

The Analytical Engine

NEWSLETTER OF THE COMPUTER HISTORY ASSOCIATION OF CALIFORNIA

Editorial: Hello, World

As I write this, there's a week of tightening and formatting to be done before ANALYTICAL ENGINE Volume 1, Number 2, gets uploaded to the Internet and our growing list of private bulletin boards. We know now how much work — and how much fun — producing the ENGINE really will be.

Luckily, for us, for you, there's joy in our hearts as we do it. What a scramble! What a puzzle! What elation! What an education! Above all, what a sense of that good old, great old.... Right Thing.

Some people answered the editorial in the July ENGINE by commenting that it almost sounded panicky, as if we thought we were alone. Well.... not exactly. It was just that, working almost entirely from experience in the East and Midwest, we were afraid that we wouldn't find a comparable level of dedication in California — one of the most interesting and pivotal places in all of computer history. We lobbed ENGINE #1 into the dark of the future, thinking that we had "dropped the rose petal into the well and waited for the splash," really only scared of silence, of politely spattering applause, of nothing.

The Internet's fiber backbones glowed. The lights of our modems lit up and stayed lit. The snail-mail arrived with rubber bands around it. The first phone call came from Toronto; the first twenty-five-dollar check, from Idaho, the second one from Pennsylvania. We heard from the Computer Museum in Boston, the Smithsonian Air and Space, the Babbage Institute, from DEC and Intel and SUN, and the list went on.

Californians especially tended to reply by e-mail, so since the first week in July when the ENGINE hit the net, we've received almost five hundred pieces of e-mail. That *doesn't* count global postings on USENET, either. We heard from people who understood what we were doing and people who

didn't. We were applauded for our guts and excoriated for our mistakes. We found inquiries, smiling from our screens, about starting the Computer History Association of Iowa; of Colorado; of the Northeast and Northwest. (Keep that going! It's great!)

So we've learned more and slept less, and now we know — begin to know — what's really out there. We are not alone, and there is no silence. There are *thousands of people* in California, in this nation, and in the world, who care about computer history, about keeping it safe, and about making it known. This is no silence. This is a roar.

And if that's what came of ENGINE #1 —

Thank you, everybody who called, posted, wrote, grinned, welcomed, barked or bit. Thank you, everybody who subscribed or donated. Thank you, everybody who wrote an article — or even promised one. Thank you, everybody who gave advice and ideas and time.

Welcome to ENGINE #2. It's six times thicker than ENGINE #1. And it's all yours!

Hello, world!!

INITIATIVE 1999:

Warnings of Extinction

If you love old iron — and so many of us do — consider this quote from *Accidental Empires* by Robert X. Cringely, Addison-Wesley, 1992:

....you and I can go even further [than Bill Gates]. We can predict the date by which the old IBM — IBM the mainframe computing giant — will be dead. We can predict the very day that the mainframe computer era will end.

Mainframe computing will die with the coming of the millennium. On December 31, 1999, right at midnight, when the big ball drops and people are kissing in....Times Square, the era of mainframe computing will be over.

Mainframe computing will end that night because a lot of people a long time ago made a very simple mistake. Beginning in the 1950s, they wrote inventory programs and payroll programs for mainframe computers, programs that process income tax returns and send out welfare checks — programs that today run most of this country. In many ways those programs have become our country. And sometime during those thirty-odd years of being moved from one mainframe computer to another, larger mainframe computer, the original program listings....were just thrown away. We have the object code....which is enough to move the software from one type of computer to another. But the source code — the....details of how these programs actually work — is often long gone, fallen through a paper shredder back in 1967. *There is mainframe software in this country that cost at least \$50 billion to develop for which no source code exists today.*

This lack of commented source code would be no big deal if more of those original programmers had expected their programs to outlive them. But hardly any programmer in 1959 expected his payroll application to be still cutting checks in 1999, so nobody thought to teach many of these computer programs what to do when the calendar finally says it's the year 2000. Any program that prints a date.... and that doesn't have an algorithm for dealing with a change from the twentieth to the twenty-first century, is going to stop working. I know this doesn't sound like a big problem, but it is. *It's a very big problem.*

Looking for a growth industry in which to invest? Between now and the end of the decade, every large company in America will have to find a way to update its mainframe software or....write new software from scratch.... Either solution is going to cost lots more than it did to write the software in the first place. And all this new mainframe software will have one thing in common; it won't run on a mainframe.... which is why the old IBM is doomed.

The reasoning behind this is various, but the simple case is that many older mainframes store their dates in the format

YYMMDD

and, when YY returns as 00, will halt on error. Sure, there may be workarounds. But for lots of older computers, the programming overhead of dealing with this kink will be the last push over the cliff. Cringely's right; mainframes will be scrapped wholesale, and the oldest first. From the standpoint of function it only makes sense, since the oldest hardware is usually the slowest. But to the historian and preservationist, the oldest hardware is often the most significant.

If we intend to respond to this crisis, we have seven years to make plans and marshal resources; seven years to find and equip facilities; seven years to nail down funding. And for a project of this size, seven years is not a long time. Anyone seriously interested in preserving the history of computing — which certainly means any reader of this newsletter — is actually advised to figure that we're in a screaming hurry.

With this issue of the ANALYTICAL ENGINE, the Computer History Association of California announces INITIATIVE 1999. What this is, and what it becomes, will be elaborated in future issues. For the moment, just plant two cardinal points in your mind:

- 1) On or before January 1, 1999, we would like to see chapters of this Computer History Association established in every state of the Union. To that end, we will advise, collaborate with, and give moral support to any responsible groups of historians and preservationists who express serious intention of founding such an Association.
- 2) On or before January 1, 1999, we intend to open a museum large enough to display a significant part of the history of computing in California, presenting the broadest available spectrum of appropriate artifacts, and using the (then) most contemporary technology for instruction by interactive and virtual means. To that end, we would appreciate the donation of (for example) a large disused factory or warehouse, convenient to freeways, and with loading docks; of pertinent hardware and software; of expert consultation, particularly with

reference to accession, registration and curatorship; and of appropriate amounts of money.

We reiterate: Seven years is not a long time. What we're trying to do here can only be done once, or given up for lost. If you're reading this newsletter, you *can* help, with a \$25 donation to the ENGINE or with that factory.

Save the mainframes!

COMPILATION PROJECT: Getting It All Together

As basic research, fundamental to tracing the provenance of donated hardware, software, and documentation, the Computer History Association of California is compiling a list of

- 1) Computer **HARDWARE**
developers/manufacturers
- 2) Computer **SOFTWARE**
developers/manufacturers
- 3) Computer **PERIPHERALS**
developers/manufacturers

now or formerly incorporated or headquartered in the State of California. Since information on businesses currently operating is relatively available from published sources, priority should be given to citation of businesses no longer operating.

Information wanted in each citation is:

- Business name
- Primary business address
- Telephone or fax number
- Date of first business done, or incorporation
- Date of last business done, if known
- Types of hardware or software produced
- Maximum annual dollar volume and year in which recorded
- Names of officers, as known

Please do *not* include citations for retail outlets; regional offices of non-California companies; or reseller/distributors.

Depending on its length, this list may ultimately be published as a request supplement to the ANALYTICAL ENGINE, or as a separate publication. We are grateful for all contributions

and for your attention. Please reply by Internet mail, or by paper mail to the CHAC El Cerrito address.

DIGITAL'S HISTORY PROJECT

Lawrence C. Stewart at the Cambridge Research Laboratory of Digital Equipment Corporation (DEC) has sent a memo about an interesting and ambitious project to produce comprehensive, documented emulations of historically important computer systems. This is the summary:

"The idea of the Computer History Project is to preserve the history of computing systems, and to make that history readily available to everyone....to publish a CDROM, or perhaps a series of CDROMs, which would contain emulators for historic computing machines and copies of their operating software and documentation.

Modern computers are so much faster than historic machines that it is possible to emulate the instruction set and I/O systems of a historic machine at full speed. Thus students will be able to actually sit down and use TENEX running on an emulated PDP-10, or Bravo on a Xerox Alto, even when no more PDP-10's or Altos exist."

To receive a copy of the complete memo, a highly worthwhile document, send a message to

engine@win.net

with a *subject* line of

decmemo

or request hard copy from the El Cerrito mail address.

VOLUNTEER LIAISON BETWEEN CHAC AND DEC

Richard Secrist of Digital Equipment Corporation has volunteered to act as an informal focal point for employees of Digital interested in contributing to CHAC.

Physical address:

R.C. Secrist
 Digital Equipment Corporation
 412 Executive Tower Drive, Suite 300
 Knoxville TN 37923
 Internet: secrist@kxovax.enet.dec.com

[Thanks, rcs! This and the previous item exemplify the concern for, and commitment to, computer history demonstrated by many DEC employees. A similar tip of the hat to our friends at Apple, Intel and SUN. Conversely, there are some awfully big corporations that we only *wish* we'd hear from....and you know who you are.]

INTEL MUSEUM REFRESHES ITS EXHIBITS

The Intel Museum, established in 1992, has been renewing and polishing its exhibits, and makes a highly recommended stop for anyone interested in recent computer history.

Intel Corporation was founded in 1968, in a small building in Mountain View, CA, USA, by Robert Noyce, Gordon Moore and Andrew Grove. The company's first-year revenue was less than \$3,000! Today, of course, Intel's 25,000 employees make thousands of products, ranging from memory chips to supercomputers and including the famous ix86 microprocessors that power the majority of the world's small computers.

The company has packed an amazing amount of history into twenty-five years, and the Intel Museum uses the latest assistance — including interactive video and real-time automated displays — to give the visitor a sense of that history in depth. While the focus is understandably on Intel's particular accomplishments, there's a lot to be learned here about the techniques of technology, as well as history. If you've ever wondered "what makes a computer a computer," this one's for you.

Intel Museum
 Robert Noyce Building
 2200 Mission College Boulevard
 (off Great America Parkway north of 101)
 Santa Clara CA 95052
 8 am - 5 pm Monday through Friday,
 admission free
 408/765-0503 for information

MUSEUM PLANS AT UNIVERSITY OF CALIFORNIA, DAVIS

Dick Walters of the UC Davis Computer Sciences Department writes:

➤ I have been involved in the microcomputer revolution since 1975, building my first IMSAI in 1976. A few years back, I started to collect....with the idea of setting up a museum at Davis. The idea is gaining momentum, but very slowly.

Some of the items we now have on hand include:

Altair; 5 or more IMSAI systems, some with disk; TRS-80 Mod 1 and Mod 2; Osborne; Kaypro; Data General DT-1 laptop; 3 or more Cromemco systems; Heath 89; Franklin; Processor Technology SOL; Sanyo; Ohio Scientific; Zendex; IBM Mod 10 keypunch; Teletype, peripherals, and miscellaneous items. We also have documentation for most of these systems.

I am interested/willing to serve as a focal point for the collection of more gear relating especially to the micro world. We do not have room here for....mainframes and workstations are probably a little beyond our current capabilities. I would also propose exchange of some duplicates for wanted items....

I welcome people interested in....applying pressure to promote the formation of a real museum/display facility for these items. We show off many of them on our annual UCD Picnic Day (last year our first effort in this regard) but we need more permanent housing than my research lab.

Interested parties should contact me at:

Department of Computer Science
 UC Davis
 Davis CA 95616-8562
 phone 916/752-3241
 fax 916/752-4767
 Internet: walters@cs.ucdavis.edu

URGENT: SPOTTERS WANTED

With this issue, the ANALYTICAL ENGINE goes mainstream, or sort of. In late October or early November, the paper edition will be distributed to selected metropolitan newspapers, to the computer

press, to archival sites, and to paying subscribers who request it.

This raises the question of finding reviews and announcements in the press. Tearsheets are a bygone courtesy and clipping services — especially for magazines — startlingly expensive. Yet we need to know what the media are saying about us, and for more reasons than simple vanity.

Save us from the dire choice between ignorance and poverty! If you see any mention of CHAC or the ENGINE on published paper, please do one of these three things:

- * If your copy of the piece is clippable, clip and mail to the El Cerrito address.
- * If you can't spare the physical copy, send net.mail to cpu@chac.win.net, or photocopy and fax to the El Cerrito address.
- * If you're too busy for that, just send the publication name, date and page number and we'll do the hunting.

Thanks!

DESPERATE PLEA FOR STORAGE

We need storage space for hardware and documentation — tight, dry space — and lots of it.

Admittedly, we had a few micros before the CHAC ever began, and now we have quite a few more. Most of them are in excellent condition and almost all are bootable. Shortly they'll be housed in a rented locker, which is expensive, and it isn't completely appropriate.

Before long, the inappropriate will become impossible, when the bigger iron arrives. We've been offered a full-house PDP-8i that spent its whole life in honorable service to the State of California — but we can't afford rented storage for it, and it won't fit in anybody's garage. Will we have to watch it end up as landfill? And when we're asked about other, bigger, computers, will we have to let those go too? Because we have no place to put them?

Just as we need the museum for the computers (see INITIATIVE 1999) we need the computers for the museum. Collectively we're awed by mainframes, fascinated by minis and completely smitten with

micros; we're constantly fighting the constraints that would force us to be "architecture bigots" and favor the platforms that take less space.

The history of computing in California takes up *serious* racks. We're getting our 501(c)3 nonprofit status precisely so that we can offer deductions to those generous people (and companies) who will donate the things that we can't pay for. Do you have warehouse space you're not using? Donate it, please! and we'll give you a writeoff, put your name on a plaque....

DESPERATE PLEA FOR MONEY

We don't just need money. We need *more* money. And there's a special reason.

The Computer History Association of California is a very small organization that needs room, in its architecture, to get much bigger over the years. We don't want to be intimidated by current constraints, build tight, and then hang bags on the sides without being strategic. We all know what that would lead to....

We *will* succeed in our mission, *if* we can reach out to a truly representative sample of the computing community. This means going beyond electronic communication to hard copy of the ENGINE, to press releases and news stories and events. It means making the ENGINE into a "real magazine" as soon as our subscriber base permits. It means forging links with trade publications, industry executives, and foundations. It means, in a word, being taken seriously.

That's why our watchword is "Do it now *and* do it right." With a handful of members and one tiny office, CHAC doesn't *need* to be a corporation — but incorporating now will smooth our path as we grow larger. With a trickle of donations, we don't *require* nonprofit status — but that voluminous paperwork is easier to file now than later. We don't *have* to arm-wrestle a VISA provider into handling subscriptions, but....

Remember: The earliest money is the best.

Help us do it now.

Help us do it right.

Help us be what we must be, today and in 1999.

Please subscribe, and give.

AND SPEAKING OF MONEY....

E-mail is like the lunch date you make over your shoulder. It's too easy to commit and forget. Tapping out an "Oh, sure" and hitting the SEND button is no trouble. Finding your checkbook, writing a check, preparing an envelope and finding postage — that's trouble.

We're sympathetic, but we're also pushy. No one *has* to pay for the ANALYTICAL ENGINE; it's shareware. It's yours whenever you want it. But please....if you send us mail that says "I'll subscribe right away," then take your next chance to write that check and mail it. Money pledged is money that we count on.

OVERVIEW OF BUREAUCRATIC PROCESSES

A. VICTORIES

Since the release of ENGINE #1 we've acquired:

- 1) A bank account. This sounds like a simple thing, but it wasn't; paper bureaucracy and electronic communication move at such disparate velocities that we actually held checks made out to the Association before we had a place to put them. This has been fixed and subscription payments are now deposited immediately. Thanks to all those who were patient about their cancelled checks.
- 2) An International Standard Serial Number (ISSN). This registers the ENGINE with the U. S. Library of Congress and the International Serials Data Center in Paris.
- 3) A new Internet mail and news address. This is the visible part of our effort to make our organization less reliant on one person. (It's also partly legalistic nonsense, but never mind that.) Those of you who mail or post to CHAC, please use

cpu@chac.win.net

effective immediately.

- 4) An Internet server request daemon. Not as daunting as it may sound, this clever item automatically provides ENGINE back issues and related useful text files via Internet mail. For

instructions and a list of what's available, send a message to

engine@win.net

with a *subject* line of

help

and the reply will be on its way to you within minutes. (The less wired may request any of these files in hard copy from the El Cerrito mail address.)

B. PROCESSES

We are in process of selecting directors and officers, filing articles and bylaws, and generally complying with the regulations that govern establishment of a California public benefit corporation. Once this is done, we will be able to apply for Federal and state nonprofit status as an educational institution. This will save us money because it has favorable tax implications; it will also mean that donations to CHAC will be tax-deductible to the donors. All this takes time and a discouraging amount of paperwork to do properly, but it's important.

C. FRUSTRATIONS

We had hoped to announce in this issue that we could accept subscription payment by credit card. This is a tough nut to crack. Several MasterCard and VISA providers have muttered that they don't much care to handle subscriptions and that low volume isn't worth their while. At any mention of electronic mail and the Internet, they turn pale. The bottom line is that they avoid dealing with any entity other than a traditional corporation.

To potential ENGINE subscribers who prefer to pay with plastic, and especially non-U. S. subscribers who have trouble paying in dollars, we have to say: Please hold. We want to make payment convenient for everyone, including ourselves, but the right mechanism hasn't appeared yet.

LOGO AND SMALLTALK: Languages that Changed the Rules

by Aaron Alpar, ParcPlace Systems

Relationships between computer languages are often more intimate in their full depth than they appear on the surface. This is especially true of Smalltalk and LOGO, two of the most experimental — and most influential — languages of the last twenty years. They were designed for substantially different purposes and primarily used in very different contexts. Yet a brief review of their history will demonstrate that they have a great idea in common: the tremendous extensibility of the computer as a tool for people.

LOGO: A TEACHING LANGUAGE THAT GREW

Developed by Marvin Minsky and Seymour Papert at MIT, LOGO began as a teaching language, to introduce children to computers. The language that resulted accommodated three central facts:

- * Children perceive work and play as unified;
- * Children care more about results than about process, meaning in this context that as they build something, they want to watch it being built;
- * Children are intolerant of technical limits. When they are told that "The computer cannot do that," their first reaction is either "Yes, it can," or "Why not?"

LOGO's response to these truths was brilliant. First of all, it was a "working language" and at the same time a "playing system" — a language that turned the computer, screen and keyboard into a toy that was deeply absorbing, completely interactive, consistently rewarding and (incidentally) very educational. Through its use of "turtle graphics," a pioneering graphical programming metaphor, and a split screen — half input, half result — it showed *immediate* output of every programming step taken. Finally, LOGO became progressively more complicated as the user's proficiency grew. Someone who went in knowing nothing more than halfway how to type could still achieve enough to provoke a great rush of self-

satisfaction. Then, as curiosity took over, LOGO's simple and uniform syntax and concepts would facilitate exploration of other parts of the system, expanding knowledge and programming skills.

In itself, LOGO was a triumph, and the proof is that it has been very durable. As a teaching tool, it is still widely used in elementary schools. It became a cornerstone of research and literature in artificial intelligence. Its influence on professional programming is inestimable. (The author has spent plenty of time blissfully, and ignorantly, typing thousands of lines of BASIC, assembler, machine code, and Pascal. None of this stimulated any thinking about real possibilities of computer/human design in programming languages. Then I encountered LOGO, which was powerful enough for serious, if slow, application development, and at the same time intuitive enough for children to perform "useful work". I interpreted this as a leap in computer usability that promised to make the technology much more broadly accessible.)

RISING UP AGAINST THE "TYRANNY OF NUMBERS"

LOGO's most profound influence has been as an *ultimatum*, not as a language. It was developed by people who knew computer architecture and programming backwards and forwards, but it was a *product* of thinking about people — of saying "What do people want to do that computers can help them do?" And the people who are most insistent about what they want to do, of course, are children.

LOGO, and after it Smalltalk, freed computers from the tunnel vision of *computing* — from the tyranny of banging numbers together. The users of these languages (who became, in a very real sense, their co-developers) were ten-, twelve-, and fifteen-year-olds who wanted to draw pictures, play music, make movies and create games. Minsky's staff at MIT, and later the working group at Xerox PARC, refused to give the turn-off answer of "The computer can't do that." Instead they plunged headlong into research and brought forth computers that *could* do "that."

Ultimately, this turned the whole hierarchy of computing upside down — from

Hardware	to	User
Software		Software
User		Hardware;

although it took decades for the implication of this to become clear. As Alan Kay said in a recent article,

Hardware is really just software crystallized early....far too often the hardware has been presented as a given and it is up to software designers to make it appear reasonable. (1)

Even for the original LOGO, the hardware was a given; it just happened that this was an unusually minor constraint because the hardware (and hardware support) available to the MIT AI Lab of the day was formidable. Smalltalk leaped the next gap; its "given" was not the software and not the hardware, but a principle. In a 1977 survey article published in *Scientific American*, Alan Kay skewered the notion that the hardware made the rules:

Ideally the personal computer will be designed in such a way that people of all ages and walks of life can mold and channel its power to their own needs.... Although the hardware of the computer is subject to natural laws....the range of simulations the computer can perform is bounded only by the limits of human imagination. (2)

SMALLTALK AND PARC: COMPUTING FOR PEOPLE

It's worth remembering that when this article appeared, very few people had ever seen an Apple II; the IBM PC was four years away, the Apple Macintosh seven. Programmers in a hardware-dominated context were preoccupied with files, compiling, libraries, syntax, railroad diagrams, virtual machines, and the fearful convolutions of installing 16k memory boards.

But the developers at Xerox PARC — Kay, Peter Deutsch, Adele Goldberg, Butler Lampson, and many others — had been crucially influenced by the operational sequence of LOGO:

Turn on computer
Type command

<Enter>

See result.

To the user, there was no distinction between language, environment, and operating system.

Enter the Smalltalk language, in its several versions. While LOGO wasn't the *only* ancestor of Smalltalk — much was also inherited from LISP, Algol and SIMULA — the connection between the two is worth emphasizing because of the tenets they shared:

- * extensibility
- * nominal syntax
- * avoidance of data types
- * concentration on objects
- * persistent connection between action and result
- * minimal demand for background knowledge
- * priority of fun and intuitiveness

Smalltalk-72, the first "complete" version of the language, in several senses began where LOGO had left off. Its extensibility was the extensibility of *objects*; it made the leap from LOGO's visual-object-as-metaphor to genuine and sophisticated object-orientation, building on the great start of turtle graphics to present friendly and familiar "objects" that were also abstract, manipulatable, and the building blocks of programmed systems. A later version, Smalltalk-76, adapted SIMULA's crucial abilities of inheritance and class support, treating classes as objects.

Was it still fun? In 1973 Marion Goldeen, age 12, wrote an object-oriented paint program in Smalltalk; Susan Hamet, age 12, wrote a drawing program much like MacDraw; 15-year-old Bruce Horn wrote a music score capture system. These and many other Palo Alto middle school students provided the "intolerance of technical limits" that spurred development. They were also the first, or nearly the first, children and adolescents to sit down at a computer and have fun.

Here, of course, innovation rested on innovation — because the children, like the PARC scientists they taught, refused to accept the answer "The computer cannot do that."

HARDWARE: COMPUTERS THAT COULD "DO IT"

In step with the development of Smalltalk, corollary hardware issues were being addressed — still with confidential development in a largely closed laboratory. Problems were formidable. Smalltalk was conceived as a completely graphical language and environment; it needed to be run on a bitmapped device. At the time, few computers other than mainframes could compose bitmaps for display, and then crudely. The disparity between hardware and software was huge. *But the hardware had stopped making the rules.*

Chuck Thacker, Bob Metcalfe, Gary Starkweather, Bill English and the other PARC builders responded with a hardware context that combined innovative brilliance with a rare grasp of systems integration. To SLOT, the first laser printer — a wildly modified Xerox high-speed copier — and the Research Character Generator, the first bitmapped font composer, they added the famous Xerox Alto and the Ethernet network.

Integration and daring were the keys that made PARC's "on-line office system" so memorable. The Alto, now often called the "first personal computer," was not the first computer sized and priced for the single user — the DEC PDP-8/S preceded it by seven years; and the first practical LAN architecture was not Ethernet, but the token-ring Pierce Loop developed at Bell Labs in 1971. The Alto was unique as a deliberate exploration of what a personal computer should be like, rather than a small general-purpose machine that accidentally gravitated to personal use.

With these inventions, the inversion of the classic hierarchy was complete. *The user drove Smalltalk; Smalltalk drove the hardware.*

LOOKING BACK: WHAT HAVE WE GAINED?

This is not the place to argue, pro or con, about Xerox Corporation's use and pursuit of these assets. Information distributed on paper had made the company into a billion-dollar establishment and a household word. To put it charitably, the balance between sustaining old technologies and exploiting new ones was not a trivial concern.

Certainly the successes of Smalltalk, of LOGO, and of their underlying metaphors have outlasted the involvement of any single corporation or institution. Graphical interfaces and object orientation now provide a unifying theme over almost the full spectrum of computing — from Microsoft Windows to Macintosh System 7, and on to NeXTStep, X Windows, OSF/Motif, and through language products like ObjectVision, Visual Basic and the various flavors of C++. This pervasiveness of object-orientation and of the graphical interface makes it all the more pleasant to realize that LOGO is taught in schools to this day, and that Smalltalk — having gone through four major and several minor revisions — is a mature, flexible and contemporary language, still commercially available and used worldwide.

The next time you sit in front of a computer that uses a windowing system, and enjoy the convenience and flexibility of the tools it brings you, remember that — through LOGO and Smalltalk — the real work of computing was changed forever by the impatience and the gravity of play.

Notes

- (1) Alan C. Kay, "The Early History of Smalltalk," *ACM SIGPLAN Notices*, March 1993, page 87.
- (2) Alan C. Kay, "Microelectronics and the Personal Computer," *Scientific American*, September 1977.

OF THEE I SING

by Tom E. Ellis

When I first heard about the Computer History Association of California, I was excited by the task of gathering together the objects of our machine inheritance, for all to enjoy. Thinking about seeing a SOL-20 again, or toggling a program into an IMSAI 8080, brought great cheer. Preserving our hardware heritage is an important step in the mapping of our craft. Yet another piece of computer history is as worthy of our attention: the software.

Initially, of course, software tended to be utilitarian in nature. The business of computing was expensive and resources were limited, so programs had to have a significant purpose. But even in that regimented context, some people insisted on the "luxury" of writing code purely for fun, or to try something that had never been done. Much of this software, even if not "serious," included innovative snippets of code that made a change in the way we did things; code that broke the bonds of conventional wisdom and strode bravely forward into new territory. We've all been "explorers" of this type at one time or another, and I hope we always will.

When I was hired by the San Francisco branch of a large nonprofit organization, to assist in the conversion from a 3x5 card system to a donor tracking system on a Honeywell 2020, we had a small problem. The system was to be based on a large master file on tape, with a weekly transaction file for updates. Pretty ordinary stuff.

A master file update was time-consuming, since it involved reading and writing every record in the system. Transactions flooded in daily from units all over the state. The more data we collected, the longer the process became. We wanted to collect our daily transactions and apply them *en masse* to the master file at the end of the week.

We had three tape drives, each as big as a side-by-side refrigerator: Master In, Master Out, Transactions In. Each drive used two long columns of air to pull the tape past the heads. You'd spin the take-up reel and the supply reel in opposite directions to produce a small buckle in the tape, the

vacuum in the air columns would pull the tape down into the chamber, and the reel motors would take up the slack. The whole process was a sight to see and hear.

The problem was that you had the choice of opening a tape drive in "read mode" or "write mode" — not both, or so we thought. Tape devices wrote data as a linear set of blocks, beginning with a header describing the record size, blocking factor and number of blocks in the file, and ending with an EOF record following the last data block. In read mode, when you hit that EOF record, you had few choices. The concept of updating an existing file was unheard of.

I was searching for a way to open the tape drive in read mode, grab the details from the tape header label, proceed to EOF, back up a block, switch into write mode, add blocks to the end of the tape and write a new EOF record. A simple append, right? Not so.

The last data block is hardly ever full. Rarely does the number of records you need to write work out to a multiple of the blocking factor. So I had to back up twice — once for the EOF record and once for the last data block — read that block into the current data buffer, set the internal buffer counters and pointers, back up again and flip the tape drive into write mode. When I finished writing all the new blocks, I'd have to write the new EOF record, rewind to the beginning of the tape, and re-write the tape label with the (now current) block and record counts.

The guys at the data center said it couldn't be done; the positioning of the tape was never meant to be so precise. When you re-wrote the label at the beginning of the tape, they insisted, you'd undoubtedly write data beyond the inter-record gap, stepping on the first data block and ruining the whole thing. And of course there was no way to make sure your last block would obliterate the EOF marker. And what if the EOF marker fell into the inter-record gap? And how would you get around the EOF condition flag that had been triggered? They had a thousand reasons why it wouldn't work!

I became obsessed with the idea. The data center turned me on to a guy that knew tape drives like no one else; I recall that his name was Jimmy. He

had studied the machine code for directing tape drive movement until he could make a drive do just about anything. With his advice, I was able to build a small routine that would fool the machine, and turn a tape drive that had been opened in read mode into a tape drive that was available for output. It was a great day when we successfully appended new data to the existing tape. And never once, during my tenure, did that routine fail to run properly.

WHISTLE WHILE YOU WORK

But as happy as I was to solve my problem that day, I was even more pleased by a little assembly program of Jimmy's that put the tape drives to very creative use. Depending on the movement of tape in the columns, the voltage applied to the record head, and a thousand other conditions, the drives would make a wide variety of buzzing, hissing and humming noises. And the pitch of the sound produced would change, depending on how far down in the column the tape was pulled.

Jimmy had figured out just what series of instructions would produce just which tones, using the tape to vary the length of the column of air in the vacuum chamber. Mind you, there were no commands available to move the tape to a certain depth in the air column — that would have been senseless; the specific tones were a by-product of starts and stops, read/write instructions, and tape movement commands strung together in an ordinarily meaningless series.

So precise was Jimmy's code that he could produce not only the tones he desired, but the rhythm required to reproduce a complete musical number, with percussive sounds thrown in for good measure. The effect was remarkable. People who hardly expected to hear a familiar song coming from a massive tape drive could listen carefully and hear the strains of a Souza march, Mary Had a Little Lamb, or America the Beautiful. We must have coded twenty songs before we were through, and Jimmy gave me a tape of the program — which has to have been the first shareware of my career.

Naturally this became a favorite trick to play on visiting executives. We would log every tape drive to the same physical address, start the program, and soon, from what looked like a busy computer

working hard at its task, out would come Happy Birthday, or God Bless America, or whatever. The computer operator would play dumb, of course, when one of the VIP's would recognize the tune. It never failed to brighten our day.

I'm sure other such programs deserve similar recognition. In the days of paper tape, everyone had a utility that would punch words right onto the tape; people's big treat on their birthday was to get a strip printed with birthday wishes from the computer. The guys in my data center wrote a routine to play baseball, using the print head on a Selectric typewriter as the ball. The pitcher controlled his throw with the keys on one end of the keyboard and the batter hit the return key for a swing. What glorious fun!

Do you know of some innovative use of traditional techniques? Something that breaks the mold? These programs, like the hardware they were born on, are trapped in their own time, gone away with the succession of improvements that have rendered them moot. We've got to catalog these bits of creativity while there are still people who can put them into perspective; and we have to find ways to foster the same kind of creativity in the programming being done now. Today's neat hack is tomorrow's breakthrough algorithm. Today we can watch SimCity™ unfold on our screens; tomorrow I want to watch it roll by from a comfortable seat on my own virtual train!

THE CHARLES BABBAGE INSTITUTE

by Judy E. O'Neill, Associate Director

The Charles Babbage Institute (CBI) is a research institute dedicated to promoting the study of the history of information processing, bringing historical perspective to the study of its impact on society, and preserving documentation relating to the development and application of the computer. An alliance of industrialists, professionals and academicians with a common purpose — to record and study the evolution of the digital computer and modern electronic communications — formed the Institute in the late 1970s. CBI has contributed substantially to the literature in the history of computing through its historical research projects. Through research and archival acquisitions the staff has developed expertise in management of records associated with the computer industry, professional organizations, and individuals. The interaction between archives and historical research is a crucial part of the philosophy of CBI: usable and appropriate records are essential to historical research while the knowledge gained through historical investigation is essential to the development of archival collection strategies.

HISTORICAL RESEARCH

Like the computer industry itself, the discipline of the history of computing is relatively young. Consequently, our knowledge of many areas in the history of computing is incomplete. Since the late 1970s, members of CBI's staff have engaged in significant historical research related to technical developments, industrial growth, technology transfer, and the government's role in technological change. CBI's staff, at times cooperating with colleagues at other institutions, have produced several historical studies which span the period from 1800 to the present.

Recently, CBI's primary effort in historical research was an investigation of the computer activities of the Advanced Research Projects Agency (ARPA). The resulting report, summarizing four years' study of the Information Processing Techniques Office (IPTO) of ARPA, incorporates computing

developments after 1960 into the framework for analysis of the history of computing. IPTO provided substantial research support for the development of computer science and engineering from its founding in 1962 to the mid-1980s. CBI's study is a history of IPTO's origins, development and evolution, and research programs it supported during this period; it includes an analysis of the management of the office, the interactions of its staff with the research and development community, and its military-related mission. The influence of IPTO programs in computer science and engineering is charted through case studies of four significant developments: time-sharing, networking, graphics, and selected areas of artificial intelligence. More generally, the study investigates the growth of computer science programs, various technical developments in computing in the 1960s and 1970s, and the pertinent interaction of government, academia, and industry. The authors are revising the report for publication by the Johns Hopkins University Press.

CBI staff has engaged in studies of the computer industry through ongoing collection of information about companies active in various areas of the computer industry. This material helps to identify computer-related companies, understand where they fit into the larger picture of the computer industry, and see how the industry has changed over time. One project currently underway, "Computers and Commerce," considers the development of Engineering Research Associates, Inc., the Eckert-Mauchly Computer Corporation, and Remington Rand after it acquired each of these companies. This study of the origins of the computer industry shows various strategies for technical development and the interplay with customers in the earliest days of the industry.

One of CBI's new research projects is a history of women in computing. The purpose of the project is to recover the achievements of women in computing and analyze the history of women's participation in the institutions of computing. The project will report its results in scholarly articles about women's roles and contributions.

CBI's new director, Dr. William Aspray, will take the Institute in previously unexplored directions, including a new focus on microcomputing. Initial activities will include recording interviews for future

research, investigating current sources describing its origins and developments, and working with the industry to heighten awareness of and interest in preserving corporate and individual records. The Institute will also strengthen its international focus.

ARCHIVAL COLLECTION

Given the importance of the computer to modern society, its application and development remain relatively undocumented. The CBI archival collection exists because of the advocacy of individuals in business, academia, and government. Without their interest in the preservation of resources for the history of computing, little documentation would be available for research at CBI and other archives.

CBI serves as a clearinghouse for information on all archival collections relating to the history of computing. CBI maintains information about other repositories' holdings, and researchers have access to a file of finding aids on non-CBI collections. The Research Libraries Information Network and the University of Minnesota's LUMINA catalog describe much of CBI's collection in summary form. LUMINA is accessible through the Internet.

The primary components of the archival collection are:

RECORDS — Collections of records at CBI document computer organizations and businesses, computer industry involvement in antitrust and patent litigation, and individuals' records.

PUBLICATIONS — CBI maintains a collection of printed matter including: manuals for specific computers and systems, product literature produced by computer companies, publications related to market analysis in the computer industry, and third-party surveys of computing machinery. CBI holds certain serial publications that offer unique perspectives on computers and computing, such as early microcomputer periodicals, that other research libraries have not retained.

ORAL HISTORY INTERVIEWS — CBI holds a large collection of oral history interviews relating to the history of computing.

PHOTOGRAPHS AND FILM — The photograph collection documents the computer and its use from 1946 through the present. In addition, CBI has a

collection of commercial 16mm film prints on computing, and videos on the history of computing topics and conferences.

GENERAL REFERENCE MATERIALS — CBI's non-circulating library contains a reference collection of works on the history of computing, a selection of books considered to be classics in computing, and reference volumes supporting research of primary materials held by CBI. Biographical, company, and subject files, as well as files on the holdings of other repositories, are also available.

ENCOURAGING RESEARCH AND INTEREST IN THE HISTORY OF INFORMATION PROCESSING

CBI fosters research in, and writing about, the history of information processing. CBI has offered pre-doctoral fellowships in an effort to increase the number of active participants in the history of computing. It currently offers the Adelle and Erwin Tomash Fellowship in the History of Information Processing, a graduate fellowship for pre-doctoral study of the history of information processing. Tomash Fellowships have supported historical studies of magnetic recording, international networks, group decision support systems, and a comparison of United States and British computer industries.

An important part of CBI's work is the development of tools to aid historical research, including oral history interviews, biographical and company information files, and published bibliographies and guides. Examples of published bibliographies and guides are a selective chronology and annotated bibliography of software sources; *Resources for the History of Computing* (the first comprehensive research guide to archival material held by repositories in the U.S. and Canada); *The High-Technology Company: A Historical and Archival Guide*; and *Guide to the Oral History Collection of the Charles Babbage Institute*. CBI has also produced a *Reprint Series in the History of Computing*, in sixteen volumes, which makes scarce material in the history of information processing available to a wider audience of researchers and other interested people.

CBI encourages and facilitates information interchange among people interested in the history

of information processing. Members of CBI's staff maintain a wide-ranging correspondence and participate in many professional activities that serve the historical and archival communities. CBI's educational program includes teaching in the University of Minnesota's Program in the History of Science and Technology and the Program in Management of Technology, sponsoring lectures relating to the history of information processing, and making presentations about both the history of information processing and CBI. Visitors, both national and international, stay at CBI for varying lengths of time conducting research and interacting with the staff. CBI staff responds to hundreds of research requests from diverse groups including participants, historians, sociologists, archivists and records managers, journalists, lawyers, hobbyists, and the general public. CBI sponsors or helps to organize conferences and symposia. These conferences include a technical documentation appraisal workshop (1984), a conference for archivists and historians to discuss the state of the history of computing (1986), *Computing in the 21st Century: A Symposium on Computing and Society, Past and Future* (1986), *Manchester Meetings on the History of Computing* (1988, 1990). Staff members also participate in many other conferences and symposia.

INFORMATION, USE, AND SUPPORT FOR CBI

The *CBI Newsletter*, published quarterly, contains current information about CBI and the history of computing. Through the *Newsletter* the Institute informs the community of work in the field, conferences, publications of interest, and its own activities. It is available free of charge to anyone who wishes to follow developments in the history of information processing.

CBI's archival collection, at its facility in Minneapolis, is open to all researchers. Prospective visitors should consult the archival staff in advance to ensure that relevant materials are available and open to research. The archives staff also attends to the needs of researchers unable to visit the Institute personally. Many requests do not need extended research time and copied documents can be mailed or faxed. CBI is experimenting with other techniques of document delivery, such as Internet

transmission and interlibrary loan of oral history transcripts.

The CBI Friends program accommodates individuals who would like to support our work directly. The Institute encourages inquiries about donating pertinent records, and values requests from individuals, organizations, and businesses to assess the historical value of collected information in all formats, including machine-readable. CBI's collection is built cooperatively with other programs. If another facility is a more appropriate repository for a given set of records, CBI staff will work to match donor and repository.

If you would like to receive our Newsletter, information about our Friends program or other donation programs, or any further information, contact:

Charles Babbage Institute
103 Walter Library
117 Pleasant Street SE
Minneapolis MN 55455
Telephone: 612 624-5050
Fax: 612 624-2841
Internet: cbi@vx.cis.umn.edu or
jeo@maroon.tc.umn.edu

PROGRAMMING THE 1401

Part 2: THE 1401 AND BEYOND

Leo Damarodas

Interviewed by Roger Louis Sinasohn

They must have been expensive to run. You had all those components, the air conditioning. The electricity... Punched cards weren't cheap considering how many you would need. Now, say you wrote a program, ran a test, and found a bug in it, what did you do?

Well, you could approach it in one of two ways. You could go back to the source code, fix the bug in the source code, recompile, which took time, get a new object deck out, and test that. Or say that to fix the bug, you had to add instructions to the program. You could use an instruction called Branch-and-Store that would branch to a location, and store the location of the next instruction — the instruction following the branch...

So it was like calling a subroutine?

Yeah. And you would have it branch into high memory, up into a space the program wasn't using. You would write your additional instruction... maybe what you wanted to do was insert instructions between two existing instructions. So the instruction that you wanted to insert code after would become the first one at the location you were branching to, then you would add your additional instructions, then your return instruction, which would bring you to the location after your branch. By doing that, you could take the object deck that you already had, patch it, and run the test again. You didn't have to wait for a compile.

So that just involved repunching one of the cards from the middle?

Right. What you could do with some card-punchers was feed a card in and duplicate it. In other words, most card machines could punch at a punching station and verify at a reading station. You could take your card that you wanted to duplicate, advance it to the reading station, which would pull in a blank card behind it, then you would duplicate column by column until you got to the column you wanted to change, make

whatever changes you had in mind, and then dupe the rest of the card.

And then just switch them?

Yeah, put it in the place of the other card. If a keypunch wasn't available, and you really wanted to work hard, the lead operator had, just for fun, a manual card punch... you could feed a card into the thing, set up which positions you wanted to punch, pull a lever, move the card a column, set up where the next punches go, pull a lever...

Did the IBM 1401 use the ASCII system, or was it EBCDIC, or did it have...?

The memory locations were set up to look just like an 80 column card — plus two more bits. So there was a bit that represented zero through one, eleven and twelve, which were the zone overpunches, and then there were what was known as the record mark and the word mark. They were two other memory... well, what we would consider bits.

They weren't called that, but memory looked like a punched card with two additional positions. Each memory location was like a column on a card. The addressing structure used three positions to represent 1K. But then you had overpunches, and I know the overpunches were used on the left and the right... there's a combination of four overpunches, so you can... Is five enough to get up to sixteen? Yeah. If you had no overpunches, it would be zero through a thousand. or zero through nine-nine-nine. I can't remember exactly what the scheme was, but they used the overpunches to make up the difference.

Now, being a programmer on a 1401, in a typical day, how much of your time was spent working directly with the computer?

That would depend on what state your project was in. If you were in a coding stage, you wouldn't be near the computer at all. See, now, when you're writing programs, you can write part of it on the fly, because you're interactive, and your compiler's so fast, and everything's so easy; you can have access to the machine the whole time, write small parts of that program, and recompile, and run it. Well, you could do this with a 1401 if you had access to the machine all day long. But where I was working with a 1410, there were eight programmers, two analysts, a lead machine operator, and two shift operators. The two shift operators

were running production jobs on the machine all day long. So the machine wasn't available to programmers during regular hours, except sometimes by prior arrangement, or maybe during a little slack space, when the operators were waiting for data to arrive for a run. Then you've got 8 programmers and only one programmer can use the machine at a time. So what we wound up doing was just about coding a whole system before we loaded it on the machine for the first time. We didn't punch up our own programs, either; we'd code them, they would go off to keypunch, come back in card form, and then we would come in at midnight and run our compiles. Or we could leave it for the night operator to compile it until the compiles were clean. You did a lot of desk checking.

Desk checking?

You would go over your code looking for syntactical errors, because it took so long for a program to compile. You had to spend time rereading your code, checking for syntax, weeding out as many errors as you could yourself. You couldn't spend a whole lot of time compiling and recompiling, because the machine wasn't available. It wasn't a resource of one machine sitting in a room out of sight, and everybody's hooked into it, and using it. That didn't even happen with the 360 and the 370; really, in my experience, not until the mid-70's when the HP came out.

Did you ever run into any non-business programming? Was it possible to do games on the 1401?

About the only things like that were... you could run calendars with pictures of Snoopy or Santa Claus, do banners. But other than that I don't remember any games, until the 360. I'm not even sure there were games on the 360. But now that I think of it, there was one thing on the 1410 that was kind of like a game. This tremendous amount of circuitry developed radiation of the type that the radio could pick up. So we could get these programs that didn't have any function, but if you loaded one and ran it, took a radio and you set it up on top of the memory unit and tuned it between stations, the radio would play a song. It would be playing a song like on an organ or a violin or something; the program would exercise the memory circuits in a way that would generate radiation for the radio to pick up and translate into

music. There were a whole series of programs that could play Happy Birthday, or Jingle Bells, or a number of other songs.

Is there anything about the way you worked, or anything about the 1401, that you miss in the modern machines?

Not a thing. (Laughs) Other than making that music. But I don't miss the cards, I don't miss having to keypunch... you know, write code on a piece of paper, and then sit down at the keypunch and punch it myself, or have somebody else translate it into punched cards, and loading the cards into the machine. Dropping decks of punched cards and having to sort them back into order. No, I don't miss any of that stuff.

When did you make the transition from batch to interactive programming?

I didn't really see any interactive programming until '78 when I started working on the HP [3000]. So it was just the last 15 years. And actually, I started programming in '65 but I took about a four-year break from 1970 through....mid '74. I burned out. I wiped myself out.

So what did you do in that interim period?

Took a year off. And then I recapped tires. Once I got sufficiently bored with that I said, I'm wasting my time doing this, I'm going to try programming again. That was '74, and I've been at it... the longest break between jobs I think has been about a month.

So how do you keep going now, then? How do you avoid burning out?

I don't do it when I'm not at work. I burned out because I was programming 24 hours a day, seven days a week, whether I was at work or not, I was programming. I mean, I was always thinking about work. I love to program. I love to play with computers. But you can't do it all the time.

So, since that break, you've been going nearly twenty years, probably close to twenty-five years total, and you still like it?

Yes. I found my niche a long time ago, and I'm happy with it. At one place, I was actually told to find myself another job, another company to work for, because I didn't want to go into management. I was offered the job of programming manager, and

I said no thanks. I even avoided being a project leader whenever I could. I didn't want to deal with managing people, be responsible for their work, see that they got it done. That's not fun to me. Programming's fun. I'm totally amazed that I make what I make for doing what I do, that people are paying me to have a good time.

You burned out earlier because you were programming all the time. Was that because of the non-interactive-ness of the system then, as opposed to now, where if you're not sitting in front of the computer, you're not really programming?

No, because when I came back into programming, it was still primarily a batch environment. It was an IBM 370 then and it was faster, it could handle a bigger workload. But it was still the type of environment where you sat down and did flowcharting and did your coding on paper. Compiles had become fast enough that, instead of a big program with multiple overlays....like a 60K program taking three hours to compile, a 60K program [on the 370] might take a minute and a half. I mean, now you can compile a 60K program in seconds! But it was still the same technique of doing programming. It was just that I learned how to put it away. It's like when you're having a good time skiing, you don't ski through dinner and ski while you're sleeping. When you're at the ski resort and you're sitting at the bar, you're not skiing anymore. And I had to learn how to do that with programming. Okay, you're enjoying yourself, it's time to stop and do something else, give yourself a chance to rest. I mean, I was going into work on week-ends, I would be sitting having dinner with my family, and I would get up and go off into the living room, sit down, and start writing code because the solution to a problem had suddenly come into my head. I don't do that anymore. You can't, and survive.

What do you foresee for the future of the computer industry?

I've never been good at predicting that. That's the reason I'm still an applications programmer.

As technology gets better, and faster and smaller, what excites you the most about the advances since the 1401, and about the possibilities looking ahead?

When I started programming, it all was done in a shop environment. You went into an office and

you sat down, with a bunch of other people, because the machinery that you worked with had to be in a specific place. You can carry around today, in your briefcase, a computer that outperforms an IBM 370. There's more power, more speed, and you're carrying it in an eight, six, four-pound package. And the stuff is going to get smaller. It has yet to reach any size limit as far as circuitry is concerned. Computers haven't reached the processor density or circuitry density of the human brain or any animal brain. The only thing that's going to stay about at its current size is the physical human interface. I mean, there's no way you can get a screen too much smaller and still have it intelligible to the human eye — if you're going to carry thousands of characters on it. Keyboards can't get much smaller and still fit our fingers. What'll get smaller is the circuitry and the storage. I eventually see the disk drive going away. There's no reason to have any mechanical devices, to have magnetic devices with moving parts for storing information. With flash memory, you can store information solid-state and not need any batteries — you set the information up inside the chips and it stays there. As far as I'm concerned, it's magic. But that stuff is going to keep on getting smaller. Somebody someday might figure out how the gray matter between our ears works, and start building circuitry that works that way, and then maybe computer programmers will be in trouble.

One last question. The IBM 1401 that you started out with was new at the dawn of the transistor. Now what about the last vacuum tube?

Oh, the one-eyed monster? That's gotta go. The LCD, or some future version of it, has to replace the CRT eventually. The CRT is so bulky, now that there's such an appetite for bigger screens.... If you have a twenty-inch screen it requires, I don't know, a foot and a half depth? That's a lot of desk space, shipping space, automobile space. But an LCD display is an inch thick, no matter how big it gets. So the last vacuum tube will eventually disappear. Sooner than later, I hope.

BOOK REVIEW

THE DREAM MACHINE

Jon Palfreman and Doron Swade

BBC Books, 1991

208 pp., £16.95

Reviewed by Kip Crosby

This book's subtitle, "Exploring the Computer Age," is telling. Survey volumes about young sciences tend toward adolescent awkwardness, and this one is no exception, but at their best they also display the vigor and excitement that derive from an honestly new subject. In its eager narrative, majestic vistas, zigzag progress and breathless grabs for overview, **THE DREAM MACHINE** is truly an exploration. Later treatments will be smoother and more settled but a lot less immediate.

The ground plan here is ambitious: A reasonably rigorous history of computing from the abacus to artificial intelligence and beyond. The subject threatens to sprawl, as always, but is well contained here by the focus implicit in the title — the *computing machine* in its broad historical and philosophical aspect. Of the book's 208 pages, over 150 treat the century bracketed by Hollerith's tabulation experiments in the late 1880s and the great AT&T system crash of 1989. A more descriptive subtitle might have been "The Computer and its Development."

To their credit, the authors recognize that inventors are as fascinating as inventions, and give us a steady parade of portraits — not only of true giants like Hollerith, Babbage, von Neumann, Eckert, Mauchly, Minsky and Wozniak, but lesser-known (and not less interesting) figures such as Maurice Wilkes, Tommy Flowers and Doug Engelbart. In this very European book, there's a much more thorough treatment of Alan Turing's life and work than there might be in an American work on the same subject; Konrad Zuse too is given the space he deserves. As a survey history, **THE DREAM MACHINE** definitely passes the test of inclusiveness — even if it's odd to find no mention of John Atanasoff, or to see the developers of VisiCalc described as "a Harvard Business School student and his friend" both left nameless. The book is less sure in its sense of proportion and, for every really thorough treatment of a subject, too many others are half-

column sound bites. Still, there's nothing wrong with a book that leaves you hungry for more. (Incidentally, a lot of interesting material is buried in the notes at the end.)

Any book derived from a TV mini-series is a chancy proposition, but **THE DREAM MACHINE** exploits most of the inherent advantage while it avoids all but a few of the pitfalls. The copious photographs came from a really good archive. The research is careful. The writing and pacing are in the BBC's best style — a big vocabulary and short, well-knitted sentences — making this book an engrossing straight-through read without a hint of condescension. It's a refreshing collection of words and pictures that, finally, conveys more than the hours of TV it descended from.

Unfortunately, poor design almost wrecks the good impression made by the research and writing. The coffee-table format tries to match the impressiveness of the parent series, but only adds gratuitous blank space and curious typography that threatens to swamp a valiant little book. The aggressively postmodern graphics are distracting at best — this reviewer likes his page layouts constructed, not deconstructed. Sometimes, as in the discussion of artificial intelligence, the artwork is so intrusive that it makes the book literally hard to read. And the relationships of photographs and their captions are arbitrary, to put it mildly.

Nonetheless, **THE DREAM MACHINE** manages to be comprehensive, intriguing and satisfying. It's accessible to the interested but "nontechnical" reader, with many amusing or startling facts included to make up for occasional lack of depth. A real survey of the field wants a thicker book, but as a quick, well-illustrated guided tour, **THE DREAM MACHINE** is definitely worth your time and money.

ACQUISITIONS

IMSAI 8080

Funds from subscriptions to the ANALYTICAL ENGINE were used to purchase a pristine, well-equipped IMSAI 8080 from Winston Gayler of Cupertino, CA.

Sometimes called the first micro produced in California — an honor that actually belongs to the Intel Intellec series — the IMSAI is secure in computer history as the world's first micro clone. In 1975 the Altair 8800, made by MITS in Albuquerque, NM, fired the imaginations of electronic hobbyists all over the country; but, bowled over by demand, MITS found it almost impossible to keep up with orders. The Altair was also difficult to build and finicky in its operation.

Under the leadership of Bill Millard — later the founder of ComputerLand — IMS Associates in San Leandro, CA, moved aggressively to fill the gap. Engineers Joseph Killian and Bruce Van Natta produced a handsome microcomputer based on the Intel 8080 CPU and the "Altair bus" (later called the "S-100 bus") but with cleaner design and more robust construction than the original Altair. The first IMSAI 8080 kits were shipped from the San Leandro factory on December 16, 1975; the IMSAI, like many later clones, went on to outsell the machine it had been inspired by.

CHAC's new IMSAI was purchased from Ithaca Audio in Ithaca, NY, in March 1978. Shortly thereafter, Gayler equipped it with a Bytesaver EPROM board and a Godbout Econoram II board. Fifteen years later, he shipped it to CHAC's El Cerrito office and included a full complement of accessories, program listings, all the original paperwork, IMSAI and Intel manuals, IMSAI and Byte Shop catalogs, complete handwritten instructions, and even full-page IMSAI ads clipped from contemporary electronics magazines! As you might expect, this beauty is in literally flawless condition; and when we set it up on a worktable and plugged it into a UPS, it booted at the first flip of the red RUN switch. The dizzying elation was like a ride in a time machine.

Our micro collection has a new crown jewel. Thanks to Win Gayler for years of fanatical care, to Tom Ellis for knowledgeable unpack and setup, and to our subscribers — of course — for making the purchase possible.

[For the story of IMSAI at full length, interested readers are referred to:

Once Upon a Time in ComputerLand

Jonathan Littman

Price Stern Sloan, 1987.]

LETTERS

IBM 1401: *Amplifications and Corrections*

HISTORICAL ACCURACY

◆ In the interest of accuracy, I hope I am not the only one left on the planet who remembers that the IBM 1401 was *NOT* a vacuum tube machine. The 1401 was all solid state with core memory. It was very compact, built in modular bays and would probably be considered the first "departmental" computer because of its reasonably compact size and modest power and cooling requirements.

I remember them being rolled into regular office environments, with good air conditioning of course, and fired up and operated, without false floors and special cooling and power wiring.

Perhaps Leo is referring to the IBM 650 which was a small vacuum tube computer, but it had a drum memory, or the IBM 604 which was really a calculator in the sense that programming was done on a plug board like the tab machines, but it was all vacuum tubes including whatever memory it had.

The rest of his reminiscence brought back fond memories of my first programming work on the same machine in 1963. I wish I would have kept all of the free documentation about the machine that IBM gave us during class. It would make for a much more accurate discussion.

— from Dean Billing, UC Davis, via Internet

[Blame that one on a rotten desk check! The closest source for correct information on this point, Cortada's *Historical Dictionary of Data Processing*, was within arm's reach and yet not consulted. We will pay closer attention to our verify pass from

now on — and we're very grateful to have an audience that can be relied on to trap errors. — Editors]

1401 AS SLAVE PROCESSOR

➡ The first 3 cards of the program deck were a "bootstrap" that set up the machine so it would load the rest of the program cards from the deck.

A large utility where I worked used them for substantial processing of property and cost information. When files or processing were significantly larger, a 7074 was used.... The 7074 was also used for billing, and had "slave" 1401's to read the output tapes and print bills. Billing input was from paper tape! and keyed-in data.

— from Harriet L. Coleman via Internet

1401: RECOLLECTIONS AND CORRECTIONS

(lines in italics are from Damarodas part 1)

....all the data. You couldn't go back and forth between overlays because the programs had to run from card decks.

➡ 1401s often had tape drives on which programs or data could be stored. While their primary use was business DP, they were also frequently used as off-line I/O processors for larger machines (putting jobs on tape for batch input and printing output).

....Data resided on disk, but not programs.

➡ Again, anything could be on disk. Disk drives were rare — I only recall using one. I think it was called the 1405 — wrote a file sort using it. Disk drives became more common on the 1440, which followed the 1401.

....What happened if you had a deck that's data to be input, and you have a deck that's a program — what happens if you mix them up? That is, you put the data in as if it were a program?

➡ You put the data deck, if any, after the program deck. If they were mixed up, it would blow up.

....So a small program might fit on a single card?

➡ None that did meaningful work — most instructions required 1 or 2 addresses. Other instructions had an optional op-code modifier. The instructions were variable length, the start of each

being indicated by a seventh-bit (the "wordmark") being set. Data fields were also delimited with word marks. Addresses could also be modified by index registers, but there was no indirect addressing. (That came on the 1410, the 1401's big brother, which also had asynchronous channels, interrupts, and more memory). [J. Philip Miller's comment: "I sure wrote a lot of 'single card' programs. What was even better was that there were toggles and push buttons on the console and you could enter your programs without a card reader at all!"]

....Only about a year, and I think the machine I worked on was actually a 1410. 1410's were the ones that had disk drives.

➡ He may be thinking of the 1440 here — they were much more common than 1410s, and had disk drives....I worked on an experimental system where we had a 7094 and 1410 sharing the same 1301 [Winchester disk subsystem,] and working with engineers from both [IBM Data Processing and Data Systems] divisions to see what would happen under various conditions was a challenge. In our system, the 1410 did I/O for the 94, feeding jobs to IBSYS — the 1410 was multiprogrammed and IBSYS took its input from the 1301 and left its output there. IBM never made a product of the system, but it was a lot quicker and more flexible than preparing tapes and printing with a 1401, which was the usual way things were done at the time.

....While I am at it — there were other tools — one was RPG, and another was called FARGO; a quick way to convert a unit record application to the 1401 — describe the card layout, report layout, totals, etc., and FARGO did the rest. (I think the A stood for "automatic" — 1401 Automatic Report Generator — but what about the "O")?

— from Laurence I. Press, via Internet

MORE ON AUTOCODER

➡ Autocoder was an assembler, not "kind of" an assembler. Its big advance over the standard assembler was that it was free form and, at least in later years [after 1960,] also included macros. Thus it was, for those days, a "super assembler" when compared to the standard, fixed field assembler [SPS or *Symbolic Programming System*]

— from J. Philip Miller, via Internet

1401 FORTRAN COMPILER

◆ The 1401 FORTRAN Compiler was an "in situ" system which used 64 (yes, 64!!!) passes to compile the program. The trick was that someone (and I don't know who, I would love to see the proof sometime) worked out that the object code for a compiled program was not greater than that for the source code (this was the early 1960's, remember!). So they replaced the source code in core with the object code! Each pass (pretty well) dealt with one aspect of the language. The details are in the appendix to my first edition of "The Anatomy of a Compiler", 1967.

— from John A. N. Lee, via Internet

MORE GLITCHES

◆ There were indeed more than a few errors in that [July] newsletter!

1) The IBM 1401 was not a vacuum tube machine! It was transistorized.

[We're not likely to forget *that* again.... — Editors]

2) Pong was not a computer game! It was a video game implemented with TTL logic, with no computer.

3) Spacewar wasn't a video game! Video encoding of data was not used to paint the image on the face of the CRT. Instead, the software drove a pair of ADC's to handle deflection in the X and Y directions, and the software painted each dot on the screen. The resolution was far better than standard video, but the number of dots that could be painted without causing flicker was limited.

(By the way, I have played Spacewar myself on a DDP 224 computer at the University of Michigan — that machine was a nice 24-bit minicomputer, and their version of Spacewar used two joysticks on the graphics display.)

In any case, the folks at CHAC clearly have their hearts in the right place, and the kinds of errors [you] made in [you]r newsletter only serve to prove the point of [you]r opening editorial! History is indeed being lost and forgotten at an impressive rate!

— from Doug Jones, via Internet

[From the strict technical standpoint, Pong is absolutely not a computer game. In our original statement we assessed it as collectors rather than as engineers; most collectors, it seems, would find that distinction less critical. And we're still astounded that there are only nine left.

Your comment about the PDP-1's display logic raises another equally salient point. If the rate of loss of *history* is impressive, what about the potential rate of loss of *working knowledge*? As the Babbage Institute has correctly maintained for years, there's a lot of scope here for recorded interviews.

— Editors]

1401s I HAVE KNOWN

◆ I had a summer job operating & programming 1401 and 7070 in 1962 and '63. The 1401 had tape, card reader/punch, and printer. It was used for some card manipulation stuff, but mostly as card-to-tape and tape-to-printer for the 7070. Our machine had a minimum configuration, 1400 bytes of core.

The 7070 and 1401 both had assemblers called Autocoder. Our teeny 1401, though, was too small to use Autocoder, so we programmed it in SPS, the Symbolic Programming System, a much less fancy assembler, fixed card fields, no macros. You loaded the reader with the assembler deck, your source, and pass 2 of the assembler, and hit LOAD. SPS read in, read your program & punched an intermediate file, and stopped. You took the intermediate file from the output hopper and put it behind pass 2, and hit START to complete assembly.

The next summer I got a job at a company that had two BIG 1401s, 12K each. On these we could run Autocoder, and use a primitive debugger called Autotest, which let you patch programs, compare core to the object deck, and even set breakpoints. These machines did payroll, wrote dividend checks, stuff like that. Such applications were written in 1401 COBOL. I was an Autocoder jock though, had my own private library of tape error recovery code, merged in the center pocket, hot stuff.

A few years later at MIT Project MAC I was able to use my 1401 knowledge; MAC had a 4K 1401 used as reader/print for the 7094, and we had a need to print in mixed case. We designed a solution which produced 7094 tapes in ASCII and printed

them using a special RPQ on the 1401 which used the word marks as a 7th bit in the print band. Squeezing the printer management & character translation into 4K was tough; I had constants in between the index registers.

Many 1401 programmers learned the machine by going to IBM school. I didn't, but I have here an exam from the Basic 1401 Programming course: Part 1 is on general stuff, parts of the machine, etc.; part 2 covers flow charting, bits in the word; part 3 asks what happens when specific instructions are executed. All multiple choice.

The main thing to remember when programming the 1401 was that IF THE B-FIELD IS LONGER THAN THE A-FIELD, AN UNEQUAL COMPARE WILL RESULT (if you were lucky enough to have a 1401 with the arithmetic compare function.) Make this your mantra.

— from Tom Van Vleck, via Internet

A COMMON BOND: *Welcomes for our Association*

FROM THE SMITHSONIAN INSTITUTION

◆ I am glad that you are starting an organization devoted to computer history on the West Coast. Right now there is activity at the Smithsonian, in Boston, in Bozeman, MT, Minneapolis, MN, and of course the *Annals*....out of Virginia. But nothing, until you came along, in California. I was very disappointed to hear that the Silicon Valley Information Center at the San Jose Public Library was closed due to lack of funding — as was, incredibly, the Foothills Museum of Electronics. Good luck and feel free to use the History of Computing Bitnet List as a forum whenever you choose.

P.S.: The Smithsonian is trying hard to preserve artifacts. We have a Xerox Alto, an Apple I, an Altair & other S-100 PCs, a CRAY-1, a PDP-8, as well as lots of stuff from the earlier days — going back to the 17th Century but including the ENIAC, UNIVAC, and Harvard Mark I. Be on the lookout for a book by myself and another Smithsonian curator about our holdings.

— from Paul Ceruzzi, Smithsonian Institution, via Internet

[Thank you very much for your encouragement and cooperation, which gives a big boost to (at this point) a very small organization.

The idea of "an organization devoted to computer history on the West Coast" is proving popular to an extent that we also find very gratifying. It's early days yet, but the ANALYTICAL ENGINE currently has more paying subscribers *outside* than *inside* California — which is startling and certainly not what we expected. The feeling that we have a national base, albeit a tiny one, is creating a lot of enthusiasm among us.

It's important to note that the foundation that underwrote the Foothills Museum is still very much alive, so we might hope for some cooperation there.

Certainly we're "on the lookout" for your book, and good luck with it....especially if you're still winding up the writing. — Editors]

FROM THE BABBAGE INSTITUTE

◆ We read with great interest your recent newsletter, *The Analytical Engine*. Your enthusiasm for the history of computing is obvious and encouraging to those of us who have been working in this field....

Fifteen years ago a group of people gathered with concerns, and plans, remarkably similar to yours. They formed the Charles Babbage Institute and the Charles Babbage Foundation. For the past fifteen years CBI has been working to preserve and understand the history of computing and information processing....

CBI has always recognized that the field is too big for the small number of repositories collecting records. The archivist keeps in close contact with other professionals in the field, and encourages other

archivists to collect in this area.... It is great to see people actively interested in the history of computing but none of us need to work in isolation. By cooperation, we can all be part of an active movement towards preserving and understanding the history of computing.

— from Judy E. O'Neill, Charles Babbage Institute
[excerpted]

[Thank you for your kind letter and elaborate package of materials. The CBI's resource listings make it clear that your Institute has done formidable work especially in collecting oral history, documentation, and the personal papers of noted scientists and industry executives. Truly, we are not all "working alone," and in great measure we have the CBI to thank for that.

Thank you also for pointing out that the quarterly *CBI Newsletter* is available free of charge. We encourage ENGINE readers to request it from the Institute at the address on page 31.]

WHAT EVER BECAME OF THE OTRONA?

➤ It has always amazed me that nothing like CHAC existed before. In fact, I've been wandering the Net for....months now trying to find a newsgroup for defunct computers in general or....home computers in particular.

As one schooled in history, I was quite disappointed to see so much history in software and hardware, primarily, being thrown out. In fact, when Computer Shopper used to carry "classic" computer articles, I wrote to them explaining that if they were willing to foot the bill, I would write a series of "what-ever-became-of-?" articles. I've always wondered, for example, about the development history of computers like Mattel's Aquarius and Exidy's Sorcerer. Some of these companies (and computers) flashed across the computing heavens like a meteor. I and, I'm sure, others would like to hear their stories. I hope CHAC and the ANALYTICAL ENGINE can pursue something like this.

I will be forwarding my subscription in the next week or so. Please consider me a member of your worthy group.

— from Allan Hamill, via Internet

[You of all people will be delighted to know that those Computer Shopper articles have been expanded, spliced, smoothed out and reformatted into:

Stan Veit's History of the Personal Computer
304 pp., paper, WorldComm, 1992, \$19.95

available from: The Computer Museum, address on page 31.

As for "pursuing something like" stories of dimly remembered hardware, you bet! No computer too small or too large. (Refer to the "Desperate Plea for Storage" on page 5.) And thanks for the sub.
— Editors]

STARTING FROM SCRATCH

➤ I just read a copy of the ANALYTICAL ENGINE, which some kind soul e-mailed me, andyour philosophy about the value of historic technology is right in line with my own.... I am a sort of amateur computer historian who is trying to collect enough info to start thinking about writing a comprehensive work on computer history.

Unfortunately, [since] the subject area is so vast, getting the kind of details I want (deep-down specifics and info) will take a lot of time. Anyway, thanks again, and let me know what is available!

— from Don Congdon, via DELPHI

[Don, if you manage to assemble a "comprehensive work on computer history," we'll be first in line to buy a copy — or a set! Anybody thinking comprehensively about computer history, meanwhile, should go in search of

Historical Dictionary of Data Processing
3 volumes: TECHNOLOGY, ORGANIZATIONS, BIOGRAPHIES
James Cortada
Greenwood Press, Westport, CT

Unfortunately these are expensive at US\$265 for all three, but they're the closest thing to a complete treatment that we've uncovered so far. Thanks for the good word. — Editors]

GREETINGS FROM IOWA

➤ How about forming a Computer History Association of Iowa, sister organization to CHAC? I'm interested in these historic systems and would

like to learn more. A local organization would be great, so we can get together and socialize. Having a larger geographic distribution will make things more interesting too. That is, unless there's some local organization that fits the bill that I'm missing.

— from Jeffrey C. Ollie, University of Iowa, via Internet

[Well, go for it! We look forward to encouraging, assisting and collaborating with CHA's in every state in the Union.

Anybody who wants to start a CHA in their state, please, write or post to us while the idea is still a lit triode above your head. Our hard-won knowledge of protocols, processes and priorities can save you major agony. — Editors]

MUSEUM PLANS IN THE NORTHWEST

➡ I am an avid collector of old iron and interested in computer history. I would like (eventually) to form a group like CHAC and a museum in the Northwest. I know of several others in this area who are collecting larger or smaller systems (I collect large ones) and who might be interested. In the meantime, I would like to join CHAC and offer what help I can....

— from Paul Pierce via Internet

[Our hat is off to anyone who "collects large ones" while we're still frantic over what to do with our small ones! If you decide to form a CHA, do get in touch with us first for strategic discussion; meanwhile, if by "Northwest" you mean Washington/Oregon, we suggest you contact David McGlone at the address on page 32. — Editors]

"THE JOB NEEDS TO BE DONE"

➡It's clear that [CHAC's] goals are very much in keeping with the goals of many of the readers who follow these newsgroups.

As near as I can figure, they're the first general membership association devoted to historic preservation of computers. Thus, they may be able to provide the kind of hacker-centered orientation that is lacking in [some other institutions]....

These people are doing the right thing, with essentially the same goals that led to the formation of *alt.sys.pdp8*, but with a broader charter and a

more formal organization. They clearly have a regional focus, but at the same time, the job they're talking about needs to be done nationally and internationally.

I feel that we preservationists should support them actively, either by joining their organization (particularly, those of us in the Bay Area), or by referring California hardware and documentation rescue work to them. They've picked up a significant membership outside of California, but my long-term hope is that we can get similar local organizations running on the east coast and in the midwest. We've got the critical mass in both places, but here in the midwest, we may not have the critical density....

— from Doug Jones, via Internet

[Doug, thank you for this glowing tribute! and, beyond that, for all the critical support offered to us through *alt.sys.pdp8*.

We wouldn't change a word of what you've written (why would we need to?) but just a couple of comments:

1) If indeed we are "the first general membership association devoted to historic preservation of computers," we're not surprised. We started it because *we also* couldn't find an existing one. But it's our hope, too, that someday soon there'll be "similar local organizations" in other parts of the country. That would ultimately provide the strong, broad grassroots base that would facilitate the organization of a true Computer History Association of America.

Now, before we started the CHAC in April, we toyed with the notion of *founding* the CHA of America. After the CHAC's first four months, we're glad we didn't try. First of all, we have to get *some* sleep. Secondly, even with our avowedly strict focus on California, we've had (welcome!) correspondence not only from (e. g.) Iowa, Pennsylvania, Massachusetts, Florida.... but (e. g.) Finland, Denmark, Sweden, Germany, the UK, and Saudi Arabia. The Internet is astounding, which is a separate topic, but none the less true. And third, the hardware....! (Details on page 19.)

2) I'm not sure we're "hacker-centered" but we're hacker-propelled. To be "hacker-centered" would, I think, risk being exclusive in a negative sense and

tend to push aside certain real pillars of computer history in California; for instance, the Bank of America's pioneering of electronic check processing with ERMA, which wasn't especially hackish but it sure is part of what we're about! On the other hand, it's true that a lot of California's (and especially Silicon Valley's) computer history is inseparable from a certain...effervescent ad-hockery, ebullience, zaniness. And we'd like to keep that as a flavor.

3) "the job they're talking about needs to be done nationally and internationally." Absolutely true! At the same time it's important to know that it *is* being *begun*, nationally and internationally. In our brief months we've heard from many fine people and organizations... and one of the most important things we've accomplished is to get some of *them* talking to *each other*. The work before us is big enough that, as we confront it, we can be bolstered by the knowledge that we're not alone in this. It makes for a tremendous sense of friendship and community.

4) Finally, I take your point about "critical mass and critical density." If you want to know about critical mass, just count the interesting micros we have stacked up in boxes! But I think any doubts about critical density of interest are alleviated, if not wiped out, by the speed and rate of diffusion of net.communication. (That, incidentally, is why we're currently composing an RFD of a newsgroup.) Computer history *is* and *is not* a local pastime — or, so to speak, "Talk locally, post globally." We have before us the means to overcome isolation completely.

5) Yes, we will try to take on rescue work. Our current capacity for it is very limited, but we at least want to know about it.

6) We need:

Money because we're getting bigger faster than we ever thought we would. Right now, and for the foreseeable future, CHAC is all-volunteer. But by the time we're working on some of what we outlined in the July ENGINE (like the museum,) we will be facing significant administrative expense and at least part-time salaries.

Space (storage) because people are offering us hardware that we don't want to see junked.

Members, first, because without members we're nothin'. CHAC is *about* computers, but also *about* the people who create them, and *by and for* the people who work with them and respect them. Second, because only with a substantial membership can we confront corporations and ask them for real support. Fundraising, too, is about people.

A couple of days ago we got net.mail from someone that began "Is this legit? If so I think it's wonderful!" Well, it's wonderful. We know that already. We know that it's real. We know, above all, that it's needed. It can also be "legit" if — and *only* if — enough people are willing to join us in what we're trying to do. Thanks again. — Editors]

CONSIDERATIONS OF HISTORY AND RELIABILITY

► I truly believe in the intended cause the CHAC people are attempting, and I really hope they succeed, but unfortunately, they seem to be victims of their own predictions. In order for such an organization to succeed, it must **flawlessly** pursue its subject matter. Towards that end, I would suggest that [USENET] groups such as *alt.sys.pdp8* act as super-proof-readers of such material, i.e., post draft copies here first and allow us pedantic pdp-8'ers and other interested hanger-outers to tear apart their articles for technical accuracy, all in the interest of upping the quality of the product, etc.

As it stands now, it's a little too revisional for me, although admittedly only a little. Forgive me for saying it, but from my vantage point, someone who happened upon a 1401 in 1965 ain't a pioneer! This is only marginally before my time, and I already knew enough to know the mistakes pointed out elsewhere, and I consider myself "second generation", not a true pioneer in the industry, etc.

....If those who are committed to wanting to preserve history can't quite get it together, we can't possibly succeed! We must help these folks get their act together completely. All of us have a part to play to get it correct, before the entire history of computing is personally attributed to Bill Gates, now sometimes erroneously attributed to as the author of the original BASIC among other things!

— from Charles Lasner, via Internet

[We're delighted to have super-proofers — that's a lot of why we're on USENET to begin with. Tear away! (Not that people haven't.) On the other hand, I will concede (being one) that editors are human, and no less fallible in that capacity than in any other; flawless pursuit of subject matter is perennially desired and less often attained. If we publish articles that need no correction, great. If our articles go out to readers and somebody finds a bug, then, to publish the article *and* the correction still does a service to the community and enriches the public record. This issue of the ENGINE demonstrates that we do publish corrections, even at a length that placates the most pedantic proofer.

I don't quite get the intent of the word 'revisional', but certainly "someone who happened upon a 1401 in 1965 ain't a pioneer." IBM announced the 1401 in October 1959 and, in fact, projected in an internal report that the end of the useful life of the series would occur in 1965. But while pioneering work is important to CHAC, and preserving the record of it is an important part of our mandate, it ain't the whole story, either. If somebody can make an interesting, illuminating point about computer use in California, at the appropriate length, then it doesn't matter whether they were the first to gain experience with a system or (as might be equally intriguing) the last.

As for Mister Gates, I wouldn't worry about him getting *all* the credit. He was born in the year that the first IBM 704 was delivered, so vacuum-tube computing is probably safe from the attribution....]

RESOURCE FOR THOSE WORKING WITH CP/M

➔ I....hope to found a museum of CP/M and Z-System computers, and to that end I have been collecting the computers, software, manuals, newsletters, magazines, etc. I am still compiling an inventory of....100-200 computers, and many file cabinets full of diskettes and printed material.

From time to time, no doubt, you get calls or letters from people who have CP/M computers but do not have the manuals, software, or both. If you cannot help them, I hope you will refer them to me. If I cannot help them, I probably know someone who can.

— from David A. J. McGlone [excerpted]

[Mr. McGlone, by way of being a nearly universal resource for the CP/M and ZCPR3 communities, is a licensed distributor of CP/M and of various commercial and public-domain software. He also offers repro documentation, a disk-copying service, and a fine newsletter, *The Z-Letter*, eminently worth reading at \$18 for 12 issues (2 years). "Finally," he says, "I am always looking for boot disks, since....CP/M helps no one if I do not have the right boot disk for a particular machine." Collectors and restorers owe it to themselves to contact Mr. McGlone at the address on page 32. — Editors]

FACE DOWN, 9-EDGE FIRST

➔ Save those old punch cards from the dump! Punch cards are a thing of the past, thank goodness, but that doesn't mean we should discard every last one. In their heyday, use of custom-printed cards with corporate or institutional logos was a point of pride for many organizations. As an example, prior to the Reagan-era substitution of new and somewhat garishly patriotic forms, the US monogram was prominently used as an elegant repeated background on the cardstock on which US government checks (such as tax refunds) were printed. These checks were printed on cardstock because they were punched cards, suitable for automated processing in the 1900-to-1970 style of automation. I dearly want a blank punched card in that format to add to my collection.

I have an extensive stock of duplicates. If you have old unused cards sitting around, I'll gladly trade! It's not stamp collecting, cards are bigger and fewer people collect them, but it can be interesting. My standard offer is simple: Mail me a stack of cards that you have a surplus of, and if you include an addressed envelope, I'll stuff it with an equal number of equal quality cards from my trading stock and pay return postage. Unless you're an established collector, there's little chance that you'll get any duplicates this way, and you'll end up with more variety than you started with!

— from Doug Jones, 816 Park Road, Iowa City, IA 52246, USA, via Internet

[But no lace cards, please, they jam the mail sorters.
— Editors]

QUESTIONS OF POLICY

◆ I have seen the first issue of the Analytical Engine and a question naturally arises. The Charles Babbage Institute has existed for some years now and it archives manuals, and other computer documents; also, the Computer Museum (Boston) preserves hardware and documents; the Annals of the History of Computing publishes articles in this area. So the question is why are you launching your project when these other possibilities exist?

— from Lloyd Fosdick, University of Colorado, via Internet

[Thank you for raising that point. In our view, computing is such a formidable field — and currently has such momentum — that it's natural to consider the establishment of *multiple* computer museums and archives, for the sake of regional emphasis. There are, as comparable examples, multiple art and science museums, multiple specialized libraries for other topics, and that's come to be expected. Computing will sustain a similar level of interest.

The Computer Museum in Boston, for example, is a truly fine institution and does a particularly good job of tracing computer development at MIT, along Route 128 (the "Silicon Ring") and at East Coast establishments further afield, like IBM. TCM has a nicely presented chunk of Whirlwind — one of the most important of the early digital computers or the earliest of the important ditto, take your pick — which is entirely proper, since it was developed across the river. It's similarly proper for the Smithsonian to have ENIAC, which was designed and built at the Moore School of Engineering in Pennsylvania.

But at the moment, there's no such institution in and for California. That's the rationale, or part of it, for CHAC. Certainly Silicon Valley, in order to tell the story of what happened there since Hewlett and Packard built their first oscillator in 1938, could endow and support an institution comparable to TCM! And while we're not taking that on singlehanded, we *are* pushing for it, making spikes, acting as a clearinghouse for expressions of interest.

As for archiving: Having archives in dispersed locations makes sense. The materials are more accessible, and they're safer, especially if a degree of

redundancy is built in. On-line databases have reduced the need for physical archiving, but they won't eliminate it in the foreseeable future.

The final assessment, probably, is that while we as a nation and society might someday come to the point of having "enough" computer museums, archives, and historical periodicals, we aren't *nearly* to that point yet. We hope for success for CHAC, but we wish it just as fervently to our colleagues at TCM, at the Smithsonian, at Apple and Intel, at the ACM, and at many other institutions all over the nation. Plenty of room for us all. — Editors]

ATTRIBUTION OF ELECTRONIC MAIL

◆ Recently there was a discussion on this mailing list regarding the use of messages posted to a newsgroup/bulletin board etc. in a scholarly publication.The messages are quoted directly....but no attributing of the messages to authors appears.

— from Dr. Rajeev Pandey, via Internet

[Thank you for raising this point and for drawing attention to a presently turbulent and indistinct border — the boundary between net.traffic and the print media.

Many publications, including PC Magazine, WIRED, and others, are reprinting Internet messages as letters, and they pursue several different policies in this regard. WIRED in particular prints the Internet address of the respondent, unless they are asked not to, in which case they run "Internet address withheld" or "unlisted Internet address." Withholding of these addresses seems to be more common now than formerly.

The ANALYTICAL ENGINE will reproduce Internet messages in its Letters column according to the following proposed policy:

1. We will ask all writers, in our net.replies, for permission to reproduce the message, if such permission is desired, and specifying whether all or part of the message is to be reproduced.
2. Current debate on privacy in electronic communication is as furious as it is copious. In recognition of this state of ferment, the ANALYTICAL ENGINE *will not* reproduce the Internet address of any individual or organization

unless the owner of that address specifically requests its publication.

3. Reprinted messages will be credited with a line at the end in this format:

— [author's name], [author's affiliation], via Internet where "Internet" is taken as generic, also comprising BITNET and USENET; or, where appropriate, "AOL," "CompuServe," *et al.*

4. If such a message is again reproduced, on public paper or on the net, it must be credited both to the author of the message and to the ANALYTICAL ENGINE.

Thank you for the opportunity to set forth this policy. We solicit your comments. As part of the broader issue of net privacy and the evolving view of intellectual property, this is an important point and we thank Dr. Pandey for raising it. — Editors]

QUERIES

MAINFRAME FRONT PANELS EAGERLY SOUGHT

➔ Unusual Systems, a consulting and development firm in Toronto, Canada, is collecting front panels (control panels) of mainframes and minicomputers, with the eventual ambition of providing museum-quality display for these important artifacts. This addresses a necessary compromise since, as Kevin Stumpf notes in the materials we received from him, "collecting entire systems is a costly venture." We can only agree with a sigh.

The collection now comprises 50 panels but Stumpf is looking for more; he claims to have "acquired every relevant [type of] panel or console in Canada" and is expanding his search to the United States. He is careful to note that this work is being done in the spirit of preservation and not in any sense for profit. If you are dismantling any of the computers in this list:

- * Burroughs B5000, B6700, B3500, B8500
- * CDC Omega, 3200, 3500, 1604, 1700
- * DEC LINC-8, PDP-8, PDP-8/S, VAX 11/780
- * GE 415, 435
- * HP 2114, 2100, 3000
- * Honeywell 1XX, 2XX, 4200, 8200, H316

- * IBM:
 - 650
 - 1401, 1410; 1130
 - 7010, 7040, 7090, 7094
 - 360/44, 67, 75, 85, 91, 195
 - 370/165 or 168
- * NCR Century 101 or 300
- * RCA 3301, 1230 or Spectra 70/xx
- * Scientific Control x700
- * SEL 810, 85
- * Sperry UNIVAC — any
- * Varian 620/i, 520i, V73

consider writing Mr. Stumpf at the address on page 32 or calling him at 519/744-2900. His work seems to us particularly relevant in light of the immense shakeout of mainframes that will take place during the next seven to ten years.

CITATIONS WANTED ON COMPUTING IN INSURANCE

➔ I am working on a study of the transition to and early use of computers in the life insurance industry. I would appreciate any information on this subject, and particularly names of people I might interview who were involved with insurance industry computing in the 1940s, 1950s, and 1960s.

Thanks very much for any information, references, or leads.

— from JoAnne Yates, Sloan School, MIT, via Internet

HUNTING PALEO-JARGON

➔ Has anyone ever heard the term "wholihan" applied to a subtle computer bug, or (even better) have any first-hand knowledge of its etymology?

I came across the term in a copy of *An Introduction to Electronic Data Processing* by Roger Nett and Stanley A. Hetzler, published in 1959 by The Free Press, Glencoe, Illinois. The authors provide the following description and etymology:

Occasionally a more difficult type of error, the obscure error....sometimes called the "wholihan", arises. This kind of error is characteristically of no truly consistent type and is the result of a combination of unique and unsuspected circumstances. It may be a very simple error but nonetheless remote

and difficult to expose. The term *wholihan* originated among persons programming for UNIVAC, where a man bearing that name once inadvertently produced a challenging and somewhat unforgettable example of the obscure error. It occurred in a simple enough routine, but in such a remote combination of circumstances that it is said to have taken 200 man-hours to discover the source of the error.

It's an interesting term....but I've never heard it anywhere, much less the above story behind it, or the reason for its ultimate demise....I suppose it's too much to hope for the "first actual wholihan found" to be preserved in some log book somewhere....

— from Lee Wittenberg, via Internet

[A member of CHAC recalls that this term was used at the Honeywell Data Center in San Francisco in the mid-1970's, not an early citation but one that broadens the use from the strict context of UNIVAC. Further citations, especially early ones, will be gratefully received and reproduced. — Editors]

PESKY GNAT IN SWEDEN

➡ Last week I bought a "vintage" computer at the local charity flea-market. It....is labelled "GNAT Computers, San Diego, California". "Model 10, SN 231". The front is labelled "MILLBANK SYSTEM 10"

The box is an all-in-one "console" type....and holds two 5 1/4" diskette drives, a green CRT monitor, and the keyboard.The processor board has a Z-80 and a LOT of glue logic and support IC's; it is labelled "GNAT 1000F level 3" and appears to be 'fully' equipped with 64Kb of RAM. There are....I/O ports at the back of the box (RS-232 etc.) Attached to the back of the CRT is another PCB labelled "GNAT 1005F Video Processor" [with,] among other things, an 8085 installed.

It is apparently a CP/M system, but since I have no diskettes, I have not been able to boot CP/M. I have, however, succeeded in starting a 'monitor' program by pressing the reset-button.

— from Henrik Carling, University of Lund, via Internet

[David McGlone was able to tell us that the GNAT disk format is DS/DD 48 tpi, but beyond that he draws a blank. Does anyone know anything about this computer? Further info much appreciated and will be forwarded. — Editors]

FLYING BLIND ON SOL-20 ASSEMBLER

➡ I've got a SOL-20 sitting in my closet.... My last act with it was to try and figure out the ALDS (assembly language development system), contained in ROMs on an S-100 card; you see, I have no docs. I managed todetermine that the previous owner had the ROMs in the wrong sockets, and straightened that outbut still couldn't figure out what I was supposed to do to get it started. Unfortunately the magazines of the period don't shed much light on anything beyond the monitor ROM.

— from "frank", via Internet

[We have purchased a SOL-20 (see above) and haven't taken delivery of it yet, but it supposedly has manuals. If they include material pertinent to the ALDS, we would be glad to forward a copy at cost. — Editors]

FIREBOTTLE QUERY

➡ I have a circuit cage, approx 9" wide, 1" deep, and 5" tall. The circuit is made of discrete components, including some semi-conductor diodes (1N480's, 1N119's). On the top of the cage are 8 electron tubes (dual-triode) with the letters IBM, and the number 317261 on them. As originally mounted, the rack had the tubes out, and the length vertical (9" high, 5" deep, 1" wide). What machine did this come from?

Extra credit: the circuit is a dual JK, resettable, presettable, addressable flip-flop. What function did it serve?

— from Data Rentals and Sales, via Internet

[This is a trick question, since the writer has the answer, but we'll publish it for the delectation of our audience. — Editors]

COMPUTERS IN MILITARY COMMAND AND CONTROL

➔ I have read numerous reference to SAGE (Semi-Automatic Ground Environment, I think...) and WWMCCS (World-Wide Military Command and Control System) in the literature over the years.

Does anyone have any pointers to literature/books concerning these systems?

— from John K. Scoggin, Jr. via Internet

[There's a good survey of SAGE, along with so much else, in:

IBM's Early Computers

Charles J. Bashe *et al.*

MIT Press, Cambridge, Mass. 1986

but nothing in our library mentions WWMCCS. Any ideas from our readers? — Editors]

OLD-IRON SPECS WANTED

➔ I'd be interested in knowing something about the workings of some of these early computers (especially the programmable ones,) e. g., amount of memory, instruction set, I/O capabilities.

I'm afraid there isn't a lot I can contribute to this, I know only what I've read in various books. They tend to give information like the number of valves (tubes) in each machine, and make some statement about the number of operations a second it was apparently capable of, but little more.

— from David Hembrow, Harlequin Ltd. (UK), via Internet

[David was referring to ENIAC but we've also seen considerable recent interest in EDSAC, EDVAC, ACE/Pilot ACE and the IBM 70x series.

We can recommend the TECHNOLOGY volume of Cortada's *Historical Dictionary*, which gives meticulous coverage to some of the older big iron, but we do feel that all possible specifications for these machines should be collated and published while they're still in the semi-public record.

As a long-term project, CHAC intends to compile spec tables of historically important mainframes and make them available through the ENGINE's request daemon. The first of these will be for the IBM

701. We'll say more about this in the January 1994 issue. — Editors]

ORIGIN OF "MINICOMPUTER" WANTED

➔ In my search through old issues of *Computers and Automation*, I've noticed that the term *minicomputer* didn't show up until 1968. The first use of the term I can find is in a full-page Interdata ad in May 1968, page 10. By December 1969, the term must have become fairly common, because they devoted a full issue to "Minicomputers (and Their Applications)". Curiously, none of the articles in that issue are by people from DEC or CDC, the two companies that are in the best condition to claim to have originated the minicomputer (either the CDC 112, the Lincoln Labs LINC or the DEC PDP-5 / PDP-8 family seem to have earned that title).

So, did Interdata coin the phrase *minicomputer*? Can someone pin down a citation to the word prior to May 1968?

— from Doug Jones, via Internet

[We have no idea, but certainly the locution must have been fairly fluid, given that as late as 1976-77, micros like the IMSAI and Altair were referred to in literature as "mainframes!" Ideas? — Editors]

DOES THE S-100 BUS STOP HERE?

➔ I've recently acquired 3 IMSAI S-100 systems that I am trying to restore to full usability. (i.e. running CP/M or MP/M)...it shouldn't be impossible to get a working system or two. One of the systems has a IMSAI MPU-B processor board (8085) that I could really use some documentation for (in particular, how to access the serial port on it, and if possible, how to set the power-on action.)

....One of the[m] seems to have stopped upgrading around the era of the Tarbell cassette interface. Anyone happen to have....old Tarbell cassette operating systems or BASICs that I have....manuals for? Is Tarbell still in business, by any odd chance? Are any of the S-100 companies (California Computer Systems, IMSAI, ALTAIR, Godbout, Compupro, etc.) still doing business in any form? Or maybe even still selling S-100 boards?

— from Timothy D. Shoppa via Internet

[Well, MITS/Altair was sold to Pertec and Pertec went under, IMSAI got absorbed into ComputerLand amid a flurry of lawsuits, and Compupro/Viasyn closed its doors in late 1991, so there's three down. Can anybody help Mr. Shoppa with docs, used boards, disks or paper tape? — Editors]

ONE FROM THE EDITORS

➔ From our last argument around the coffee and doughnuts: What was the first electronic digital computer in California, and when, and whose was it?

(We win either way with this one. It'll turn into an article topic or great mail.)

PUBLICATIONS RECEIVED

Guide to the Oral History Collection of the Charles Babbage Institute. Aspray, Bruemmer, Melehy and Traub, eds. Charles Babbage Institute, Minneapolis, MN, 1986. 110 pp.

Resources of the History of Computing: A guide to U. S. and Canadian records. Bruemmer, Traub and Brosenne, eds. Charles Babbage Institute, Minneapolis, MN, 1987. 187 pp.

Charles Babbage Institute Newsletter, Volume 15, Number 3, Spring 1993. 10 pp. Oral History Cataloging Initiative; Supercomputing '92; Tomash fellowships; Technology museum in Paderborn; Fourth centenary of Schickard's birth; Bank of America honors Al Zipf; more.

(The three above from Judy E. O'Neill.)

A CBO Study: Promoting High-Performance Computing and Communications. Congressional Budget Office, June 1993. 63 pp. History of the Internet, of supercomputer technology, and of their markets.

"Builder of First Analog Computer Trying to Recreate Invention," James McWilliams, *The Huntsville Times*, Huntsville, AL, June 6, 1993. Helmut Hoelzer and his construction of the world's first fully electronic analog computer.

"Preserving Software's History As It Is Written," George Tibbits, Associated Press, August 9, 1993. A survey of efforts to preserve historically

important software at the Smithsonian Institution, the National Museum of American History, and various companies.

(The three above from Philip Webre.)

The Z-Letter: Newsletter of the CP/M and Z-System community. Number 25, May-June 1993; Number 26, July-August 1993. A forum for the support, preservation, collection and sale of early microcomputers running the CP/M and ZCPR3 operating systems. (From David A. J. McGlone.)

"An Introduction to Bookbinding: Particularly aimed at the preservation of old DEC handbooks," Douglas Jones, 1992. Jones, a mainstay of the USENET newsgroup *alt.sys.pdp8*, has written a lucid and extensive treatise on dissecting, photocopying, rebinding and restoring old DEC manuals, with principles generally applicable to any technical documentation. If you ever rely on docs that are yellowed and crumbling, read this — soon.

[This document is an ASCII file of about 35K. To request it by automagic e-mail, send a message to

`engine@win.net`

with a *subject* line of

`docfix`

Alternatively, contact CHAC at the El Cerrito mail address to request hard copy.]

"Microcomputer History and Prehistory — An Archaeological Beginning," Harold A. Layer, *Annals of the History of Computing*, Volume 11, Number 2, 1989. (From the author.) An illustrated article/listing of a formidable collection of early micros, calculators, electronic games and computer-related toys.

ADDRESSES OF CORRESPONDING ORGANIZATIONS

The Computer Museum, 300 Congress Street, Boston MA 02210. Brian C. Wallace, curator of historical computing.

Charles Babbage Institute, 103 Walter Library, University of Minnesota, 117 Pleasant Street S. E., Minneapolis MN 55455. Judy E. O'Neill, associate director.

Natural Resources and Commerce Division,
Congressional Budget Office, 410 Ford House Office
Building, Washington DC 20515. Philip Webre,
principal analyst.

Intel Museum, P. O. Box 58119, Santa Clara, CA
95052-8119. Jodelle A. French, curator.

Smithsonian Institution, Air and Space Museum.
Washington, DC 20560. Paul E. Ceruzzi, curator of
historical computing.

Smithsonian Institution, Division of Computers,
Information and Society. Washington DC 20560.
David Allison, director.

Unusual Systems, 220 Samuel Street, Kitchener,
Ontario N2H 1R6, Canada. Kevin Stumpf,
president.

The Z-Letter, Lambda Software Publishing, 149 West
Hilliard Lane, Eugene OR 97404. David A. J.
McGlone, editor and publisher.

THANKS TO....

Scott Callan for insisting that hard work isn't much
use without a big mouth.

Susan deRenne Coerr for hours of invaluable advice
on accession, registration, curatorship, and the key
point — loading docks!

Robert X. Cringely for permission to quote at
length from *Accidental Empires*.

Hilary Crosby for setting up our accounting and
subscription system, and smoothing the path to our
501(c)3.

Tom Ellis for on-site support at impossible hours.

Michael Tague, Rob Conklin, Jamie Warren and
Connie Rogers at Computer Witchcraft for
supplying an Internet connection with maximum
reliability and minimum pain.

All the great people at *news.newusers.answers*,
alt.folklore.computers, and *alt.sys.pdp8* for making
USENET such an interesting place to live.

GUIDELINES FOR DISTRIBUTION

The ANALYTICAL ENGINE is intellectual
shareware. Distribution of *complete, verbatim* copies
through online posting, Internet mail or news, fax,

postal service or photocopying is encouraged by the
Computer History Association of California.

Excerpting or brief quotation for fair use, including
review or example, is also permitted, with one
exception: Any material copyright to or by a third
party and reprinted in the ANALYTICAL ENGINE
by permission shall not be used in another
periodical or context, unless the permission of the
copyright holder is separately secured for the new
use.

Alterations, abridgments or hacks of the
ANALYTICAL ENGINE which change the intent
or meaning of original content; or which contrive
to bring income to any person or organization
other than the Computer History Association of
California; or which contrive to injure the
Computer History Association of California, its
officers, contributors, volunteers or members; are
PROHIBITED. Reproduction of the
ANALYTICAL ENGINE without its subscription
coupon is abridgment in this sense.

GUIDELINES FOR SUBMISSION

The ANALYTICAL ENGINE solicits manuscripts
of 600 to 1500 words on the general topic of the
history of computing in, or with significant
reference to, the State of California. Articles should
focus on one interesting or illuminating episode and
should be written for a technically literate general
audience. Submissions are welcome from both
members and non-members of the CHAC and a
one-year membership, or extension, will be given for
each article published. Article deadlines are the first
of each month prior to publication: June 1 for the
July issue, September 1 for the October issue,
December 1 for the January issue, and March 1 for
the April issue.

Decision of the editors is final but copyright of all
published material will remain with the author.

The preferred document file format is Microsoft
Word for DOS or Windows, but almost any DOS
or Macintosh word processor file will be acceptable.
Submit manuscripts on DOS 5.25" or DOS or Mac
3.5" diskettes. Alternatively, please provide an
ASCII file attached to Internet mail. Please avoid
submitting on paper unless absolutely necessary.

NINES-CARD

Here's an authentic chunk of computer history, ERMA's last words. ERMA was the first computer designed here in the San Francisco Bay area (later the Silicon Valley), in development at Stanford Research Institute and at GE from 1950 to 1959. It was a check processing system for Bank of America, the first of its kind, and among other things it established an international standard for machine readable characters on bank checks — the account and other numbers printed on your checks.

There were about a dozen ERMA systems in production use, and when the last one was shut down this was the final message. You can see this last machine carefully restored in a good exhibit in the lobby of a building at the Bank of America Technology Center in Concord, across the Bay from San Francisco. The exhibit includes video testimonies from the original ERMA developers. Call Ed Hawthorne at the Bank of America, 510/675-1303, if you're in the area and want to see the exhibit.

— from Peter Nurkse, SUN Microsystems, via Internet

ERMA Says Goodbye

A MESSAGE TO ALL MY CO-WORKERS
FROM ERMA

IN MY ELEVEN YEARS OF SERVICE TO THE BANK OF AMERICA, I HAVE BEEN PRIVILEGED TO WORK WITH SOME OF THE BANK'S FINEST EMPLOYEES. FROM PEOPLE LIKE EMMETT JENKINS, DICK DAVIS, JOHN COOMBS, AND MANY MORE WHO WERE WITH ME FROM THE BEGINNING, TO BOB LEE AND ALL OF MY CURRENT CO-WORKERS WHO ARE ASSISTING ME IN THE PROCESSING OF TRAVELLERS CHEQUES — MY FINAL APPLICATION — I CAN ONLY SAY THANK YOU. TOGETHER WE HAVE MADE GREAT STRIDES IN BANKING. AND I CANNOT HELP BUT FEEL, AS THE FIRST COMPUTER SYSTEM TO BE USED FOR BANKING APPLICATIONS, THAT MY RETIREMENT BRINGS TO CLOSE AN HISTORIC ERA. TO BE THE FIRST IN SOMETHING IS A GREAT ACHIEVEMENT,

AND I AM VERY PROUD. BUT MY SUCCESS COULD NOT HAVE BEEN POSSIBLE WITHOUT THE HELP OF SO MANY FINE PEOPLE.

ALTHOUGH THE END OF AN ERA IS NEAR, AND WE WILL SOON PART, I WILL NEVER FORGET MY FRIENDS, AND I WISH YOU ALL THE GREATEST SUCCESS IN THE FUTURE.

TO MY CURRENT — AND FINAL — ASSISTANTS, I BID FAREWELL —

MY BEST TO ALL,

ERMA

The ANALYTICAL ENGINE

Volume 1, Number 2

October 1993

ISSN 1071-6351

newsletter of the Computer History Association of California, is published four times a year — in January, April, July and October — at El Cerrito, California.

Subscriptions are available with Association membership at \$25 per year individual, \$75 per year corporate or institutional, or \$15 per year low-income, student or senior. Use the coupon to subscribe, or contact the Association at:

FAX: 510/528-5138

Internet: cpu@chac.win.net

US Mail:
1001 Elm Court
El Cerrito, CA 94530-2602

Kip Crosby, Managing Editor
Jude Thilman, Telecomm Editor

CONTENTS

<p>EDITORIAL: Hello, World 1</p> <p>INITIATIVE 1999: Warnings of Extinction 1</p> <p>Compilation Project: Getting It All Together 3</p> <p>Digital's History Project 3</p> <p>Volunteer Liaison Between CHAC And DEC 3</p> <p>Intel Museum Refreshes Its Exhibits 4</p> <p>Museum Plans At University Of California, Davis 4</p> <p>Urgent: Spotters Wanted 4</p> <p>Desperate Plea For Storage 5</p> <p>Desperate Plea For Money 5</p> <p>And Speaking Of Money 6</p> <p>Overview Of Bureaucratic Processes 6</p> <p>LOGO AND SMALLTALK, Aaron Alpar 7</p> <p>OF THEE I SING, Tom E. Ellis 10</p> <p>THE CHARLES BABBAGE INSTITUTE, Judy E. O'Neill 12</p> <p>PROGRAMMING THE 1401, Leo Damarodas 15</p> <p>Book Review: THE DREAM MACHINE 18</p> <p>Acquisitions: IMSAI 8080 19</p> <p>LETTERS</p> <p> Historical Accuracy 19</p> <p> 1401 As Slave Processor 20</p> <p> 1401: Recollections And Corrections 20</p> <p> More On Autocoder 20</p> <p> 1401 Fortran Compiler 21</p> <p> More Glitches 21</p> <p> 1401s I Have Known 21</p> <p> From The Smithsonian Institution 22</p>	<p>From The Babbage Institute 22</p> <p>What Ever Became Of The Otrona? 23</p> <p>Starting From Scratch 23</p> <p>Greetings From Iowa 23</p> <p>Museum Plans In The Northwest 24</p> <p>"The Job Needs To Be Done" 24</p> <p>Considerations Of History And Reliability 25</p> <p>Resource For Those Working With CP/M 26</p> <p>Face Down, 9-Edge First 26</p> <p>Questions Of Policy 27</p> <p>Attribution Of Electronic Mail 27</p> <p>QUERIES.....</p> <p> Mainframe Front Panels Eagerly Sought 28</p> <p> Citations....On Computing In Insurance 28</p> <p> Hunting Paleo-Jargon 28</p> <p> Pesky GNAT In Sweden 29</p> <p> Flying Blind On SOL-20 Assembler 29</p> <p> Firebottle Query 29</p> <p> Computers In....Command And Control 30</p> <p> Old-Iron Specs Wanted 30</p> <p> Origin Of "Minicomputer" Wanted 30</p> <p> Does The S-100 Bus Stop Here? 30</p> <p> One From The Editors 31</p> <p> Publications Received 31</p> <p> Addresses Of Corresponding Organizations 31</p> <p> Thanks To 32</p> <p> Guidelines For Distribution 32</p> <p> Guidelines For Submission 32</p> <p> Nines-Card 33</p>
--	---

✂ + \$£DM¥ + ✉ = SUBSCRIBE!

___ Yes! Please enroll me in the CHAC for the next year. I enclose

- ___ Individual US\$25 e-mail / \$35 paper
- ___ Institutional US\$75 e-mail / \$85 paper
- ___ Low-income US\$15 e-mail / \$25 paper

Send four issues of the ANALYTICAL ENGINE to my e-mail address:

___ I would prefer to receive paper copies.
(Please include your physical address.)

NEXT ISSUE!

JANUARY 1994: The REAL first micro. DSP on a Zilog Z-80. IBM 701, the stuff of legend. Literature. Queries. Answers. Great mail. More....

Downloadable January first — don't miss it!