to maintain the necessary broad scope and viewpoint required for full portability.
- o ECMA (European Computer Manufacturers Association) has done excellent work, but we can't entrust our export balancing act to them.

Strength of a country, like corporate strength, can come from cooperative ventures. From combined R&D.

Let us be grateful that CODASYL exists today, for we might not be able to establish it new today. Reminds me of a T-shirt slogan - "Do it now, before it becomes illegal". CODASYL is unique. Its fairness and propriety have been established.

It's a DATA World

It's heartening that today's meeting confirms that we backed the right horse. It's Committee on DATA Systems Languages. Most computer usage was still for numerical calculation then; at least the visible emphasis was. Papers on data manipulation were few at the conferences of the day.

I said then that business problems were 10-20 times as difficult as numerical problems, which was not a popular opinion. But it hasn't been until the last few years, working with live databases, that I realize -- in simple enough terms -- why this is so.

Computers can usually process an input number, for it may lie anywhere within a known spectrum according to certain formation rules. Non-numeric data won't work that way. More representations are possible, which is why we have alphabetic license plates, and why the Post Office wants to add four more digits to the ZIPcode. And they'll probably still insist that you give the city and state, too, or they won't send the mail.

That's the key to data processing -- pattern matching. A pattern of bits, of characters, of words, or of total behavior. It's still the key.

Real World Databases

Two years ago I was asked to demonstrate some relational database processing to a certain government agency. I was to leave Phoenix on Sunday, and their sample data just arrived Friday afternoon. I went home, fired up the terminal, built a Martini, and looked at the file which had been loaded. It appeared to be all capitals, and studded with spaces, like maybe they had used an old-fashioned keypunch to enter the data. So I used TEX to replace all double spaces with a separator, until none were left. Storing the file under a new name showed that it only took half as much space in that form! So they were paying their supplier twice too much for disk packs and drives.

But that's not the most important part. The concordance I ran was astonishing. It said that a certain petroleum company had three high-level executives with sound-alike names -- Wohlegemuth, Wohlgemuth, and Wolgemuth.

You've guessed that they are really just one man, that the database was dirty. Never mind, on a query they're only going to find him one time out of three. Do you want our government to make decisions from such data? Or do you think that maybe that's what the problem is?

That's what I mean by realworld databases and realworld people associated with entry and processing of the data. Note that CODASYL DBTG schemas would also fail in this instance if there were pointers to all three men. In the real world one has fuzzy sets. "His name sounds like ... ", or "I think it began with Gy". And in the real world one doesn't always know in advance what is expected to be extracted from a database upon query or display of some subset. One doesn't put in pointers to all of the other people in the entire world that have 1973 blue Dodges. Just as playing chess by computer cannot be done by projecting ahead all possible results of the next nine moves. It gets too astronomical.

Surely we see from the present usage of microcomputers that many of our previous tasks will be transferred to them. What's left for large computers to do? Manipulate databases, for one. Take in, via communications, the small private databases, agglomerate them, and parse them to extract new information that was previously unsuspected or unavailable to the single owner.

I don't mean to downgrade the DBTG work (and get shot by my colleague Bachman). It's a most important tool, but not the last that CODASYL should concern itself with in the database world. Parsing with pattern-matching devices of sufficient ingenuity can handle those "fuzzy sets". And it can handle the huge number of today's existing databases that are not of the pointered type.

Brief Recommendations

Peter Landin's paper "The Next 700 Programming Languages" said that if a user body becomes large enough it is economically viable to specialize to segments of usage. We must remember that we must supply tools for the pre-DBTG users, and for the post-DBTG users, as exemplified by the microcomputer and communication networks.

With this in mind I offer CODASYL some brief and modest recommendations on future actions and needs. They may be covered fully already. If so, ignore my points as fulfilled. Here they are:

1. In your database work, accord equal rights to all programming languages. Right now, COBOL is favored like a sprained ankle over FORTRAN.

2. Give consideration to pointerless databases. Consider flat files and co-files for them. Memory costs say we can do it now, and make a database understandable by simply printing it serially. Support efforts to reserve the upper half of ASCII for tokens, not printing characters (which are adequately covered by code extension). Then you may have flat files describing the token-to-actual relationships, but run the relational database with tokens only.

3. Continue the good work for the end-users, but remember that we may wish to manipulate a database, not just interrogate it.

4. Continue with the operating system command language work, but remember that it can all be done with a text processing language. Witness the success of the UNIX system and TEX. Keep close liaison between the COSCL work and the "Nicola" project under W. German government sponsorship. Mapping COSCL to Nicola will be your test of success.

5. Gerry Weinberg says "While 80% of commercial applications programming is done in COBOL, not even 5% of the programming literature deals with COBOL". What are you going to do about it?

6. Carry the COBOL lesson of levels and modules one step further, Carry it to many languges, not one. What's so particularly "database" about COBOL? You're answering that with FORTRAN, at least. But not with all of the important languages -- including BASIC, PASCAL, and PL/I. What's so particularly "realtime" about PL/I, that doesn't apply to COBOL, FORTRAN, CORAL, PEARL, etc.?

   There's a way to get rid of expensive duplication between so many programming languages. CODASYL could sponsor work to extract the common parts of all of these language. Then the standard specification for PASCAL could say "See CODASYL standard referent 6.2.3 for this function".

   Look at any big operating system. It's compiling source programs in several programming languages simultaneously, and each compiler has its own code for the functions otherwise identical in each language. I don't worry about saving memory. It's the building and maintenance of duplicate and redundant software modules that is bad. Landin says to cloak the function in whatever language is easiest for your audience and users. But I say -- don't use that as an excuse to build duplicate software.

Conclusion

The success of COBOL has been overwhelming. Its aspects and concepts have entered and modified our way of life in many ways. The 1979 March 15 issue of Computing (the British weekly) reports hearing a luncheon conversation like this: "As a manager, he's all Data Division and no Procedure Division".

Nevertheless, I want to remind you that the job isn't done, it can be done better, and it will pay off to do it better. Get in step with the microcomputer people; they're doing things we have said couldn't be done.

It has been fun talking to you. I hope that we have new achievements to be proud of ten years from now. I want to express, on your behalf and mine, appreciation for the vision of the known pioneers, for those who have left (like Roy Goldfinger), for those that worked behind the scenes (like Mary Hawes, Saul Gorn, Walter Carlson, and many others), for Rik Blasius and the helpful Canadian Government, and for those who have labored these two decades to enlarge and improve the work.

... (Balance of POOGOL satire)

Pooble put out bi-weekly bulletins on the progress of POOGOL. These were reprinted in the Journal of the Metacomputer Society, the Annals of the National Association of Machine Admirers, and even in the mimeographed newsletter of the Digital Convention Association of Southern California.

The world watched with bated breath while the shapers of POOGOL debated the limit to the number of times a subroutine was to be permitted to call upon itself, and many hours of heated debate were spent on resolving the question of whether certain quantities in a restricted class of paralyzed variables were to have "name" or "titles".

At last POOGOL began to shape up. It was announced that simple means had been discovered to permit a machine to simulate itself, or any other machine having an identical set of operation codes to its own. Further, this simulation could, in certain carefully-selected cases, proceed at only one tenth real time.

Late in February it was confidently expected that POOGOL would soon be complete, and celebrations were prepared from Upper Sandusky, Ohio, to Potts, Nevada, but sadly, at that point, the project came to an abrupt halt.

The project lost its driving force, and collapsed on internal entropy. Word came from Inscrutable Atomic that Pooble had resigned to accept a job as Horton Dreamer's assistant. Rumor had it that Pooble had double his previous salary.

102 Safe and Secure Systems (SEC-1)

EC-1 Controlling File Access and Physical Security

Mon., April 9, 1979
4:00-5:30; Lawless Room

n ingenious young student in a najor northeastern city managed an ınauthorized access to the local oolice department's data base. He ised that unauthorized access to ssue warrants for the arrest of his igh school principal and an ınfriendly teacher. The potential for uch mischief (and much worse) is omnipresent, and must be guarded gainst with techniques that stay one r two steps ahead of the mischief-nakers. Security of data is a vital onsideration for any installation, but o also is protection against such atural disasters as fire, flood and arthquake. Attendees at this session vill get a good grounding in these asic security concerns.

hairman: **Harold Uhrbach,** President, DBD ystems, Inc., Rockville Centre, NY. Mr. hrbach is an internationally recognized onsultant and lecturer with 25 years xperience with management information ystems and data base development. He has esigned and implemented systems for overnment agencies, as well as banks, nsurance companies and other industries. reviously, Mr. Uhrbach was involved in nformational systems development, EDP eview and audit, and hardware/software valuation and selection for Auerbach ublishers, Inc. in New Jersey. He holds a BS egree in accounting from New York niversity and is a member of ACM.

Robert William Bemer, Consultant, Honeywell Information Systems, Inc., Phoenix, AZ. Since 1974, Mr. Bemer has been Editor of the Honeywell Computer Journal and Staff Consultant to V. P. ASTO. Previously, he directed various projects at IBM, Sperry Univac and GE. He is credited with writing the first paper on Timesharing and developing COMTRAN, one of three major inputs to COBOL. Mr. Bemer has authored over 80 articles and has been published in German, Dutch, and British magazines. He holds an AB in mathematics at Albion College and a Certificate in Aeronautical Engineering at Curtiss-Wright Tech.

James Booth, Marketing Manager, Info-guard Data Security Products, Motorola, Phoenix, AZ. Mr. Booth has been in his present position for the past three years. Previously, he worked for Motorola's Semiconductor Division as Product Marketing Engineer for their M6800 microprocessor family of products. He is a member of ANSI X9. A. 3 (Security Committee) and ANSI ADHOC (Encryption Committee). Mr. Booth received a BS degree in electrical engineering from Arizona State University.

Jacob Sternberg, President, Conversational Systems Corp., New York, NY, a firm specializing in the research and development of computer systems and computer technology. Mr. Sternberg, a former assistant professor of operations research at the Polytechnic Institute of New York, Graduate School of Engineering, has invented and patented several automated handwriting verification systems. He holds a BS degree in mathematics from City College in New York and an MS degree in operations research from New York University.

120 Word Processing and Datacomm (WPD

WPD-1 The Communicating Word Processor

Mon., April 9, 1979
2:15-3:45; Mayer Room

Word processors are nice. *Communicating* word processors are better! That seems to be the rationale underlying an explosive increase this past year in the sale of communicating W-P systems. This systems-oriented session will examine what's currently available at varying price levels. Attendees will learn that communicating word processors are more than just word processors with a telephone interface. Would that it were that simple. INTERFACE's independent experts, however, will provide you with the necessary guidance.

Chairman: **Robert William Bemer,** Consultant, Honeywell Information Systems, Inc., Phoenix, AZ. Since 1974, Mr. Bemer has been Editor of the Honeywell Computer Journal and Staff Consultant to V. P. ASTO. Previously, he directed various projects at IBM, Sperry Univac and GE. He is credited with writing the first paper on Timesharing and developing COMTRAN, one of three major inputs to COBOL. Mr. Bemer has authored over 80 articles and has been published in German, Dutch, and British magazines. He holds an AB in mathematics at Albion College and a Certificate in Aeronautical Engineering at Curtiss-Wright Tech.

Dale Kutnick, Director of Research, The Yankee Group, Cambridge, MA, where he edits the Yankee Group's newsletter, "The Technical Office—Research and Analysis." Previously, he was an editor/analyst at International Data Corp., where he was responsible for IDC's *EDP Industry Report*, and where he wrote the "white paper" on "Information Processing in the Office of Tomorrow" that appeared in the October '77 issue of *Fortune*.

John Peers, President and Chief Executive Officer, Logical Machines Corporation, Sunnyvale, CA, where he is involved in manufacturing and marketing the ADAM minicomputer, TINA small business computer and ABLE multi-terminal system. Mr. Peers is the former chairman and chief executive officer of Allied Business Systems Ltd. in England, sales director of Interscan of Britain, a firm selling OCR equipment throughout Europe, and he has had various assignments in programming, systems analysis and sales at International Computers, also in England.

Codasyl20

AGENDA
CODASYL 20TH ANNIVERSARY
MAY 21-22, 1979
WASHINGTON, DC

MONDAY, MAY 21, 1979

0800-0900	Registration - Capitol View Ballroom
0900-0915	Welcoming Address John L. Jones, Vice President, Southern Railway System, Chairman, Executive Committee
0915-0945	Keynote Address The Honorable Elmer B. Staats Comptroller General of the United States
0945-1000	Coffee Break
1000-1045	Presentation of Codasyl System Architecture Mr. Richard Kurz, Southern Railway System Executive Committee and Committee Chairman
1045-1115	Report of the Cobol Committee Mr. Donald F. Nelson, Control Data Corporation Chairman, Cobol Committee
1115-1200	Presentation - Distributed System Report Mr. William H. Stieger, Standard Oil Company (Ohio) Chairman, Systems Committee
1200-1400	Luncheon - "Reminiscing" Speaker Bob Bemer, Honeywell
1400-1430	Data Description Language Committee Report Mr. Michael L. O'Connell, Digital Equipment Corporation, Chairman, DDLC
1430-1515	Codasyl Data Base Implementation Mr. John Cullinane President, Cullinane Corporation

DATE INFOTECH UPDATE INFOTEC

THE CODD WAR

Data base technology is one of today's hot topics — and its hottest area is the controversy over relational data bases. The relational school, led by IBM's Ted Codd, offers a theoretically-rigorous, intellectually-satisfying data base model — so far only implemented on a dozen or so experimental systems, with data bases of a few thousand bytes. On the other side is the data base 'establishment', CODASYL, with several large-scale commercially-available implementations and currently the de facto industry standard. Father of the CODASYL approach is Charles Bachman of Honeywell, who started it all back in 1962 with GE's IDS which provided most of the ideas underlying the CODASYL specification, and which also, incidentally, inspired IDMS and TOTAL, (both conceived as IBM 360 implementations of IDS).

According to the relational school CODASYL technology is a dead end. CODASYL adherents point out in defence that, to be effective, the relational data base model depends on a hardware technology that is still at the research stage — large-scale content-addressable storage.

Honeywell has a large investment in DBTG-oriented technology having inherited both IDS and Backman from GE. But it also has a strong relational contingent led by Bob Bemer, one of the best-known figures in the industry. He has implemented an on-line enquiry system based on the relational model, and is to describe his work at the Infotech State of the Art Conference on On-line Data Bases this November.

Bob Bemer, leading Honeywell proponent of the relational approach, flies into London in November at Infotech's invitation to speak at the State of the Art Conference On-line Data Bases.

IBM, too, shows signs of corporate schizophrenia. Its own DBMS, IMS, is not relational or CODASYL based. Neither is it one of IBM's most successful products. Its inherent weakness is that its complexity and resource-hunger make it unsuitable for middle-range users who provide the bulk of IBM's income.

IBM is firmly committed to IMS and appears to be against the DBTG proposals (as it was once firmly against the ANSI COBOL — remember?) It is now rumoured that a back-end IMS processor is planned. This could extend IMS's coverage and incidentally prove inconvenient for many of the plug-compatible vendors (including Amdahl Corp, with its plug-compatible mainframe), since the back-end is sure to be very choosy about whose equipment it interfaces to. Even with a back-end, however, IMS does not look too promising.

In the meantime, the relational philosophy is making considerable headway. The main reason seems to be that it has become associated with the very useful technique of data normalisation (see pages 6 & 7 of this issue). Most of the current research on relational data bases is sponsored by IBM, particularly at the San Jose California centre, where Ted Codd works. Some commentators see IBM's sponsorship of the relational approach as being a way of undermining the DBTG proposals without official involvement. An alternative view is that IBM is preparing the market for a new-generation IBM data base system — a relational DBMS packaged with some form of content-addressable bulk storage (for example, head-per-track disks incorporating comparator logic to perform a fast sequential search 'on the fly'). Such a package might skuttle both the suppliers of IBM-compatible hardware and the vendors of competitive DBMSs, and add more fuel to the CODASYL/relational controversy. An interesting prospect.

INSIDE

The program development dilemma

Question: You have developed 24 applications on your computer over the past four years with a systems/programming staff of 30. With the same staff, you want to develop 10 new applications, each of which takes 2 man-years. When will they be complete?

Answer 1: Never.

Table 1 below shows the stages in trying to produce the extra 10 programs mentioned in the question above. It is based on the assumption that all existing applications have a four-year life cycle (ie a quarter of them must be rewritten each year) and that 0.5 of a man-year is needed to maintain each existing application.

In 1976, the year starts with 24 applications, so 6 of them need rewriting. At 2 man-years each, this occupies 12 of the staff. At the same time, 0.5 of a man-year's maintenance per application takes 12 more staff. This leaves 6 for development, and they produce 3 applications during the year.

Table 1

Year	Applications at year start	Rewrites required during year	Required rewrite staff	Required maintenance staff	Available development staff	New applications developed
1976	24	6	12	12	6	3
1977	27	6	12	13	5	2
1978	29	6	12	14	4	2
1979	31	6	12	15	3	1
1980	32	7	14	16	-	-

1977 starts with 27 applications (24 original + 3 new) and another 6 of the original 24 need rewriting taking 12 staff. Because of the new applications, the maintenance load is increased to 13 staff, leaving 5 for development producing 2 new applications (ignoring fractions).

Following this process through leads to the situation in 1980 when 32 applications are developed and all the staff are needed for rewrites and maintenance, so no new development can take place. Only 8 of the extra 10 programs have been achieved and no more can be developed without more staff.

Although the figures may vary from installation to installation, the effect in many organizations is the same — the whole dp staff become bogged down in maintenance, and increased numbers of staff is seen to be the only way of increasing the number of applications. But is it?

Answer 2: 3 years

Table 2 shows the same basic situation but with the life of the system extended from 4 years to 5 years, and the maintenance effort decreased from 0.5 of a man-year to 0.4 of a man-year per application.

The results speak for themselves. The extra 10 applications which were impossible to achieve in the first case are easily developed in just over three years. In fact, as shown, 12 new applications can be developed, bringing the total to 36 with 6 staff still available for development.

Table 2

Year	Applications at year start	Rewrites required during year	Required rewrite staff	Required maintenance staff	Available development staff	New applications developed
1976	24	5	10	10	10	5
1977	29	5	10	12	8	4
1978	33	5	10	13	7	3
1979	36	5	10	14	6	3 (if required)

This example illustrates the importance of improving the maintainability of programs and hence, not only reducing the effort needed, but also extending the system life cycle. These are precisely the advantages obtained through the use of a structured design and programming methodologies. Figures for the Michael Jackson Technique in fact show reductions in maintenance effort far larger than that quoted above. 50% - 80% is more the region of maintenance reduction quoted. Extension of the life of the project is more difficult to prove because a number of years have to elapse before this benefit can be noted. Some dramatic results of failed projects rewritten with structured techniques, however, illustrate that the likelihood of systems lasting longer is very high.

The example shown does not take into account any reduction in time for developing or rewriting systems using structured techniques. At present, some reductions have been shown but these have generally been minimized by extra time taken sorting out specification problems. This ability of structured techniques to discover such problems at an early stage is, of course, a major contributory factor to the reduced maintenance effort required.

Advantages of structured design are usually phrased in the form of improved quality, easier understandability and easier maintenance. These are undoubtedly true, but the real advantage may well be that it assures the future of the data processing function. ●

66 OPEN QUOTES 99

'It has been said that most programming is done by simple-minded people and that is why COBOL is so widely used. My conclusion is that I must be an absolute moron; I am so simple-minded that I cannot use COBOL because it is much too difficult. It is so chaotic that I just cannot use it.'

E W Dijkstra, Technical University of Eindhoven, in the State of the Art Report *High Level Languages.*

'COBOL aimed at readability but achieved only prolixity.'

C A R Hoare, Queen's University Belfast, in the State of the Art Report *Computer Systems Reliability.*

'I have never actually heard anybody say how good COBOL is; I have heard people say it is useful, invaluable, or the only thing we have, but never good.'

C Strachey, University of Oxford, in the State of the Art Report *High Level Languages.*

'The realization that COBOL was designed not only before the System/360 but also before its predecessor gives an indication of the size of the technology gap now being dealt with. The original compilers were written for machines having core store at a premium and with an instruction set that today would disgrace a pocket calculator.'

J Maynard, Independent Consultant, in the State of the Art Report *Commercial Language Systems.*

Commentary

Standards for Keyboards

R.W. BEMER
Honeywell Information Systems, Phoenix, AZ 85023, USA

Most Video Display Unit (VDU) standards in force, or in process, concern factors of physical health. Mental health aspects, particularly for keyboards, have been neglected due to failures within the standards bodies. A method of correction is proposed.

Key words: VDU, keyboard, ergonomics, human factors, typewriter, layout, character set.

When using more than one type of terminal or microcomputer, unwarranted variations in keyboard layout and functionality are not just disturbing. They lead to error and reduced productivity. This note suggests certain areas where designers and manufacturers could find agreements, if not standards, to solve these problems.

We are bogged-down. The standards bodies have failed. The existence of more than one formal standard for keyboards has released any possible pressure upon designers to follow any standards at all, opening the door to individual and arbitrary choices. Moreover, most designers have exercised this individuality with little knowledge of worldwide needs.

I do not ask for standards to be retrofitted to hardcopy and dumb terminals. They will disappear soon enough because of (a) greater maintenance and purchase costs for hard copy; and (b) higher system costs for dumb terminals (i.e., intelligent terminals can do work cheaper than the combined front-end and communication costs).

I do seek standards for terminals that are controlled by one or more microprocessors (the "or more" is because at least one of them should be downline loadable). Given this premise, each key may be identified by its physical position on the keyboard, and may create alternate encodings (perhaps unrelated) according to the use of the shift or control keys. This implies that, regardless of what is engraved on the keytop, there should be some basic agreements on the positions themselves, and how many. For example:

- No more than ten keys should lie between lefthand and righthand shift keys (e.g., no new key to the left of the Z in the QWERTY keyboard) because:
  (a) Experienced typists, whom we may wish to use computers with a minimum of training, find this frustrating.
  (b) Computer programmers, often (unnecessarily) using uppercase for their programs, work with the caps lock on. This applies only to capital letters. The shift key must still be used to get a parenthesis for a formula. If accustomed to a keyboard without such an interposed character, they invariably enter it instead of the parenthesis. The U.S. space program knows well the cost of wrongly punctuated FORTRAN programs!
- Certain functions should not be assigned to the central area of a keyboard. Break and Line Feed are examples of functions which should be outside, to avoid accidental actuation.
- In general, there should be a fairly smooth

Bob Bemer has worked with computers for 33 years – at RAND, Lockheed, IBM, Univac, Bull GE, General Electric, and Honeywell. His main areas are software and standards. He developed FORTRANSIT (first language to run on both binary and decimal computers) and Commercial Translator, an input to COBOL (COBOL and CODASYL are his coinage). He was a major factor in the development of the ISO Code (ASCII) and contributed six characters to it, being the inventor of the escape sequence. He drafted the original scope and program of work for ANSI X3 and ISO YC97, and chaired TC97/SC5 for eleven years.

border that could be drawn around the key positions assigned to generate ASCII (ISO Code) encodings. Keys that generate escape sequences should lie outside this border (except as they may be created with ASCII keys and Control).

Now consider the need for plural standards for keyboards. I exclude the differences necessary for national languages, choosing not to argue with the voluminous IBM catalog of keyboards. The major problem is the "bit-paired" keyboard where the character pairs on each key, no matter how dissimilar to each other or contrary to a typewriter keyboard, are forced by the fact that the two characters differed by the one shift key controlled bit in the bit-coded representation. It is an anachronism, arising from the early technology of the Model 33 Teletype and dumb CRTs. But a microprocessor can be programmed with tables of correspondence between the keyboard location (with shift or control) and the generated code. One might imagine the program interrogating the currently attached keyboard to select the tables appropriate to it!

However, complete freedom is not as possible as some U.S. designers think. Bit-pairing is still desirable for the left and right bracket-brace combinations, and the reverse slash and vertical bar, because they become dual-case Scandinavian (or other) alphabetic letters. Most people wish to have the same words generated even if the shift key state changes. In this instance we see the need for programming the caps lock.

So far only layout and assignments have been discussed. Functionality is another problem. The most blatant misusage has been equating the backspace key with cursor backward. When designers do this in such profusion, you *know* that their guidelines from the standards bodies are insufficient!

The best of future terminals will indeed display compound characters as they would be printed for hard copy. It is a necessity for highlighting alone. The backspace is a primitive but effective way to make compound shapes/characters. Cursor movement, to the contrary, is associated with editing à la Word Processing, where "what you see is what you get" is the major principle. This corruption of backspace is contrary to all Word Processing and other good usage.

An important human factor is consistency. Most autos have the brake pedal for the right foot, making it most difficult to simultaneously depress the accelerator. One expects it to be there. Many questions of human factors for keyboards remain unresolved for standards, e.g.:

- What is the best location for cursor backward and forward? Many keyboards have them in a North/East/South/West cluster around cursor home. Many microcomputers have them at the ends of the space bar, a possibly more natural location, being within the home position of the hands.
- What is the best location for function key groups? I don't like soft function keys on the screen because one must reach up further out of the home position, and then drop back to key numeric parameters following them.

Conclusion

An overview of my opinion is:

- A single standard keyboard is not feasible because of language differences, even within the Roman alphabet family.
- These variations, and the existence of dual standards for even the same language, have fostered independence and arbitrary differences to the detriment of users.
- It is however possible for the standards bodies to:
  (a) Formally cancel the bit-paired standard, as of some published date, for certain classes of equipment.
  (b) Write guidelines or standards for keyboard design, incorporating recommendations on points such as those considered in this paper.

In 1970, as part of a proposed National Computer Year promoted in conjunction with ACM 70, I instigated the "First Workshop on Terminals". Unfortunately the dialog started there has not continued to keep up with technological developments, probably for lack of regular formal sponsorship. Perhaps it could begin again, partly under the auspices of standards-making bodies or technical associations. Possibly CBEMA, which sponsors ANSI X3 and the secretariat of ISO TC97. More probably, ECMA (European Computer Manufacturers Association), which is more concerned with the multilingual aspects, and operates more decisively as well.

There are monetary and human losses of some magnitude that are due to arbitrary variations and mistakes in keyboard design. I think something should be done about it before the whole world uses keyboards.