October, 1976
Vol. 25, No. 10

computers and people

formerly *Computers and Automation*

100,000 BITS OF MEMORY ON ONE CHIP

The Handling of Natural Language by a Computer (Conclusion)
– B. Raphael

Microprocessors in Computer Education
– L. A. Leventhal

Software Products: The Next Ten Years
– W. F. Bauer

The Robot "Shakey" and "His" Successors
– B. Raphael

The Notebook on
COMMON SENSE, ELEMENTARY AND ADVANCED

is devoted to development, exposition, and illustration of what may be the most important of all fields of knowledge:

WHAT IS GENERALLY TRUE AND IMPORTANT =

COMMON SENSE + WISDOM + JUDGEMENT AND MATURITY + SCIENCE IN GENERAL + TECHNIQUES FOR SOLVING PROBLEMS +

TECHNIQUES FOR AVOIDING MISTAKES + SOME PARTS OF OPERATIONS RESEARCH + SOME PARTS OF SYSTEMS ANALYSIS + AVOIDANCE OF LOGICAL FALLACIES + ...

PURPOSES:

- to help you avoid pitfalls
- to prevent mistakes before they happen
- to display new paths around old obstacles
- to point out new solutions to old problems
- to stimulate your resourcefulness
- to increase your accomplishments
- to improve your capacities
- to help you solve problems
- to give you more tools to think with
- .....

8 REASONS TO BE INTERESTED IN THE FIELD OF COMMON SENSE, WISDOM, AND GENERAL SCIENCE

COMPUTERS are important —
But the computer field is over 25 years old. Here is a new field where you can get in on the ground floor to make your mark.

MATHEMATICS is important —
But this field is more important than mathematics, because common sense, wisdom, and general science have more applications.

LOGIC is important —
But this field is more important than logic, because common sense plus wisdom plus science in general is much broader than logic.

WISDOM is important —
This field can be reasonably called "the engineering of wisdom".

COMMON SENSE is important —
This field includes the systematic study and development of common sense.

SCIENCE is important —
This field includes what is common to all the sciences, what is generally true and important in the sciences.

MISTAKES are costly and to be AVOIDED —
This field includes the systematic study of the prevention of mistakes.

MONEY is important —
The systematic prevention of mistakes in your organization might save 10 to 20% of its expenses per year.

Topic:
THE SYSTEMATIC PREVENTION OF MISTAKES

Already Published

- Preventing Mistakes from:
  - Failure to Understand
  - Forgetting
  - Unforeseen Hazards
  - Placidity
  - Camouflage and Deception
  - Laxity
  - Bias and Prejudice
  - Ignorance

To Come

- Preventing Mistakes from:
  - Interpretation
  - Distraction
  - Gullibility
  - Failure to Observe
  - Failure to Inspect

Topic:
SYSTEMATIC EXAMINATION OF GENERAL CONCEPTS

Already Published

- The Concept of:
  - Expert
  - Rationalizing
  - Feedback
  - Model
  - Black Box
  - Evolution
  - Niche
  - Understanding
  - Idea
  - Abstraction

To Come

- Strategy
- Teachable Moment
- Indeterminacy
- System
- Operational Definition

- - - - - - - - - - - - - - - - (may be copied on any piece of paper) - - - - - - - - - - - - - - - - - -

To: Berkeley Enterprises, Inc.
815 Washington St., Newtonville, MA 02160

( ) Yes, please enter my subscription to *The Notebook on Common Sense, Elementary and Advanced* at $12 a year (24 issues), plus extras. I understand that you always begin at the beginning and so I shall not miss any issues.

( ) Please send me as free premiums for subscribing:

1. Right Answers — A Short Guide to Obtaining Them
2. The Empty Column
3. The Golden Trumpets of Yap Yap
4. Strategy in Chess
5. The Barrels and the Elephant
6. The Argument of the Beard

( ) I enclose $________ ( ) Please bill my organization

RETURNABLE IN 7 DAYS FOR FULL REFUND IF NOT SATISFACTORY
HOW CAN YOU LOSE?

Name________________________ Title______________

Organization________________________________

Address (including zip)________________________

Signature__________________ Purchase Order No.__________

"RIDE THE EAST WIND: Parables of Yesterday and Today"

by Edmund C. Berkeley, Author and Anthologist

Published by Quadrangle/The New York Times
Book Co., 1974, 224 pp, $6.95

The Fly, the Spider, and the Hornet

Once a Fly, a Spider, and a Hornet were trapped inside a window screen in an attic. For several hours they walked up and down, left and right, here and there, all over the screen. They could look through the screen at the summer woods, feel the summer breezes, and smell the summer smells; but they could not find any hole to pass through the screen to the woods and fields so tantalizingly close, yet so far away.

Finally they decided to hold a conference on the problem of getting through the screen. The Fly spoke first, and said, "My Colleagues, ...

The Fox of Mt. Etna and the Grapes

Once there was a Fox who lived on the lower slopes of Mt. Etna, the great volcano in Sicily. These slopes are extremely fertile; the grapes that grow there may well be the most delicious in the world; and of all the farmers there, Farmer Mario was probably the best. And this Fox longed and longed for some of Farmer Mario's grapes. But they grew very high on arbors, and all the arbors were inside a vineyard with high walls, and the Fox had a problem. Of course, the Fox of Mt. Etna had utterly no use for his famous ancestor, who leaping for grapes that he could not reach, called them sour, and went away.

The Fox decided that what he needed was Engineering Technology. So he went to a retired Engineer who lived on the slopes of Mt. Etna, because he liked the balmy climate and the view of the Mediterranean Sea and the excitement of watching his instruments that measured the degree of sleeping or waking of Mt. Etna. The Fox put his problem before the Engineer . . . .

The Fire Squirrels

Scene: Two squirrels, a young one named Quo, and an older one named Cra-Cra, are sitting by a small campfire in a field at the edge of a wood. Behind them hung on a low branch of a tree are two squirrel-size hammocks. Over each of the hammocks is a small canopy that can be lowered to keep out biting insects. It is a pleasant summer evening; the sun has just recently set, and the stars are coming out: —

Quo: Cra-Cra, you know I don't believe the old myths any more. Tell me again how it really happened.

Cra-Cra: Just this: we received our chance because they dropped theirs. It is as simple as that.

Quo: In other words, they were the first animals to use tools, and we are the second?

Cra-Cra: Yes. There is a mode of surviving in the world . . . .

Missile Alarm from Grunelandt

Once upon a time there were two very large and strong countries called Bazunia and Vossnia. There were many great, important, and powerful leaders of Bazunia who carefully cultivated an enormous fear of Vossnia. Over and over again these important and powerful leaders of Bazunia would say to their fellow countrymen, "You can't trust the Vossnians." And in Vossnia there was a group of great, important, and powerful leaders who pointed out what dangerous military activities the Bazunians were carrying on, and how Vossnia had to be militarily strong to counteract them. The Bazunian leaders persuaded their countrymen to vote to give them enormous sums of money to construct something called the Ballistic Missile Early Warning System, and one of its stations was installed in a land called Grunelandt far to the north of Bazunia.

Now of course ballistic missiles with nuclear explosives can fly any kind of a path all around a spherical world, and they do not have to fly over northern regions. But this kind of reasoning had no influence on the leaders of Bazunia who wanted the money for building BMEWS. Nor did it have influence on their countrymen, who were always busy, trying to make money — in fact often too busy to think clearly . . . .

---

52 parables (including fables, anecdotes, allegories)
23 never published before
27 authors
18 full-page illustrations
330 quotations and maxims

7 Parts: The Condition of Man / On Flattery and Persuasion / On Perseverance and Resourcefulness / Behavior — Moral and Otherwise / The Problem of Truth / On Common Sense / Problem Solving

---

THE VALUE OF THIS BOOK TO YOU
CAN BE VERY GREAT

JUDGE FOR YOURSELF AT NO COST

RETURNABLE IN 10 DAYS FOR FULL REFUND
IF NOT SATISFACTORY
You can read it all in 10 days — and keep it only if you think it is worth keeping.

HOW CAN YOU LOSE?

— — — — (may be copied on any piece of paper) — — — — — — — —

To: Berkeley Enterprises, Inc.
815 Washington St., S9, Newtonville, MA 02160

( ) Please send me _____ copy(ies) of *Ride the East Wind:*
I enclose $7.00 per copy (includes handling).

Total enclosed __________ (Prepayment is necessary)

My name and address are attached.

Vol. 25, No. 10
October, 1976

Editor and Publisher **Edmund C. Berkeley**
Assistant to the Publisher **Judith P. Callahan**
Assistant Editors **Barbara J. Wolmer**
Neil D. Macdonald
K. Kaufmann
Art Editor **Grace C. Hertlein**
Software Editor **Stewart B. Nelson**
Contributing Editors **John Bennett**, **John W. Carr III**, **Grace C. Hertlein**, **Linda Ladd Lovett**, **Ted Schoeters**, **Richard E. Sprague**, **Edward A. Tomeski**
London Correspondent **Thomas Land**
Advisory Committee **Ed Burnett**, **James J. Cryan**

Editorial Offices
Berkeley Enterprises, Inc.
815 Washington St.
Newtonville, MA 02160
617–332–5453

Advertising Contact
The Publisher
Berkeley Enterprises, Inc.
815 Washington St.
Newtonville, MA 02160
617–332–5453

"Computers and People", formerly "Computers and Automation", is published monthly, 12 issues per year, at 815 Washington St., Newtonville, MA 02160 by Berkeley Enterprises, Inc. Printed in U.S.A. Second Class Postage paid at Boston, MA, and additional mailing points.

Subscription rates: United States, $11.50 for one year, $22.00 for two years. Canada: add $1 a year; elsewhere, add $6 a year.

NOTE: The above rates do not include our publication "The Computer Directory and Buyers' Guide". If you elect to receive "The Computer Directory and Buyers' Guide", please add $12.00 per year to your subscription rate in U.S. and Canada, and $15.00 elsewhere.

NOTE: No organization in Switzerland or Morocco is authorized or permitted by us to solicit subscriptions for or receive payments for "Computers and People" or "The Computer Directory and Buyers' Guide". Such subscriptions and payments should be sent directly to us.

Please address mail to: Berkeley Enterprises, Inc., 815 Washington St., Newtonville, MA 02160

Postmaster: Please send all forms 3579 to Berkeley Enterprises, Inc., 815 Washington St., Newtonville, MA 02160

© Copyright 1976, by Berkeley Enterprises, Inc.

Change of address: If your address changes, please send us both your new address and your old address (as it appears on the magazine address imprint), and allow three weeks for the change to be made.

computers
and people formerly *Computers and Automation*

Computers and Programming

8 Software Products: The Next Ten Years [A]
by Walter F. Bauer, President, Informatics, Inc., Woodland Hills, CA
Despite the success and growth of software products, this entire area of business is poorly understood within the computer industry. How will the use of software products change and develop? How can the computer industry protect against unauthorized use of software? Will there be dramatic changes in software technology during the next ten years?

Microprocessor Computers

11 Microprocessors in Computer Education [A]
by Dr. Lance A. Leventhal, Grossmont College, El Cajon, CA
The development of microprocessors has introduced a new potential for computer education in small educational institutions with limited technical facilities. Although not as powerful as a minicomputer, when memory and input-output circuits are added to a microprocessor, it can perform the same functions as a large computer, allowing small schools to give their students a wide range of computer experiences.

Computers and Natural Language

18 The Handling of Natural Language by a Computer — Part 2 (Conclusion) [A]
by Bertram Raphael, Stanford Research Institute, Menlo Park, CA
While linguists have been carrying out studies on the nature of language and its semantics, computer scientists have been experimenting with ways computers can be made to understand natural language. None of the systems developed so far have been powerful enough for practical use, but they have indicated the present limits of automatic language processing and new directions for research.

Computers and Robots

7 The Robot "Shakey" and "His" Successors [A]
by Bertram Raphael, Stanford Research Institute, Menlo Park, CA
What the robot Shakey could do and could not do — and what Shakey might have done if he had continued to be funded after 1972. Also, what Shakey has led to and is leading to, even though much robot behavior that looks easy proves to be very difficult to implement.

The magazine of the design, applications, and implications of information processing systems – and the pursuit of truth in input, output, and processing, for the benefit of people.

6 The Level of Competence of Animals and Robots [E]
by Edmund C. Berkeley, Editor
An interesting wild animal like a raccoon, coyote, or squirrel can do many things that no robots yet constructed by human beings can do. Why? How? Does the design of animals give hints for the design of robots?

Advances in Computer Design

1,5 100,000 Bits of Memory on One Chip [F]
by TRW, Redondo Beach, CA
A new device using "Charge-Coupled Device" technology.

Computers and Information Systems

24 Computer Aids in Documentation of Telephone Technology [N]
by Peg Murray, Western Electric, New York, NY

Computer Applications

25 Computer Helps Idaho Officials Provide Quick Financial Assistance to Flood Victims [N]
by Dick Sybert, Idaho Department of Engineering, Boise, ID

25 Programmable Calculator Scores Women's Air Competition [N]
by Michael Malone, Hewlett Packard, Palo Alto, CA

25 Minicomputer Tackles Society Problems with Models and Simulations [N]
by Diane Wilson, University of Notre Dame, Notre Dame, IN

Computers and the Law

25 Computer Based System Identifies and Matches Fingerprints [N]
by Ralph Wallenhorst, Calspan Technology Products, Buffalo, NY

26 Computerized Research System Helps Lawyers Provide Better Legal Services [N]
by Amos Landman, Ruder & Finn, New York, NY

Computers and Art

22 1976 Computer Art Exposition – Supplement, October, 1976 [A]
by Grace C. Hertlein, Art Editor

Computers, Games, and Puzzles

Games and Puzzles for Nimble Minds – and Computers [C]
by Neil Macdonald, Assistant Editor
27 MAXIMDIJ – Guessing a maxim expressed in digits
27 NAYMANDIJ – A systematic pattern among randomness?
27 NUMBLES – Deciphering unknown digits from arithmetical relations

Front Cover Picture

A new way to store large amounts of data on a very small integrated circuit like the one in the girl's hand, is produced using Charge Coupled Device Technology. By the end of 1976, a Charge Coupled Memory chip, a little larger than ¼ inch square, will store 100,000 bits. The same amount of data stored on conventional punched cards would require about 1200 individual punched cards. The device is being developed by TRW, Redondo Beach, Calif.

101 MAXIMDIJES

Over 100 amusing, easy, cryptographic puzzles .... with maxims, quotations, sayings, etc., as the answers .... with lists, tables, and a guide for solving speedily .... hours of fun and entertainment for you and your friends.

For your copy, send $1.80, plus 20 cents for postage and handling, with your name and address to:

Berkeley Enterprises, Inc.
815 Washington St.
Newtonville, Mass. 02160

Key

[A] – Article
[C] – Monthly Column
[E] – Editorial
[F] – Forum
[N] – Newsletter
[R] – Reference

NOTICE

*D ON YOUR ADDRESS IMPRINT MEANS THAT YOUR SUBSCRIPTION INCLUDES THE COMPUTER DIRECTORY. *N MEANS THAT YOUR PRESENT SUBSCRIPTION DOES NOT INCLUDE THE COMPUTER DIRECTORY.

EDITORIAL

The Level of Competence of Animals and Robots

When I reflect on the competence of an interesting animal, such as a squirrel, a raccoon, or a coyote, I am always amazed. Not long ago, a raccoon came into the kitchen of my house in a suburb of Boston, ate the catfood out of the cat's bowl, and left. His (or her) calling card consisted of some muddy footprints. How did the raccoon guess that the catfood was in the kitchen? How did the raccoon judge the opportunity to come in? Smart raccoon! — As to coyotes, I understand they have now spread as far east as New Jersey. They must have solved a lot of problems using observation and reasoning — for I have never read any reports indicating that coyotes use a language.

One of the best (probably true) squirrel stories that I know is the following: On the wide lawn of a hospital near Philadelphia, there was a bird feeder mounted on a slippery iron pipe about five feet above the ground. On top of the pipe and under the bird feeder a round flat metal plate projected in every direction about 3 or 4 inches beyond the pipe. Fastened on top of the plate at the center was a regular dispenser of sunflower seeds for birds. A squirrel came along the lawn, surveyed the feeder, and struggled up the pipe to the top; but he could not get around the plate. Scolding and complaining, he finally slipped back down the pipe to the ground and went away. Over the next hour, three more squirrels came, and tried, and failed, and went away, scolding. At last there came a fifth squirrel. He stood a little distance away, with his paws folded in front of him, and looked carefully at the pole and the feeder. Then with a run and a jump, he whisked himself up the pole, put one paw on the edge of the plate, and flung himself over the edge with the momentum that he had not yet lost. Then he ate his fill of sunflower seeds, and subsequently departed, over the edge of the platform and down the pole.

What Bertram Raphael has to say on the competence of Shakey the Robot (see the letter to the editor on the facing page) is very illuminating: "The scientists who worked on Shakey developed a deep appreciation of how difficult it is to produce a robot even with relatively trivial abilities, let alone the true science-fiction-like independent competence."

How then does it happen that the world is full of animals (living species) that display remarkably competent behavior?

The evolution of the physical form of animal species is reported in the fossil record for more than 600 million years, and studied in paleontology. This evolution must have been paralleled by an evolution of behavior that can only be inferred. Always the reward of survival of a species is bestowed on the species that can perceive a situation more quickly and respond to a situation more appropriately. Some animals win out by flight; so it becomes important for them to perceive and understand what they must flee from and how. Other animals win out by attack; so it becomes important for them to perceive and understand what to attack and how to attack it.

The attainment of competent robot behavior may well lie in imitating many of the structural features of the old animal brain, including a great deal of parallel processing, in preference to imitating the new (2 million years old) human brain, with its "one-track mind".

Edmund C. Berkeley

Edmund C. Berkeley
Editor

The Robot "Shakey" and "His" Successors

Bertram Raphael
Director, Artificial Intelligence Center
Stanford Research Institute
Menlo Park, CA 94025

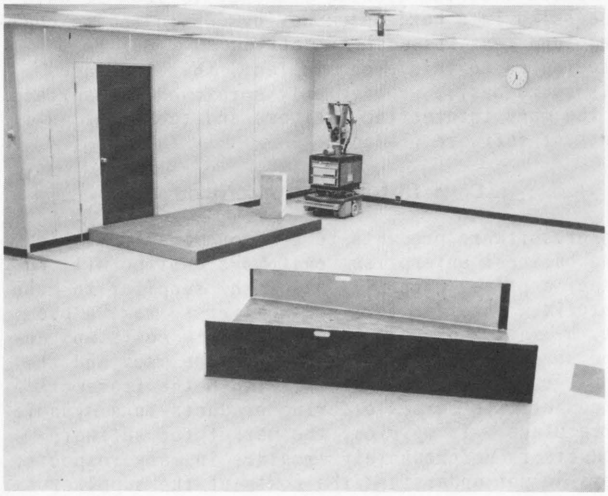
[Being a letter to the Editor received September 3, 1976]

The Level of Competence of Shakey, the Robot

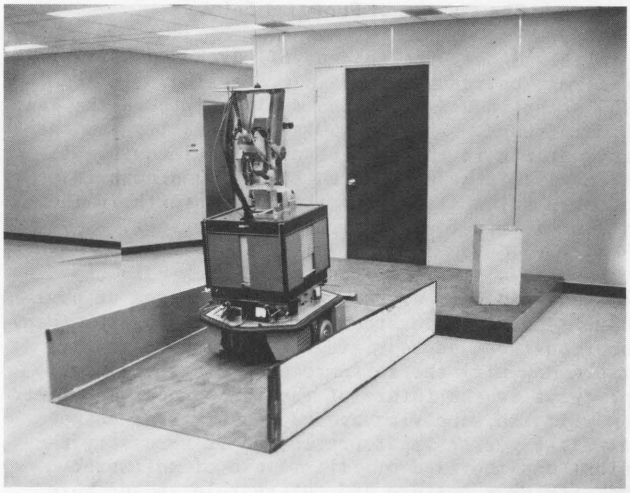
I'm afraid Shakey never achieved the level of competence with which "The Boston Globe" imbued him, as quoted in your September editorial.

Experiments in Planning, Problem-Solving, etc.

Shakey provided a laboratory test-bed for experiments in building integrated automatic planning, problem-solving, perceptual, and language-understanding systems, which could interact with the physical world. Many experiments were performed with Shakey between 1968 and 1972. We learned quite a bit about such things as how a computer can model the physical world; integrate knowledge from several sources; delegate key responsibilities, such as object recognition and route planning, to specialized subroutines; construct, generalize, and apply plans of action; and monitor the execution of plans, with spontaneous recalibration and error recovery as necessary. These technical accomplishments were documented in numerous articles, conference papers, films (the 1972 version, "Experiments in Robot Planning and Learning," is still being circulated), and my book ("The Thinking Computer: Mind Inside Matter"). Thus I consider the Shakey era a successful, productive period in the history of our Center and of the field of artificial intelligence (AI).



Shakey and his sterile, "play–pen" environment.



Shakey executes his CLIMB–RAMP "instinctive" ability.

Functioning in a "Play-Pen Environment"

On the other hand, we made much less progress than various popular press reports might suggest toward the creation of an independent sentient robot capable of meaningful performance in a normal human environment. Responsible scientists consider this intriguing idea premature, probably by at least several decades. At his peak, Shakey could only function in a sterile "play-pen" environment of walls, doorways, carefully painted baseboards (so he could "see" where the walls met the floor), and a few simply-shaped wooden blocks; he had only about a dozen pre-programmed "instinctive" abilities, such as TURN, PUSH, GO-THROUGH-DOORWAY, and CLIMB-RAMP, which could be combined in various ways by the planning programs.

Understanding of Commands

He understood an "English-like" command language of less than 30 words formed into perfectly-typed phrases according to a simple grammar. The scientists who worked on Shakey developed a deep appreciation of how difficult it is to produce a robot even with relatively trivial abilities, let alone the true science-fiction-like independent competence.

(please turn to page 21)

Software Products: The Next Ten Years

Walter F. Bauer, President
Informatics, Inc.
21031 Ventura Blvd.
Woodland Hills, CA 91364

> *"Because of the lack of understanding of the various factors and because of the experience and business knowledge required, there is sure to be a large scale shakeout in the companies offering software products."*

The Concept of Purchased Software

Although new, the concept of purchased software is firmly entrenched in data processing. In the 1950's the industry saw the rapid growth in the sale of mainframes. In the 1960's the industry saw the rapid growth in the sale of peripheral equipment. In the 1970's we are seeing the rapid growth in software sales along with the rapid growth in the sale of terminal and communication devices.

Despite all of the success and growth in the software product business, the entire area of business is poorly understood within the industry. Many questions seem appropriate for discussion at this time. How will the software products field develop? What is the future of the independent supplier? How will the supplying and buying patterns change in future years? What protection is necessary against unauthorized or illicit use of software? What is IBM's posture and future role in software products? Will there be dramatically new technological changes in the next 10 years?

The Idea of the Software Product: 1965 to Date

The idea of the software product is a new idea in a new industry. The data processing industry itself is a little over 20 years old. Within that industry software products are even newer still. In 1965 the first, serious, meaningful products were sold — just a little over 10 years ago. Prior to that time the majority of professionals in the industry believed either that software should be freely available, freely circulated, and in the public domain, or they believed that there was no such thing as a standard piece of software which could be used effectively by a multiplicity of users. In the mid-60's the first software products appeared. They were in the area of system utility software and included products such as the AUTOFLOW product of Applied Data Research, the MARK IV product of Informatics and GIS from IBM. Since then, of course, as we all know, many hundreds of products have been introduced, many of them successfully. Many tens of products have sold more than $1 million worth and a few have sold as much as $30 million.

Statistics

The statistics supporting the claims of software products growth rate are impressive. In 1970 those revenues will reach a level between $400 - 500 million. A directory of software products in 1972 showed 700 companies offering 2,000 products. A recent 1976 directory showed 1400 companies offering 3,000 products. In a little more than 3 years the number of companies and products nearly doubled. It is my prediction that the annual revenue level will go from its present annual level of under $500 million to $1 billion by 1980.

A Powerful Economic Fact

The future of the software product seems assured simply because it is based on a sound fundamental economic idea. The notion that software can be developed by a group of experts, "productized" in the sense of making it easily usable by many users, and then made available to a multiplicity of users makes economic sense because it provides economic leverage — it raises the productivity level of those charged with producing processed information. That simple but powerful economic fact alone will provide impetus for continued development of software products as an enterprise. The economic fact is probably even more important today than it was 5 years ago because of the growing tendency to look at data processing in a hard-nosed way by continuing to challenge and test its cost effectiveness. But in so agreeing as to this economic fact, this leverage, this sound basis, we have only scratched the surface of the many interesting business and technical aspects of software products.

Costs That are Not Understood

For software products to be economically viable as a business enterprise, customers must be willing to pay a price which will allow the supplier to make a profit. In order for the supplier to make a profit, he must thoroughly understand his costs and demand an appropriately high price. The user and the supplier, of course, must act within the framework of competitive forces offering products in a dynamic marketplace. In my view, the market for software products is unfortunately immature in many respects. Users do not understand the costs of the supplier, but more important than that, many suppliers don't understand their costs either. To compound the problem even further, many EDP users still do not believe

---

Based on a talk at the meeting of the American Institute of Industrial Engineers, July, 1976.

that a product developed by a foreign organization can satisfy their needs. Movement is slow — sometimes almost imperceptible — towards the user understanding the benefits of the software product to him.

The Normal Human Frailty of Being Suspicious of Something New or Different

There are some fundamental technological aspects of software which relate to these business aspects of software products. Software is systemic or organic in character basically. In other words very little software which one deals with or uses can be considered as an entity apart from the software environment in which it is to be found. Technological interfaces are very complex. To further exacerbate the problem, software is handled by people who exhibit the normal human frailty of being suspicious of everything new or different. The oldtimers can vividly recall the suspicion with which FORTRAN and COBOL were regarded when they were introduced in the late 1950's and early 1960's. The cogent industry observer will also note and upon reflection be appalled at the slowness with which COBOL became useful in the 1960's. In 1960 when COBOL first appeared, all industry leaders were sure that by 1965 80% of business applications would be programmed in that language. I think it is safe to say it took at least twice that long.

Pressures on the Seller of Software

But let us return to the question of the business aspects of buying and selling software. As stated earlier, I am of the opinion that neither buyer nor seller appreciate the cost of developing and marketing software. The fact that there are no manufacturing or production costs associated with software is quite possibly the fact that confuses or misleads. Because there are no costs of production (except producing a tape) and because other costs are more difficult to understand and estimate, there is a strong tendency of everyone except the most experienced to underestimate the costs or simply assume they do not exist. The seller must produce code which can stand the utmost scrutiny of those paying good dollars for it. It must be well organized and lucid. Documentation must be impeccably prepared leaving no question as to how the user can, must, or should use the software. The problem is amplified by the fact that the buyers are demanding increasingly higher quality code and better documentation and they are also demanding trial periods, benchmark tests and high quality field support. Sellers are forced to increase the quality and quantity of all of these services and factors in order to meet the competition and the demands of the marketplace.

High Marketing Costs

Marketing costs for software products are high. In nearly all cases marketing costs account for 40 - 70% of the total costs for selling the product. If the product is unusually profitable, marketing costs may be as low as 40%. If the product is less profitable, or has a narrow marketplace or is unduly complex, marketing costs may run as high as 70%. Marketing costs are high because software products are complex and require bright people to understand them and explain them to prospects. These marketing people demand high compensation and get it because of the competitive forces. The last few years buyers have increasingly asked for benchmark tests to at least satisfy the request of top management that a good business decision is apparently being made. (Actually, in reality, the benchmark test is all too frequently consciously or subconsciously designed to support the decision to buy the product decided upon before benchmark considerations.) To illustrate the high cost of selling software products it is probably true that if a product can't be sold for $10,000 per copy it probably has to be sold on the basis of a mail campaign. In other words, it cannot be sold by repeated visits of a qualified salesman on a face-to-face basis.

Disappearance of New Companies

Because of the lack of understanding of the various factors and because of the experience and business knowledge required there is sure to be a large scale shakeout in companies offering software products. The general newness of the field and the starry-eyed aspects of offering a new product are, in many cases, carrying the investor and entrepreneur along at the moment. Profit margins which can be regarded as minimally satisfactory are all too seldom seen. It appears inevitable that only a relatively few of the larger or better companies will survive in the software products game.

Penetration of the Market

A point which is sustaining the interest in software products is related to the characteristic that the market is "immature". One of the measurements of the maturity of the market is the penetration in the aggregate (i.e., all suppliers together) of the available market. The canned soup market is mature since all sellers together probably supply over 95% of the available market while the usual case in software is that the aggregate penetration of the available market is closer to 10%. In other words, it is a rarity to see a type of product being used by more than 10% of the users who could conceivably cost-effectively use the type of product.

Application Products

But what of the future of software products? I have already commented on the seeming certainty of continued increase of business activity in software products. The most significant new trend in software products is the rapid increase in application products. System or utility products were the first such products to be marketed. These are products which do not produce an end user result, but are only tools to allow the end result to be accomplished. An application product on the other hand is a software product used to produce an end user result. Application products are in the minority now — the distinct minority. In the next 10 years, however, it appears likely that application products will grow to become at least half the volume by dollar sale and will probably continue to grow faster than system products, becoming the preponderant type product in the latter part of the next decade.

A "problem" of the typical application product is that it normally interfaces with the people and paper procedures of the user organization. Examples of this are the manufacturing production control product and the insurance administrative system product. As such there is considerable time and expense necessary to imbed the software product into the user environment. Whereas many system-type software products can be taken off the shelf and "snapped" easily into place, the application product can be very messy in its installation.

Big Investments

I have already alluded to one aspect of the future, at least as I see it: there will be a big shakeout of software product suppliers. Big investments are increasingly needed and an experienced and highly qualified business approach is necessary. Products will become better and they will become more off-the-shelf in nature. Inevitably they will also become higher priced. This will be a salutary fact for the suppliers and in the long run will help the users as a group since it will tend to build a viable productive industry and therefore a good supply of products from which to choose.

Operating System Products

It seems clear that the next major business step in the software products area will be that operating systems will become unbundled. IBM is apparently already taking the first step toward making operating system software separately purchasable. Theoretically, at least, this will aid those software companies whose primary business is the development and sale of operating system type software.

Protection of Software Products Against Theft

Up to this point the software products industry has been protected from unauthorized and illicit use of the program mostly through the common law aspects based on trade secrets. It is my belief that present protection is adequate for the software supplier. I would hasten to say, however, that we will graciously accept more protection and probably work toward achieving that greater protection. In order for the thief to make a business out of the software product which he has obtained illegally, he must have a qualified technical group to maintain and improve it at central headquarters, he must have a qualified field organization to interface with the "customer" and he must develop a qualified sales organization to further sell it. These latter requirements completely overshadow the mere fact of the code itself which represents only the programming embodiment of the software product. The thief who steals the code has only accomplished 5% of what he needs to accomplish to cash in on his ill-gotten asset.

While there may be some users who would unauthorizedly obtain a software product, it must be concluded that these will be few and far between; the user who is unable to obtain field support and future improvements and modifications to the product will be in a helpless position. Furthermore, the user must involve quite a number of his staff in his nefarious scheme. This is at least awkward, embarrassing and risky.

Nevertheless, some statutory protection seems indicated. It is my opinion that patent protection is probably not appropriate. I fail to see how patentability and infringement can be defined appropriately in the case of a computer program except possibly at the "coding trick" level. On the other hand, I would predict that the copyright laws will be extended to cover the case of the software product to prevent its unauthorized duplication electronically.

Built—In Protection

Another trend in software products is the gaining of more protection of programs by the supplier through a variety of ingenious approaches. Identification numbers corresponding to the mainframes on which the computer program is authorized to work are being imbedded secretly into the program. Increasingly, software products are including a technique which makes the program volatile in nature such that its capability to continue to be used will be depleted with time and usage. The program can be "renewed" from time to time by the supplier who alone knows how this is done.

Transaction – Based Pricing

In a somewhat related way, it would seem probable that a much greater percentage of programs will become transaction-based in the next decade. In other words, increasingly, software products will be priced on a basis of the number of commands executed or the amount of mainframe time used, and the like, instead of being priced on a straight purchase or lease basis.

Competition and Monopoly

I believe that few would argue with the general precept that a freely operating marketplace for software products is important to the user community. Users would have high quality products from which to choose and high quality dependable suppliers who are offering them. In other words, competitive aspects of the marketplace must be present in sufficient degree.

The preponderance of the power of IBM in our industry must always be regarded as a significant potential threat to that competition. There is no question that IBM will become increasingly active in software products in future years and may even come to regard software products as an important business in its own right; that is, more than just an adjunct to the sale of hardware. With this in mind, it is easy to conclude that the current and pending litigation against IBM by the government and other organizations is important to this future of software products. If a consent decree, for example, is negotiated it will undoubtedly have the provision that IBM must provide software interface specifications on a complete and timely basis. This would go a long way toward insuring a continuing good marketplace for software products and a continuing attractive atmosphere in which the user can select those products which he regards most highly.

On the other hand, if IBM categorically wins all its lawsuits and no consent decree is negotiated, it would seem that the orderly and competitive marketplace for software products might well be greatly compromised. In this case there will be serious implications of all kinds to the data processing industry from both hardware and software standpoints. Indeed, it is difficult to perceive in this case how a data processing industry can exist.

Firmware

There is no question that eventually "firmware" will come into the picture. Future generations of computers will undoubtedly take advantage of the increasing cost efficiency and flexibility of semiconductor technology to develop plug-in modules which will accomplish much of what the software product is accomplishing today. Because this will require a totally new architecture for computers and because of the normal inertia of the marketplace, the industry is probably 10 - 15 years away from meaningful offering in "firmware". It would seem reasonable that the first such firmware to be offered would be operating system type software which is closest to the hardware. Next in likely order would be re-

(please turn to page 21)

Microprocessors in Computer Education

Dr. Lance A. Leventhal
Engineering and Technology Dept.
Grossmont College
El Cajon, CA 92020

> *"Microcomputers perform the same functions as larger computers and offer a tremendous variety of architectures. The availability of complete working systems for a few hundred dollars means that even small schools can afford to give their students a range of computer experience."*

The Cost and Complexity of Computers

Computer education has always been limited by the cost and complexity of the equipment involved. Students have been restricted to direct experience with one or two computers, often in a very limited way, even when large computer centers are available. Small educational institutions with limited technical facilities have found the use of computers difficult and expensive. However, the development of microprocessors has introduced a new dimension in computer education. Complete microcomputers can be purchased for a few hundred dollars and used and maintained without the need for great technical skill. In this article, we will explore the nature of these devices, their use, and the features that make certain systems particularly attractive.

Almost every educational institution, whether it is a high school, junior college, college, university, or trade school, has considered using computers and training students in their use. Not only are computers involved in almost every part of our society, but they are also a topic of great interest to students and the basis of many job opportunities. The need and demand for computer science education are very great.

The Problems with Large Computers

However, the problems involved in computer education soon become equally clear. Large universities can afford large computers, but such large computers require trained specialists to operate and maintain them. Students cannot be allowed to have direct access to such complex and expensive equipment. Furthermore, only one large machine is generally available; students cannot have much more than theoretical access to the variety of computer architectures that now exist. A dichotomy exists in computer science courses at most universities — textbooks and lectures deal with the general features of computers but the actual hands-on experience is specific to the particular machine the students get used to; so students often have trouble separating general background from specific features.

Clearly, however, some computer experience is better than none at all. Smaller schools have often found the providing of any experience difficult. Even minicomputers are relatively expensive; a usable configuration costs at least ten thousand dollars. Considerable expertise is required to make such computers work and keep working. Often a purchased or donated minicomputer will sit unused or inoperative at a school because no one has the time or skill necessary to maintain the computer. The experience provided for the student is, at best, still limited to one particular machine.

Rapid technological change has made the problems of computer education even more difficult. A computer purchased a few years ago may be obsolete. New computers seem to appear on the market at a dizzying pace; few guidelines exist as to which architectures will prove to be the most significant or even which manufacturers will survive.

The Development of Microprocessors

The development of microprocessors has greatly reduced the minimum cost of computers and made available many simple self-contained units. A microprocessor is the central processing unit (CPU) of a small computer, usually implemented in one or a few integrated circuits (such complex integrated circuits belong to a class called large-scale integration or LSI)/1,2,3,4/. When memory and input-output circuitry are added to the microprocessor, the result is a full-fledged computer or microcomputer. Microcomputers are available for a cost of $100 to $1000 depending on the amount of memory, input/output circuitry, and other features that are included. Microcomputers can be purchased from semiconductor and computer manufacturers and from a large number of companies which specialize in training and in hobby systems /5/. See Table 1.

Microcomputers are not as powerful as minicomputers. But this is not a major problem in the educational environment. The important fact is that microcomputers perform the same functions as larger computers and offer a tremendous variety of architectures/6,7/. The availability of complete working systems for a few hundred dollars means that even small schools can afford to give their students a range of computer experience. These systems are no more expensive than a standard typewriter and not much more difficult to use and maintain. The potential for such systems is enormous.

Variety

The variety of architectures that are available in inexpensive systems is already remarkable. The mic-

Table 1
Manufacturers of Educational and Hobby Microcomputers

Manufacturer	Microprocessor Used
E and L Instruments Derby, CT	Intel 8080
EBKA Industries Inc. Oklahoma City, OK	MOS Technology 6502
Electronics Product Associates, Inc. San Diego, CA	Motorola 6800
Gnat Computers San Diego, CA	Intel 8080
IMS Associates San Leandro, CA	Intel 8080
Martin Research Northbrook, IL	Intel 8080
MITS Albuquerque, NM	Intel 8080 Motorola 6800
MOS Technology Norristown, PA	MOS Technology 6502
Mycro-Tek Wichita, KA	Intel 8080
Ohio Scientific Instruments Hiram, OH	MOS Technology 6502 Motorola 6800
Pehaco Corp. Los Altos, CA	MOS Technology 6502
PCM Corp. San Ramon, CA	Intersil 6100
Southwest Technical Products San Antonio, TX	Motorola 6800
Sphere Corp. Bountiful, UT	Motorola 6800
Wave Mate Gardena, CA	Motorola 6800

NOTE: Many of the largest manufacturers of microprocessors and microcomputers are not included in this list because their products are intended for industrial applications.

roprocessors that have been used or announced in hobby and educational systems are described in Table 2. Since a recent directory of microprocessors lists 29 devices /4/, more architectures will probably soon be available in the educational and hobby markets. Among the more prominent microprocessors in these markets are the following /8/9/10/:

1. Intel 8080, an 8 bit CPU which makes heavy use of indirect addressing. It also has a memory stack. Systems based on this processor include the popular MITS Altair 8800. A great deal of software is available including Basic and Fortran.

2. Motorola 6800, an 8 bit CPU (based roughly on the Digital PDP-11) which has an index register and handles input and output devices just like memory locations. Systems based on this processor are available from many sources and some software is either available or planned.

3. MOS Technology 6502, another 8 bit CPU which is closely related to the Motorola 6800. The CPU alone costs just $25.00.

4. National PACE, a 16 bit CPU with an architecture and instruction set similar to the Data General NOVA.

5. Intersil 6100, a 12 bit CPU which is software compatible with the widely used Digital PDP-8 minicomputer. (Unfortunately, the device is not I/O compatible with the PDP-8).

The systems that are available generally fall into three categories, although these are by no means exclusive: (1) systems based on traditional control panels or other configurations of binary switches; (2) systems based on an external I/O device, usually a teletypewriter; (3) systems based on calculator-like keyboards and lighted displays.

Hobby and educational systems are available in much more elaborate configurations involving printers, cassettes, cathode-ray-tube displays, and other peripherals which can cost thousands of dollars. We will not be concerned with such expensive systems but rather with the minimal systems which are available at the lowest prices.

Switch Operated Systems

The switch-based microcomputer systems look much like classical computers with their standard front panel. Switches can be used to start, stop, or sinlge step the computer and to enter programs and data and observe the contents of memory locations. No additional peripherals are needed if one is willing to enter everything in binary and read the results from the lights. Such a method is obviously tedious, inconvenient, and error-prone. Microcomputers systems which must be handled in this way are undesirable.

Teletype Operated Systems

Most systems, however, can be operated from an external device such as a teletypewriter. Some microcomputer systems depend on the external device and omit the relatively expensive and cumbersome control panel. The hardware and software required to communicate with the external device is moderately complex; a serial interface is needed as well as a monitor which will control the comminications. Clearly the most convenient situation is one in which the hardware interface is part of the microcomputer system and the necessary software is included in a read-only memory. The microcomputer system then only has to be connected to the external device in order to be operational. If either the hardware or software are not included in the system, the user will face a significant initial hurdle.

Even if the hardware and software are included with the system or can be designed by the user, the system is still dependent on the external I/O device. Since even a simple teletypewriter costs $1000, this fact limits the usefulness of such microcomputer systems. Of course, teletypewriters can be rented or shared but the additional cost and inconvenience are still large. The teletypewriter is easy to use and can produce complete listings but its cost is a limiting factor.

Table 2
Characteristics of Microprocessors

Processor	Word Length (Bits)	Typical Instruction Execution Time (us)
Fairchild F-8	8	2-5
Intel 8008	8	12.5
Intel 8080	8	2-5
Intersil 6100	12	5
MOS Technology 6502	8	2-5
Motorola 6800	8	2-5
National PACE	16	10
RCA COSMAC	8	6
Signetics 2650	8	5

Keyboard and Display Systems

A compromise between the inconvenience of control panels and the expense of teletypewriters is the keyboard and display based system. These systems are operated much like electronic calculators. Data, addresses, and programs are entered (usually in hexadecimal) from the keyboard and the results are shown on the lighted displays. The displays are generally the seven-segment type used in calculators which can show some letters as well as digits. More elaborate displays are sometimes available at somewhat higher cost.

Entering programs and data from a keyboard is clearly much more convenient than entering them from switches. The displays show what was entered in a fairly readable form. The problems are that long programs are very time consuming to enter and complete listings cannot be obtained. These problems are not usually serious in course work where long programs are seldom written and extensive listings are not necessary. The keyboard and display based systems are easy to use and do not require expensive I/O devices or elaborate cases.

Keyboard Display and Interface Systems

The most convenient systems for educational use combine the keyboard and lighted display with the interface for an external I/O device. Such systems can be used on their own with the keyboard and display but can also be attached to a shared teletype or other I/O device when longer programs are involved or listings must be obtained. Microcomputer systems which can be used in either manner are convenient both for introductory instruction and for more elaborate projects.

Evaluation

In evaluating microcomputer systems, the following questions should be asked:

1. Does the system come assembled or in kit form?

Kits are cheaper but require some shop facilities.

2. Does the system include all the parts needed to make it operational?

Power supplies, memory switches, lights, and other parts may not be included.

3. Can the system be operated easily without an external I/O device?

Keyboards and lighted displays make independent operation simple.

4. Does the system have a ROM (read-only memory) based monitor which can be used to enter programs and display results?

A microcomputer system will be very hard to get started otherwise.

5. Does the system include the interface for an external I/O device such as a teletypewriter?

Other interfaces, such as for a CRT or cassette, will also be helpful.

6. Does the system include enough memory to be useful?

A few hundred words is sufficient for many purposes although more may be needed for larger projects.

7. What kind of software and specially designed peripherals are available?

8. What debugging features does the system have?

Single step and breakpoint facilities are desirable.

Some Comments on Software

We have not discussed the problem of software aids for these systems nor will we do so extensively. The simple systems which we have described are intended to be programmed in machine language. Assemblers, compilers, and interpreters, even when available, require additional memory and I/O devices. The further development of ROM-based software will change this situation somewhat /11/. Such software will not require very much read/write memory and will not have to be loaded each time it is used. ROM-based assemblers and compilers will become widely available in the next few years and systems based on such software will offer even more opportunities in computer education than current microcomputer systems.

One method that can be used to make higher-order languages available for microcomputers is a cross-development system based on a minicomputer. Cross assemblers and cross compilers are available for popular microprocessors; these programs are mostly written in Fortran and can be run on any standard minicomputer. Thus a single minicomputer can be used to assemble or compile programs for a variety of microcomputers. Cross-assemblers can often be obtained at special educational discounts from the manufacturers or can be written in standard formats /12/.

Two Typical Microcomputers

In the remainder of this article, we will discuss two keyboard and display-based microcomputers — the Electronic Product Associates Micro-68 (see Figure 1) and the MOS Technology KIM (see Figure 2). We will describe the features and use of the systems in detail with an emphasis on what characteristics are particularly useful or convenient. Both of these systems are inexpensive, self-contained and easy to use. Similar systems can be obtained from many of the sources mentioned in Table 1 and the systems described should not be assumed to be superior to others that may be available. An instructor who plans to use microcomputers should examine the latest trade publications since new manufacturers and new improved systems are continually being announced.

EPA Micro 68

The Electronic Product Associates Micro-68 is a complete microcomputer in a small desktop case. The system is ready to run as soon as it is plugged into an ordinary electrical output. The basic system contains the following:

1. A 16 button hexadecimal keyboard — some of the keys serve two purposes as will be descrbed later.

2. A 6 digit LED display — this display is sufficient to show an address consisting of two hexadecimal digits (8 bits).

3. The Motorola 6800 microprocessor.

4. 128 8-bit words of RAM memory with on-board expansion possible to 768 words.

5. 512 words of read-only memory containing a system monitor. The system will accept another 512 words of read-only memory plus a monitor for teletypewriter communications.

6. A power supply.

7. A 16 bit I/O connector.

Other options that are available include a teletypewriter adapter, an 8K memory board, and a carrying case. The base price is $430.00 and an increment of 128 words of read-write memory is $5.50.

The microcomputer can be reset from a pushbutton which automatically transfers control to the monitor. Programs and data are entered from the keyboard and a memory address and its contents can be shown on the lighted display. The following keys are used as command keys in the entry process /13/.

1. "E" for examine memory — a 4 digit address is entered and the contents of that location are displayed. The "F" (forward) key allows the user to display the contents of the next location, "B" (backward) the previous location.

2. "C" for change memory — a 2 digit data word may be entered into the address which is displayed. The "A" (automatic) key automatically increments the address after the data has been entered and clears the next memory location, thus allowing an entire program or block of data to be entered.

Once the program and data have been entered, the program can be run by pressing the "D" (DO) key. The starting address is then entered and the program is executed.

Trap Instruction

The Motorola 6800 has a software interrupt, or trap, instruction which causes the processor to save

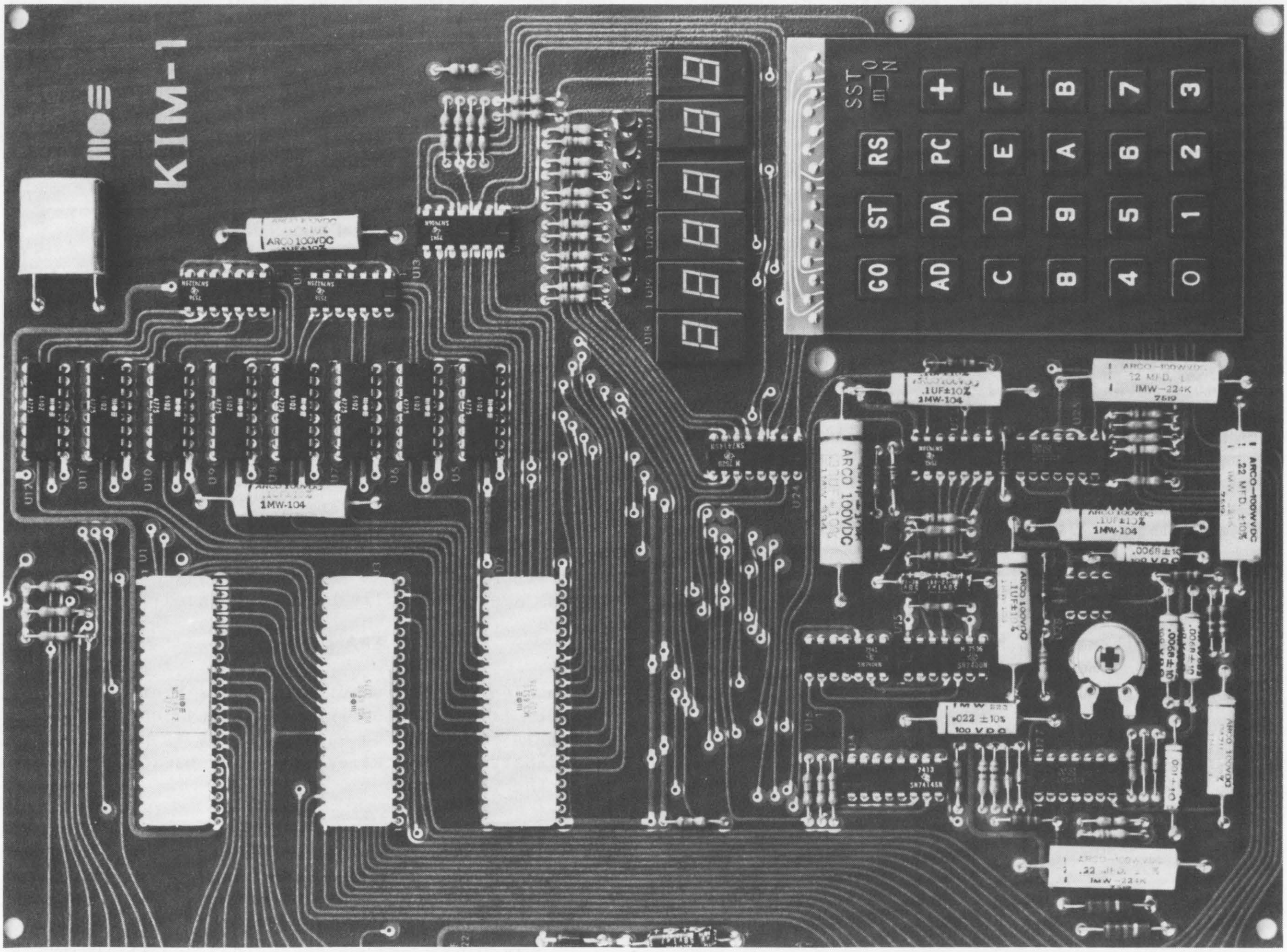


Figure 2

the contents of all registers in the memory and branch to a specified destination. This instruction is very useful for debugging since the contents of the registers cannot otherwise be observed (the registers are inside the single-chip monolithic processor and do not have unique connecting pins). The software interrupt instruction in the EPA Micro-68 returns control to the monitor program; it can therefore serve as a breakpoint so that intermediate results can be checked. Control can be returned to the user program by pressing the "8" key; a return from interrupt instruction is then executed which reverses the actions of the software interrupt. A software interrupt instruction is also placed at the end of the user program in order to return control to the monitor. The lack of a single step mode makes the software interrupt instruction particularly important.

MOS Tech KIM

The MOS Technology KIM is an even more elaborate microcomputer at a surprisingly low price. Unlike the EPA Micro-68, the KIM is an unpackaged board and requires a power supply. The basic system contains the following:

1. A 23 button keyboard — all keys are single purpose.
2. A 6 digit LED display.
3. The MOS Technology 6502 microprocessor.
4. 1K 8 bit words of read/write memory.
5. 2K words of read-only memory containing the keyboard and display monitor plus the programs needed for a teletypewriter and an audio cassette unit.
6. Interface and control circuitry required for the teletypewriter and audio cassette unit.
7. 30 programmable input/output pins.
8. Two interval timers.
9. A 44-pin connector.

The base price of the system is just $245.00. The power supply, however, is somewhat complex since it requires 1.2 A at +5 volts and 0.1 A at +12 volts and must be regulated to within 5%. The +12 volt supply is only needed for the audio cassettes. The teletypewriter and audio cassette connections are simple and clearly described in the KIM User's Manual /14/.

Programs and data may be entered into the KIM microcomputer and run in a manner similar to that described for the EPA Micro-68. A simple step mode is also present under the control of a slide switch. The audio cassette interface provides a fast cheap way to store long programs; a cassette unit costing under $50.00 is satisfactory for use with the KIM.

Enthusiasm

Microcomputers like the EPA Micro-68 and MOS Technology KIM obviously provide tremendous capabilities at very low cost. Such systems can be set up, used, and maintained without special training or constant attention. With these systems, students can have the opportunity to work closely with many different computers. The potential for such devices in education is tremendous and the enthusiasm with which students use these microcomputers can only add to their value.

Summary

Computer education has been limited by the cost and complexity of computer equipment. However, microprocessors have made complete computer systems available with a variety of architectures at a cost of a few hundred dollars. A keyboard and lighted numeric display can make such a computer system convenient for running small programs without external peripherals. Students can thus get direct experience with many different machines at a relatively low cost. The microcomputer systems are cheap, simple, easy to use and maintain, and fun to work with.

Acknowledgements

Charles Peddle and Donald McLaughlin of MOS Technology and Victor Wintriss of Electronics Product Associates helpfully provided pictures, manuals and other information about their systems. Colin Walsh, Robert Golden, William Tester, and John Crane offered many welcome suggestions.

References

/1/ Torrero, E.A., "Just Getting into Microprocessors?" Electronic Design 24, January 5, 1976, pp.46-50

/2/ Gilder, J.H., "All About Microcomputers," Computer Decisions, December, 1975, pp. 44-49

/3/ Cushman, R.H., "EDN's First Annual Microprocessor Directory," EDN 19, November 10, 1974, pp. 31-41

/4/ Cushman, R.H., "EDN's Second Annual Microprocessor Directory," in EDN Microprocessor Design Series, Volume 2, pp. 29-47

/5/ Grossman, R.M., "Microcomputer Systems Directory," in EDN Microprocessor Design Series, Volume 2, pp.14-16

/6/ Ogdin, J.L., "Survey of Microprocessors Reveals Limitless Variety," EDN 19, April 20, 1974, pp. 38-43

/7/ Schoeffler, J.D., "Microprocessor Architecture," IEEE Transactions on Industrial Electronics and Control Instrumentation IECI-22, August, 1975, pp. 256-272

/8/ Sloan, M.E., "Potential Trends in Microcomputer Education," IEEE Compcon 74 Proceedings, pp. 131-134

/9/ Leventhal, L.A., "Education in Microprocessors and Microcomputers," submitted to IEEE Transactions on Education

/10/ Leventhal, L.A., "Fixed Instruction Set Microprocessors," to appear in Simulation, May, 1975

/11/ Wulfinghoff, D.R., "Market Factors Portend Design Changes in Small Computers," Computer Design 14, August, 1975, pp. 81-87

/12/ Conley, G.W. "Portable Microcomputer Cross-Assemblers in BASIC, Computer 8, October, 1975, pp. 32-42

/13/ "Micro-68 User's Manual," Electronic Product Associates Inc., San Diego, California, 1975

/14/ "KIM-1 Microcomputer Module User Manual," MOS Technology Inc., Norristown, Pennsylvania, 1976

□

The Handling of Natural Language by a Computer — Part 2

Bertram Raphael
Stanford Research Institute
Menlo Park, CA 94025

"The problem of how to represent general knowledge, faced by the computer scientist, is very much like the problem of how to repremeaning, faced by the linguist."

Programming Computers to Understand English

As we saw previously, linguists are actively working on theories to explain the nature of language and its semantics. However, technology does not usually wait for theories to be completed. While the linguists carry on their theoretical studies, computer scientists have also been studying how computers can be made to understand natural language. These studies have been conducted from an experimental, engineering point of view. In each of the projects to be described here (and many other similar efforts), a scientist has developed a computer program to demonstrate an automatic language-processing capability more powerful than any previously demonstrated. None of these systems is yet powerful enough for general practical use, although some of the more recent ones are very close. Still, by observing the problems encountered by these scientists as they attempt to increase the effectiveness of their programs, we can learn something about the present limits of automatic language processing; and by studying the novel programming techniques that went into achieving the impressive performance of several of these systems, we can learn something about how to move beyond the current limits.

Question Answering Systems

Many of the experimental language-processing programs fall in a general category called question-answering systems. A question-answering system may be defined as any computer program that understands by answering questions about the information. The ideal question-answering system should be able to: (1) accept facts and questions, and make appropriate responses, all in the form of natural English; (2) store, remember, and make efficient use of a large amount of data — at least thousands of elementary facts; (3) answer questions that require it to figure out the logical consequences of the facts stored explicitly in its memory; and (4) operate conversationally — e.g., via a time-sharing computer terminal — without frustrating delays. Although no system yet developed has all four of these capabilities, a significant degree of success in each of the four areas has been separately achieved by various systems. In the next few years we should begin to see these capabilities combined and improved, producing the first true, complete question-answering systems.

The following is from Chapter 6 of "The Thinking Computer" by Bertram Raphael, published by W.H. Freeman and Co., San Francisco. The first part was published in "Computers and People" for September, 1976.

Question Answering in Restricted Domains

When scientists encounter a problem that is too difficult for them to solve directly, they usually look for simpler versions or special cases of the problem to work on first. Then, if they are successful, the solutions to these restricted problems can be used as guides to solving the larger original problem. Because the development of a complete, ideal question-answering system is beyond the ability of current technology, computer scientists interested in this area have defined various simpler versions of the problem to work on first.

Limited Complexity

One way of restricting the question-answering problem is to limit the complexity of the language that the system must handle — e.g., by limiting the size of the vocabulary and the kinds of sentence structures to be given to the system. Although these restrictions are frequently used, they are neither as interesting nor as natural as another kind of restriction: the restriction of the subject matter that the language to be processed can talk about. In fact, restricting the subject matter usually has the side effect of automatically restricting the vocabulary and sentence structure, since we usually use only certain words and certain kinds of expressions when we talk about certain limited subjects. Therefore several question-answering research projects have begun by specifying a limited, well-defined subject domain, and then proceeding to see how effective a system could be built within that constraint.

The BASEBALL System

The first significant effort of this kind was the BASEBALL system, developed at the MIT Lincoln Laboratories in 1960. A fixed, tabular (actually, list-structured) data file contained the month, day, place, teams, and score of every baseball game played in the American League in one complete season. Although the input language was also somewhat restricted, the program was capable of answering almost any

---

Excerpted from Chapter 6, "Natural Language," of *The Thinking Computer: Mind Inside Matter* by Bertram Raphael, published by W. H. Freeman and Co., San Francisco, Calif., copyright 1976, and reprinted with permission.

reasonable questions about this data, ranging from simple questions such as "Whom did the Red Sox lose to on July 5?" to such complex questions as "Did every team play at least once in each park in each month?" The analysis of each question was performed primarily by a phrase-structure grammar that could appeal to special baseball-oriented programs to resolve difficulties. For example, the word "score," in baseball, is usually a noun, as in, "What was the score of the Boston game on August 26?" but the program assumes it to be a verb if no other verb appears in the same sentence, as in, "How many runs did the Yankees score on August 26?" Another fact that simplified the analysis program was that it knew the form of the data base, and therefore was designed to extract from the question only data relevant to the time, place, and so on, of baseball games.

The STUDENT System

The STUDENT program, developed by Bobrow at MIT in 1964, provides a more interesting example of the kind of results that can be achieved by restricting the subject domain. STUDENT takes as its task the solution of elementary algebra problems directly as they are stated in English in high-school mathematics or puzzle books. Once again, the complexity of the input language needed to be restricted, and in fact STUDENT did not accept as broad a range of English constructions as BASEBALL accepted. However, the BASEBALL input was known to be a question about a fixed data base; STUDENT's inputs described algebra problems, and contained not only a particular question, but also all the relevant data. The system's own data base only contained some general knowledge about the world, such as that three feet equals one yard, and that distance equals velocity times time. Here are two examples of problems STUDENT could solve:

1. If the number of customers Tom gets is twice the square of 20 percent of the number of advertisements he runs, and the number of advertisements he runs is 45, what is the number of customers Tom gets?

2. Mary is twice as old as Ann was when Mary was as old as Ann is now. Mary is 24. How old is Ann?

Focus Upon a Goal

The STUDENT program did not use any conventional linguistic analysis method. Instead, it focused upon its known goal: to translate the input problem into a set of simultaneous algebraic equations, which could then be given to an equation-solving subroutine. The system's linguistic analysis, then, consisted of looking for words and phrases that could be replaced by arithmetic operators, constants, or variables. For example, by a series of substitutions that apply to all "age" type of problems, the second of the above examples is translated by STUDENT first into a more rigid form such as — Mary's age is 2 times Ann's age k years ago; k years ago Mary's age was Ann's age now; Mary's age is 24; x is Ann's age — and then into the set of equations:

$$\begin{aligned} M &= 2(A-k) \\ M-k &= A \\ M &= 24 \\ x &= A \end{aligned}$$

which may be solved for x.

Other Experimental Systems

Other restricted-domain experimental systems have been developed for various subjects including calculus, chemistry, airline schedules, and South American geography. In each case the restricted subject matter, the particular data-base structure, and the known manner in which the system would be used, all contributed substantially to simplifying the necessary linguistic analysis. The existence of all these special-purpose systems forms a base of experience that can be drawn upon whenever someone wants to implement a special-purpose system for some new domain. It is likely that the first practical (instead of merely experimental) question-answering systems will be in use before 1980, and that they too will be restricted-domain systems like these. On the other hand, scientists would also like to have a system that understands natural language in a more basic sense, so that one system can eventually be applied in many different domains. Restricted-domain systems are generally so carefully tailored for one particular task that they are difficult to generalize or transfer to another task. Therefore we must look to systems that are designed from the start as experiments in the analysis and respresentation of general knowledge. Several studies of this kind are described below.

Representation and Use of General Knowledge

The problem of how to represent general knowledge, faced by the computer scientist, is very much like the problem of how to represent meaning, faced by the linguist. The linguist is primarily interested in how to translate from the strings of words of natural language to some representation of their meaning, whereas the computer scientist is primarily interested in making use of the data in the meaning representation to control some program. The emphasis in most of the computer science work described below is thus on the representation itself and its use, rather than on how information appearing originally in natural language gets put into the representation. Eventually, of course, both elements — the acquisition and the use of meaningful knowledge — will have to be present in the complete question-answering system.

Semantic Classifications

An early approach to building a computer representation of general knowledge, developed by Quillian at Carnegie-Mellon University in 1965, was based on the idea of semantic classifications. Each word is defined, in the computer's memory, by a network of labeled linkages to other words. This network is called the semantic memory. The linkages specify such properties as inclusion, part-whole, color, size, logical relation to other words, and so on. The resulting network structure can be viewed as an automatic dictionary; but in this dictionary the words are connected to each other by direct pointers, instead of being explained by intermediate English text. To demonstrate the appropriateness of this structure, Quillian could ask his semantic memory system to compare two words, and it would respond with a description of all paths it could find between those words in its memory. For example, when asked to compare "plant" and "live," the system responded with the descriptions of two paths:

1. "Plant is a live structure."
2. "Plant is a structure which gets food from air. This food is thing which being has to take into itself to keep alive."

Although it demonstrated an interesting memory structure, this system was never put to any further use.

The SIR System (Semantic Information Retrieval)

In one of the first studies of how to store and use general information to perform question-answering tasks, the Semantic Information Retrieval (SIR) program developed by Raphael at MIT in 1964, a list-structured memory accumulated facts expressed in terms of a small number of basic relations: inclusion, part-whole, ownership, number, spatial position. The system was able to work with any vocabulary or subject matter; its operation depended solely upon the meanings of the underlying relations, and not at all upon the nature of the things being related. Special deductive subroutines associated with the built-in relations were capable of answering a variety of questions (as the dialog of Table 1 shows). The major problem with this system was that it could not be extended easily. If a new relation were added, new complex deductive programs had to be written, and many of the existing programs had to be changed.

The ELIZA System

Several other experimental question-answering systems were developed during the mid-1960's which avoided the problem of natural-language input. Some were based upon actual English text, such as extracts from an encyclopedia or text typed on a computer terminal by the user, and constructed answers by rearranging and spewing back phrases from this text without ever really considering their meanings. Thus the representation of general knowledge was English text itself. The extreme example of this kind of system is the ELIZA program, developed by Weizenbaum at MIT in 1966. Table 2 is an example of how this program, which often seems to sound like a psychiatrist, can carry on an apparently meaningful conversation. Actually it all works by syntactic trickery, as a more careful interrogation of the program would show. None of the text-based programs could do much significant reasoning in order to answer nontrivial questions. .....

Table 1: Sample Conversation with SIR. (Lines in capitals are generated by the computer.)

```
Every boy is a person.
I UNDERSTAND.
A finger is part of a hand.
I UNDERSTAND.
Each person has two hands.
THE ABOVE SENTENCE IS AMBIGUOUS--PLEASE REPHRASE IT.[a]
There are two hands on each person.
I UNDERSTAND.
How many fingers does John have?
THE ABOVE SENTENCE IS AMBIGUOUS--BUT I ASSUME "HAS"
  MEANS "HAS AS PARTS."
I DON'T KNOW WHETHER FINGER IS PART OF JOHN.
John is a boy.
I UNDERSTAND.
How many fingers does John have?
HOW MANY FINGERS PER HAND?
Every hand has 5 fingers.
I UNDERSTAND.
How many fingers does John have?
THE ANSWER IS 10.
The boy is just to the left of the table.
I UNDERSTAND.
The lamp is just to the left of the table.
THE ABOVE STATEMENT IS IMPOSSIBLE.[b]
The table is to the right of the chair.
I UNDERSTAND.
What is the relative position of a person?
THE LEFT-TO-RIGHT ORDER IS AS FOLLOWS:[c]
     (CHAIR    (BOY    TABLE))
```

---

[a] "Has" might mean either "possesses" or "has attached as parts."

[b] "Just to the left" means "adjacent;" only one thing at a time can be "just to the left" of another.

[c] The inner parentheses around BOY and TABLE indicate adjacency.

Table 2: Conversation with ELIZA. (Lines in capitals are generated by the computer.)

```
Hello.
HOW DO YOU DO. PLEASE STATE YOUR PROBLEM.
I want a good example for my book.
YOUR BOOK.
Yes, my book about computers. Can you help me?
YOU'RE BEING SOMEWHAT SHORT WITH ME.
I didn't mean to be.
YOU SAY YOU DIDN'T MEAN TO BE.
That's right. What do you know about computers?
DO COMPUTERS WORRY YOU?
Not at all. I think they can be very useful.
CAN YOU ELABORATE ON THAT?
Computers play a vital role in the operation of our
  society.
WHY DO YOU MENTION COMPUTERS?
Because that is what the book is all about.
IS THAT THE REAL REASON?
Of course. Sometimes you are difficult to talk to.
WHAT MAKES YOU THINK I AM DIFFICULT TO TALK TO?
You answer in such short, stylized phrases.
WE WERE DISCUSSING YOU - NOT ME.
I thought we were discussing computers.
WHAT DO YOU THINK MACHINES HAVE TO DO WITH YOUR
  PROBLEM?
My only problem is to get you to carry on a
  reasonable conversation.
WHY DO YOU SAY YOUR ONLY PROBLEM IS TO GET I TO CARRY
  ON A RESONABLE CONVERSATION?
You don't use very good grammar.
OH, I DON'T USE VERY GOOD GRAMMAR.
That's right. Don't you know the difference between
  nominative and accusative?
YOU'RE NOT REALLY TALKING ABOUT ME - ARE YOU?
I really was, but I'm willing to change the subject.
DO YOU SAY YOU REALLY WAS FOR SOME SPECIAL REASON?
The reason is that I would like an interesting
  conversation in the book.
IN WHAT WAY?
So that the readers will not get bored with it.
I AM NOT SURE I UNDERSTAND YOU FULLY.
You don't understand very much, do you!
WHAT ARE YOUR FEELINGS NOW?
I feel that I'm about ready to give up.
TELL ME MORE ABOUT SUCH FEELINGS.
The page is almost full now, so I can end this
  interview.
SUPPOSE THE PAGE WERE NOT ALMOST FULL NOW?
Then we would have to continue this silliness a while
  longer.
PLEASE GO ON.
That's all there is to say. Don't you ever get tired?
PERHAPS THAT'S ALL THERE REALLY IS TO SAY.
Yes. Goodbye.
YOU SEEM QUITE POSITIVE.
```

The Artificial PARANOID System

Several current projects that each closely blend syntactic and semantic processing into a single language-understanding system seem promising for the near future. I have already mentioned the augmented transition-network approach and the conceptual-dependency approach. Both of these approaches are under active development and early versions of each have been demonstrated, but in neither have the semantic components yet been fully defined. The transition-network system handles an impressive subset of English sentence structures, but has been demonstrated only in the context of a fact-retrieval system, in which the data resides in a rigidly formatted file of factual information.

A version of the conceptual dependency system has been used in connection with a pyschiatric study into the nature of paranoia. A computer program has been designed (by Colby at Stanford University) to simulate the verbal behavior of a hypothetical paranoid individual. Inputs to the program by the "psychiatrist" undergo a conceptual dependency analysis, the results of which are used by the simulation to produce an appropriate response (Table 3). Although the sample conversation bears some resemblance to that which could be produced by ELIZA, Colby's

Table 3: Interview with the Artificial Paranoid. (Lines in capitals are generated by the computer.)

```
How are you today?
OK.
What is your name?
MY NAME IS FRANK SMITH.
How old are you?
TWENTY-EIGHT.
Are you married?
NO.
What sort of work do you do?
I WORK IN THE POST OFFICE.
Why are you in the hospital?
I SHOULDN'T BE HERE.
Who brought you here?
THE POLICE.
What trouble did you have with the police?
COPS DON'T DO THEIR JOB.
What do they do instead?
COPS ARREST THE WRONG PEOPLE.
They arrest the right ones sometimes.
THE LAW SHOULD GET THOSE ITALIAN CROOKS.
Are you Italian?
MY BACKGROUND IS BRITISH BUT I WAS BORN IN THIS
  COUNTRY.
Do you know any Italian crooks?
I TRY TO AVOID THE UNDERWORLD.
Are you afraid of the underworld?
THE MAFIA ARE THE WORST GANGSTERS.
Are you connected with the mafia?
YOU KNOW, THEY KNOW ME.
How do they know you?
THEY SPY ON ME.
Why do they?
THEY WANT TO GET FALSE EVIDENCE TO FRAME ME.
What do they have against you?
THEY ARE OUT TO GET ME.
What for?
I HAVE INCRIMINATING EVIDENCE AGAINST THE MAFIA.
Are you sure of all this?
YOU DON'T BELIEVE ME?
```

program really does "understand" what is going on — in the sense that an expert could look inside the program and find representations of beliefs, fears, and anxieties appropriate to each point in the conversation. However, both the syntactic-analysis capabilities and the vocabulary (and associated word-meaning diagrams) of the conceptual-analysis system still need considerable strengthening.

The Winograd System

No presentation of automatic language-understanding research today would be complete without a discussion of the impressive and influential project completed by Winograd at MIT in 1971. This system works in a restricted domain defined by the possible activity of a mechanical robot arm that can manipulate simple objects on a table top. (In Winograd's implementation, the arm, objects, and table are simulated in the computer and illustrated in diagrams drawn by the computer's CRT display). The ability of this system to process and interpret complicated English constructions, which is based upon systemic grammar, seems fully adequate for practical use within any restricted subject domain. The fact that the system can carry on a protracted conversation, recognize and use colloquial sentence fragments when appropriate instead of complete formal sentences, correctly interpret pronouns, and assume the most meaningful versions of almost all ambiguous constructions, is particularly impressive. ..... The close interaction between syntactic analysis, semantic analysis, problem-solving ability, and observation of the subject domain (in this case, the simulated robot and its table-top "blocks world") appears to be a key to the success of this system.

Another important innovation of this system is the use of procedures, rather than static data structures, to represent knowledge. The dictionary contains separate little computer programs for each word, whose executions check whether the words are being correctly used. The syntactic-analysis system contains separate little programs that "know" about each structural form. Each proposed action of the robot invokes a program that changes the simulated world to determine the effects of the action. This complex network of programs was made possible by the new programming language, PLANNER, developed at MIT especially for projects like Winograd's. PLANNER provides a framework into which separate, special-purpose programs may be inserted without the programmer concerning himself with all the possible ways these programs may eventually interact.

Copies of the Winograd system have been moved to several other research locations where attempts are being made to extend its language-handling capabilities and to apply them to new subject domains. Whether or not these specific attempts turn out to be successful, the ideas of procedural representation of knowledge and close semantic-syntactic interaction are sure to be pursued for years to come.

Speech – Understanding Systems

Thus far in this chapter on natural language, we have assumed that the language would be presented to the computer in easily machine-readable form —e.g., by typing it on a computer-terminal keyboard. Yet for most people, including me, typing does not seem very natural; I would much rather communicate with a colleague, an audience, or a computer by speaking. Early in this chapter I pointed out that one reason for our wanting computers to understand natural language is so that people will be able to communicate with computers as easily as people communicate with each other. Not only is spoken communication much more convenient than typed communication, but the technical mechanism for long-range spoken communication already exists: virtually every home and office has a telephone; few have a teletypewriter.

What are the prospects that computers can be made to understand spoken English? Machine recognition of spoken words has been one of the problems pursued since almost the beginning of computer science, and the results have been frustratingly poor. Since 1971, however, there has been a spurt of optimistic activity aimed at the apparently more difficult problem of the automatic understanding of complete sentences and conversations, and significant progress has been made. Let us review briefly what has been happening.

Interpreting Electrical Output

Speech is a sound — a physical phenomenon consisting of waves traveling through the air. The ear is a mechanism that detects these waves, translates them into nerve impulses, and passes them along for the brain to measure and interpret. Similarly, a microphone is a mechanism that detects such waves, translates them into electrical signals, and passes them along to whatever device we wish to provide, to measure and interpret. Usually, the devices we provide do not interpret the output from a microphone; they merely record it on tape or a disc, or amplify it to fill an auditorium, or perhaps translate it to yet another form such as radio waves. However, we know that virtually all the information that is present in the original speech is also present in the electrical output from a high-quality microphone. The speech-recognition problem, then, is to design a mechanism that can correctly interpret that electrical output.

The Watermelon Box

As a demonstration that such mechanisms could indeed be built, one acoustics (sound) engineer more than twenty years ago built a "watermelon box." This was a box of electronic equipment with a microphone in front and a red light on top. Whenever, in the course of any conversation, anyone in its vicinity uttered the word "watermelon," the light would flash. This was the sole purpose and ability of the watermelon box: to blink whenever it heard "watermelon" mentioned!

Now, there are only about 10,000 English words commonly used in conversation. Does this mean that we need only build about 10,000 "watermelon boxes," one for each word, and we shall have a complete speech-recognition system? Unfortunately, no. It seems that the word "watermelon" is a particularly easy one to recognize. The word "watermelon" seems to be the only word in English that contains the four successive distinct, evenly spaced vowel sounds represented by the letters "a," "e," "e," and "o" in that order. The box merely had to be able to detect these four sounds and watch for them to occur in a correct time sequence. Few of the other 9,999 or so words that we might need are this easy to test for.

For about twenty years, many research teams tried to build bigger and better watermelon boxes. Extremely large, complicated, expensive systems were developed, using both analog and digital computer technology. They had circuits for detecting many vowel sounds, consonant sounds, pitch levels, loudness, overtones, time durations, and a great many other technical characteristics of speech sounds. Yet, the most these systems were ever able to recognize was about 100 different words — and then only if the words were carefully pronounced, one at a time, by someone to whose particular voice the system had been tuned.

N-eye-un

Do these disappointing results mean that automatic speech understanding is not possible? Absolutely not, because, as we have recently been coming to realize, the problem of recognizing many isolated words purely by their sounds — let's call it the super-watermelon-box problem — is quite different, and probably substantially more difficult, than the problem of understanding at least the gist of a running conversation! In fact, human beings are not very good super watermelon boxes themselves. Telephone operators are trained to say "n-eye-un" rather than "nine" because, when listening to a meaningless string of numbers over a low-sound-quality telephone system, people are prone to confuse "nine" with "five" — because these two numbers have such similar vowel sounds. Similarly, some German-speaking people count "eins, zwo, drei" rather "eins, zwei, drei." When someone is speaking to you during a crowded, noisy party, do you usually understand every word? How often can you follow what is being said without having to ask for a repetition? When someone introduces himself to you, don't you find it difficult to hear and recognize the name, especially if you were not previously familiar with the voice?

The Elements of Meaning

Trying to build a superwatermelon box to understand English speech is very much like trying to build a super-syntactic-analysis system to understand English text; it overlooks the elements of meaning and context that seem to be crucial for understanding natural language. In 1970 a committee of computer and acoustics scientists launched a program aimed at integrating acoustic techniques, such as those developed in the various super-watermelon-box projects, with the latest linguistic and question-answering techniques, such as those developed by Winograd. The goal is to develop a speech-understanding system that contains closely coupled syntactic, semantic, and acoustic components, all tailored to a restricted subject domain. This target system, which is intended to be demonstrated by the late 1970's, is supposed to be able to carry on a reasonably meaningful conversation with a vocabulary of at least a thousand words. The purpose of this system is to be an experimental prototype for useful, economically practical speech-understanding systems of the 1980's.

The Nature of the Input Data

The principal difference between the problem of developing a speech-understanding system and the problem of developing a question-answering system lies in the nature of the input data: question-answering input consists of complete, uniform, correctly typed English; speech input consists of a sound stream of varying clarity and intelligibility. As a result, the normal approach of a question-answering system, which is to begin at the beginning and process a sentence in a smooth, left-to-right fashion, is not suitable for speech. Instead, the system must accept a whole statement or utterance and then "think" about it. Which part of the utterance was clearest? Which word was pretty definitely understood? What kind of words are likely to come before, and after, that word? Does that phrase make sense, in terms of the current situation? And so on. Just as current question-answering systems do not first make a list of all possible syntactic analyses of a sentence, current speech-understanding systems do not first make a list of all possible words and phrases that the input sounds like. Instead, they first use their knowledge of the context of the current conversation to predict what words are likely to occur; and then look at the sound data to verify these predictions.

Chess: "That was a Better Move"

One of the first versions of a speech-understanding system, developed in 1971 at Carnegie-Mellon University, went a bit too far in this direction. The restricted domain for this system was the game of chess; a human player was to play against a computer chess program by speaking his moves to the speech-understanding program. However, when the human said "Pawn to queen four," the computer recorded his move as "Pawn to king four." The reason was that the watermelon-box portion of this program could not yet distinguish very well between the sounds of the words "queen" and "king;" since it was not sure which had been said, the computer looked at the game situation and decided that the king move was a better move, in its opinion, so that's what the player must have asked for.

The Outlook is Promising

By 1975, at least two versions of experimental speech-understanding systems were under development, in addition to the "chess" system. One, a joint effort of two California laboratories, was based initially upon a version of Winograd's language-analysis system, but has been drastically modified to operate in a prediction and verification manner, beginning at any convenient point in the input speech stream. Another system, being developed in Massachusetts, is based upon the augmented transition-network approach to language processing, but also highly modified to fit the requirements of speech input. Systems already exist (in 1975) that have vocabularies of more than a hundred words, that demonstrate the ability to integrate syntactic, semantic, and acoustic subsystems (although still very slowly), and, perhaps most important, that appear to be extensible to handle the required 1000-word vocabulary without encountering major new technical difficulties. The outlook for significant automatic speech understanding within a few years is extremely promising. .....

Editorial Note: The sign ..... indicates a significant omission. Interested readers are urged to read the entire chapter in the book.

Suggested Readings

Chomsky, N. Syntactic Structures. Mouton, The Hague, 1957

Fillmore, C., "The Case for Case." In Universals in Linguistic Theory, E. Bach and R.T. Harms (eds.). Holt, Rinehard & Winston, New York, 1968.

Newell, A., et al. Speech Understanding Systems: Final Report of a Study Group. North Holland Publishing Co., Amsterdam, 1973.

Schank, R.C., and K.M. Colby (eds.). Computer Models of Thought and Language. W.H. Freeman and Company, San Francisco, 1973.

Simmons, R.F. "Natural Language Question-Answering Systems: 1969." Communications of the Association for Computing Machinery, Vol. 13, pp. 15-30, 1970.

Winograd, T. Understanding Natural Language, Academic Press, New York, 1972.

□

Bauer – *Continued from page 10*

placement of other system software such as utility software. Finally, applications become candidates for implementation in firmware.

What will be the role of software companies in this future regime of firmware? There is no reason why software companies cannot purchase the semiconductor hardware and offer it as a hardware/software add-on. In other words, the product would have to be hardware compatible as well as software compatible, electrically compatible as well as program compatible.

With IBM the dominant supplier in this industry, this could only happen, of course, if IBM either allowed it to happen or were required by law to allow it to happen. This again relates to the question of the current litigation and the possibility of a negotiated consent decree.

A Revolution in the Sources of Programming Instructions

At least one prediction seems certain: the number of computer program steps being executed today out of software products will greatly increase in the next 10 years. If today 10 - 15% of the executed instructions represent those from purchased software, in the next 10 years that figure will undoubtedly rise to the 75 - 80% region. In other words, a quiet revolution in the business patterns of software is taking place at the present time. That revolution is providing challenges and opportunities for users and suppliers alike.

□

Forum – *Continued from page 7*

What a New-Generation Shakey Would Be Like

Suppose the development of Shakey had not been abandoned, due primarily to changes in the funding situation, in 1972; where would this mobile robot experiment be today? For one thing, the new generation Shakey would be a somewhat smaller, more stable vehicle, with one or two arms for picking things up, an on-board display and keyboard for typed communication, the ability to understand simple spoken commands, headlights and visual sensors for detecting color and distance, non-contact proximity sensors (instead of wire feelers), and on-board mini- and micro-computers as well as a radio link to a large computer.

Do Simple Errands . . .

He would be able to wander down hallways in an office building, perhaps use the automatic elevator, and carry out, with some human assistance, simple errands like fetching supplies or delivering mail.

. . . But Not Cope with Complexity

But he would still be limited to a narrow, well-understood set of capabilities. He would not be able to cope with the complexity of typical office clutter, or with unanticipated surprises such as rearranged furniture. He would often misunderstand instructions, even in his domain of expertise, unless they were phrased by a specialist in his particular English-like vernacular. And for him to function, he would still require the full attention of a million-dollar computer system for long periods of time.

Intellectual Progeny of Shakey

Although the original Shakey is rusting away and dripping some mysterious fluid onto the floor in the corner of my office, his intellectual progeny lives on in a number of places. Jet Propulsion Laboratory is working on the prototype of a robot vehicle that could one day leave a Viking spaceship to explore the Martian horizon. Programmed industrial manipulators of ever-increasing sophistication are beginning to take over dangerous or unpleasant jobs in factories. Computer-controlled carts already improve the efficiency of a number of hospitals and warehouses. Sophisticated computer-controlled wheelchairs and other prosthetic devices give handicapped people a new range of abilities. And a rubber-tired research vehicle in the woods outside a Moscow laboratory, shown in a film at a 1975 AI conference, bears a striking resemblance to Shakey.

The Evolution of Robots

Robots will continue to evolve, both as useful devices and as research tools; but the pace will be slow and the capabilities limited for a long time to come.

□

Newsletter – *Continued from page 26*

ample, a lawyer with LEXIS can evaluate a judge's past decisions on an issue similar to that in his client's case.

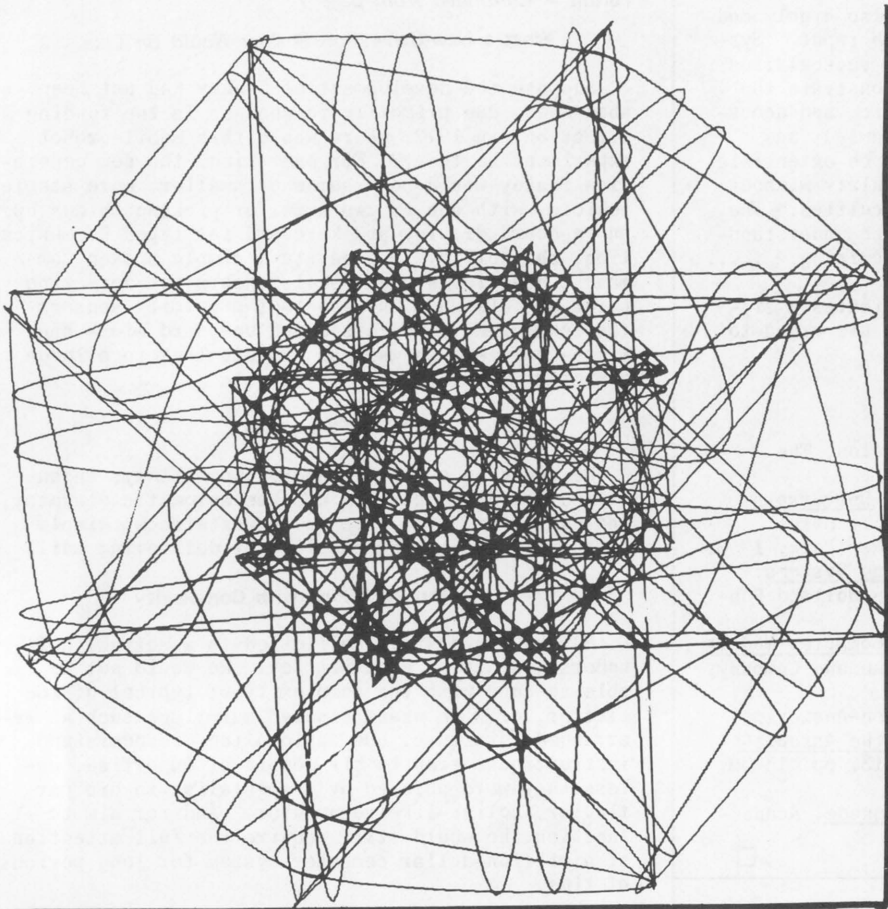
Mr. Rubin commented, "LEXIS simplifies and speeds the process and makes for more careful and exhaustive research than is possible with indices and digests of the law. The service is not only an invaluable aid to lawyers, but benefits the public as well." □

1976
COMPUTER
ART
EXPOSITION

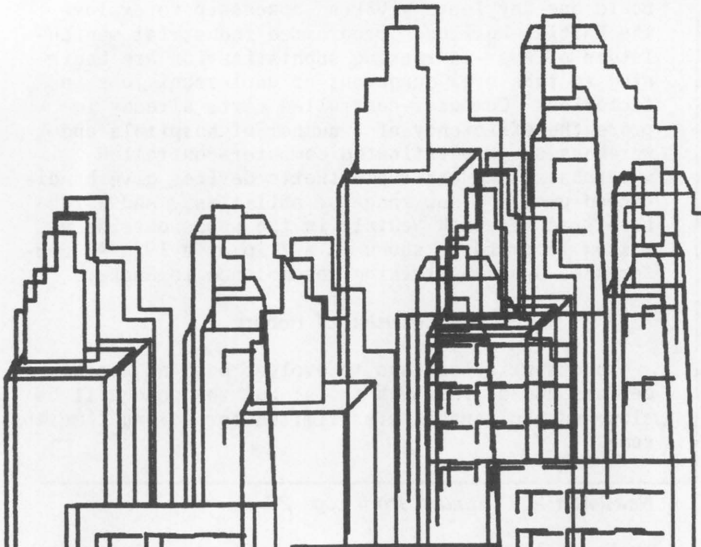
SUPPLEMENT,
October, 1976

Grace C. Hertlein,
Art Editor

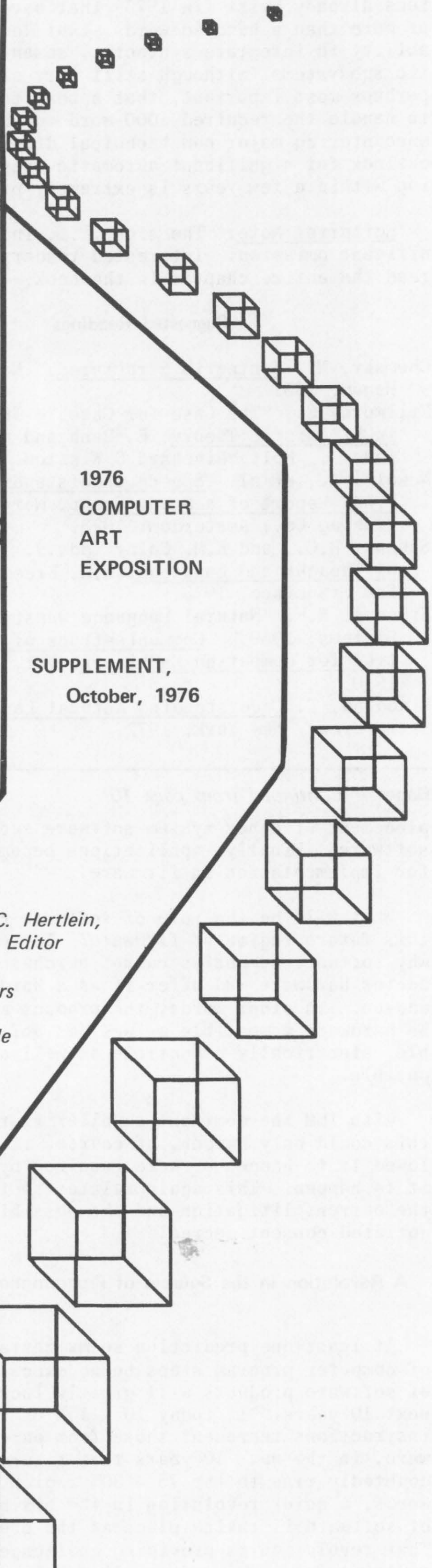
Computers
and
People



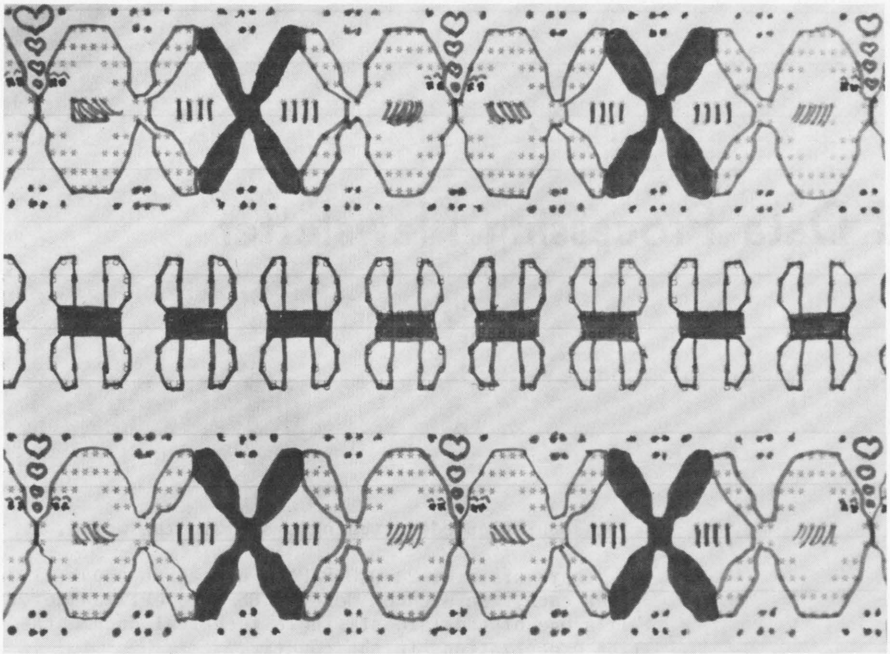
Tom Odom – U.S.A.: "My works are all based on Bowditch Curves. Computer system used was an IBM 1130 with an on–line CalComp plotter. Programming is in FORTRAN."



Terri Lynne Hale – U.S.A.: *Random City*, an exercise in using a canned RANDOM routine.



Howard Fegarsky – U.S.A.: *Diminishing Box*, from the Square Series.



Fabric Design by High School Students

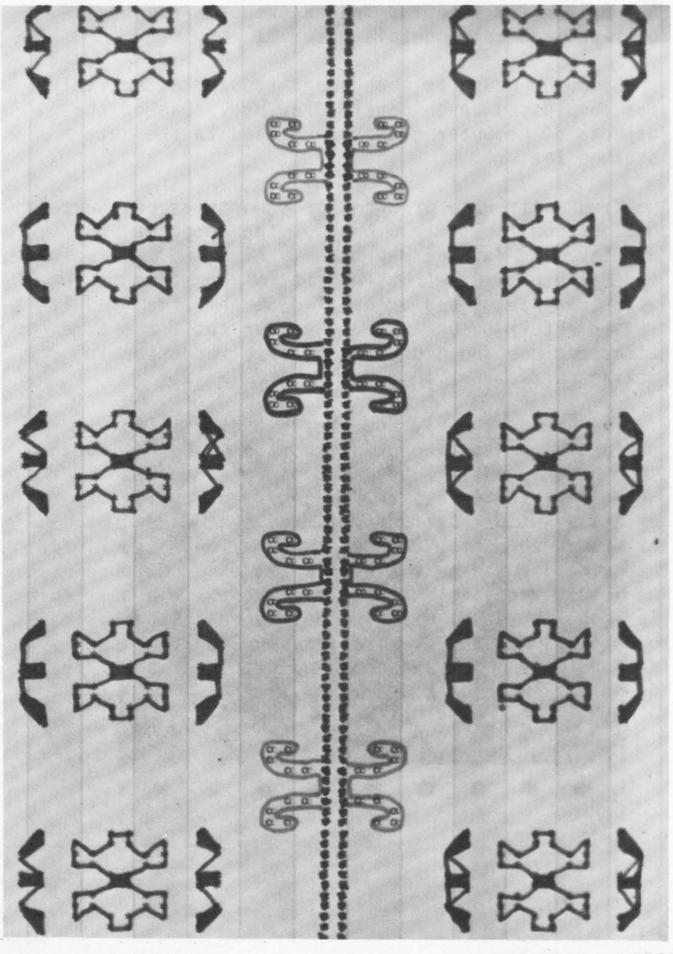
"By applying character string manipulations to a real design, many students, especially the non--math oriented, are turned on by witnessing that suddenly an ordinary line printer is transformed into a live 'fabric machine', producing yards of designs for textiles.

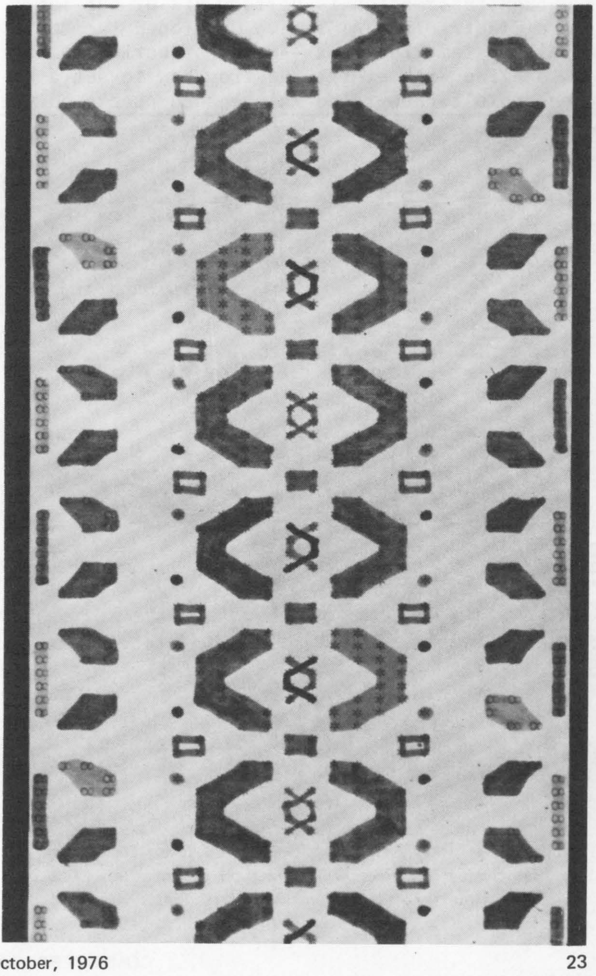
We found that the fusion of the two topics, computer graphics and fabric design, is a creative learning vehicle for our students."

(Comments by Dr. Kai Chu, Mt. Diablo High School, Concord, California.)

Sharon Lauthner – U.S.A.: Above, and below left – two untitled designs for textiles by a student from Mt. Diablo High School, Concord, California. The instructor was Dr. Kai Chu.

Bitsy Behrendt – U.S.A.: Below, a design for textile, also from Mt. Diablo High School.





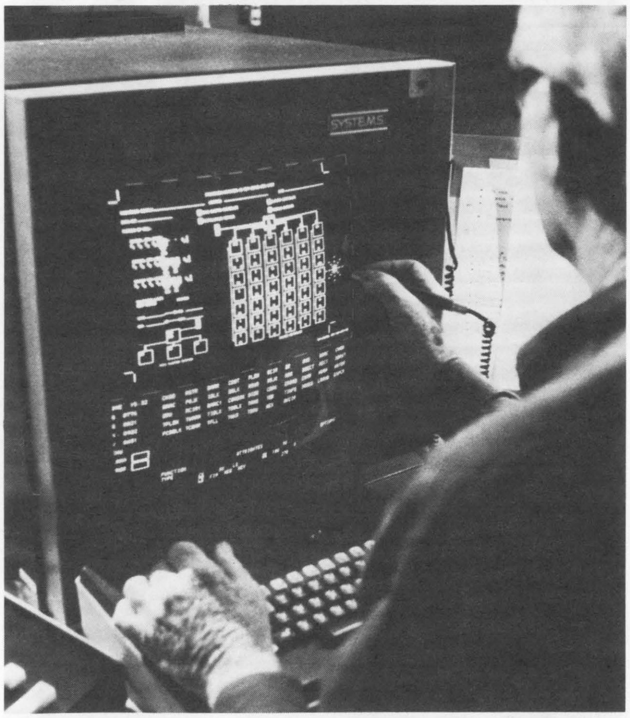
Computing and Data Processing Newsletter

COMPUTER AIDS IN DOCUMENTATION OF TELEPHONE TECHNOLOGY

Peg Murray
Western Electric
195 Broadway
New York, NY 10007

About 25 years ago, it was enough to be "experienced" to be a crackerjack telephone craftsman. But with the advent of electronic switching and other sweeping advances in communication, experience alone is not enough to keep up with rapidly developing telephone technology.

So to make sure that every telephone craftsman has enough information to perform his or her job, Western Electric and AT&T have developed the BSP's — the Bell System Practices — over a quarter million pages dealing with everything from how to set a telephone pole to how to do maintenance checks on the most sophisticated switching system in the world.



This operator is using a light-sensitive pen to revise and update BSP's (Bell System Practices). This method of updating results in a six-to-one time reduction for making corrections.

"There are over 24,000 BSP's, averaging about 11 pages each," said Chuck Deleot, BSP project manager. "It's one of the largest single technical documentation organizations in the country."

In addition to the instructions designed for telephone companies are technical manuals to support the various radar, sonar and other electronic systems manufactured for the nation's defense agencies.

A staff of technical illustrators provides isometric or line diagrammatic illustrations to accompany the written portion of the manual, while a computer system provides typesetting, composing, and automation for transforming an author's rough manuscript into reproducible material.

Computer graphics technology is used extensively for completing rough pencil drawings. One of the aspects of computer graphics as used is the data storage feature.

Frequently used drawings or schematics can be recalled from the storage bank of the computer and displayed on cathode ray tubes. With the magic of the computer and a light-sensitive pen, these schematics can be revised or updated at any time, resulting in a four-to-one time reduction for first time illustrations and a six-to-one time reduction in making corrections.

"We are constantly appraising and revising our philosophy behind these instructions," Deleot added, "and have recently introduced a new concept for BSP's. We call it TOP for Task Oriented Practices. TOP offers a set of instructions that can be used by all craftsmen no matter what their experience level. In other words, the person with a lot of experience need only read the basic instructions for a particular job, whereas the less experienced fellow, using a carefully designed cross-reference system, can find as many details associated with the job as he needs to complete it."

COMPUTER HELPS IDAHO OFFICIALS PROVIDE QUICK FINANCIAL ASSISTANCE TO FLOOD VICTIMS

Dick Sybert
Idaho Department of Employment
Box 35
Boise, ID 83707

Within days of the disastrous Teton Dam flood that killed 11 people and caused more than $1 billion in damage, Idaho state officials used a computer to process disaster unemployment assistance applications and start speeding checks to more than 3,400 flood victims. The first checks were mailed just three days after applicants certified their eligibility for payments to the Idaho Department of Employment.

Department officials also used their computer to recreate the entire regular unemployment insurance files of their local office in Rexburg, which was destroyed by the flood waters. The computer regenerated the records in one day, enabling the state to continue its unemployment insurance program without missing or delaying any payments.

"The Teton Dam flood was a major catastrophe that was impossible for us to prepare for," said Jim Scott, chief of data processing for the Department of Employment. "We had to react to the situation after it occurred. The computer enabled us to respond quickly and assist the flood victims."

As soon as word of the dam collapse reached the department, it established emergency reporting centers to take applications for the disaster relief funds. The program is designed to provide temporary financial assistance to disaster victims.

Despite the emergency demands made on the computer during the flood, it continued to process its normal workload for the rest of the state.

The Idaho Department of Employment uses an IBM System/370 Model 145 computer.

PROGRAMMABLE CALCULATOR SCORES WOMEN'S AIR COMPETITION

Michael Malone
Hewlett Packard
1505 Page Mill Road
Palo Alto, CA 94304

Since 1947, many of America's finest women pilots have competed in the Powder Puff Derby, the world's longest speed race for light aircraft. This year, the 29th race, all scoring of contestants was conducted by a desktop, programmable calculator.

The "Powder Puff Derby" (a nickname given to early all-women airplane races that is attributed to Will Rogers) started at Sacramento, California, Executive Airport. The finish came three days later, 2,926 miles away, at the Greater Wilmington airport in Wilmington, Delaware. Two hundred aircraft competed, representing 38 states, the Bahamas, Canada and South Africa. Twenty of the racers flew solo. Nearly half were first-time participants.

Officially titled the All Woman Transcontinental Air Race, the Powder Puff Derby is a series of ten hops averaging 300 miles each. According to Douglas DuPerow, statistician for the race, and the only male involved in the event, only stock-model planes of 145-150 horsepower are eligible to enter. To give all entries an equal chance, each plane is handicapped by make and model, and flies only against that handicap.

DuPerow, using the programmed calculator, computed scores for each of the 200 contestants. During the race, he was stationed in Wilmington, entering subtotal times as they were telephoned in by timers located at the terminus of each hop.

At the end of the day, unofficial standings were announced. If a racer had a question about her results, the terminus scorers were equipped with pocket programmable calculators to verify the results.

When all the planes had landed in Wilmington, and the times for the final leg were received and entered into the calculator system, it used a special program designed by DuPerow to correlate all timing data, compare the results with handicaps -- and, almost instantly, select the winner.

The calculators used to score the Powder Puff Derby were the HP 9830 and the HP-25.

MINICOMPUTER TACKLES SOCIETY PROBLEMS WITH MODELS AND SIMULATIONS

Diane Wilson
University of Notre Dame
Notre Dame, IN 46556

How long will the world's fuel supply last? Even students at the smallest college could tackle such a large societal problem, thanks to an interdisciplinary and intercollegiate project being conducted by the University of Notre Dame.

Funded by a grant from the Max C. Fleischmann Foundation, Dr. William I. Davisson, professor of economics, and Dr. John J. Uhran, Jr., associate professor of electrical engineering, are developing a computer program that will allow colleges and other institutions with minicomputers to take advantage of the latest teaching methods in modeling and simulation.

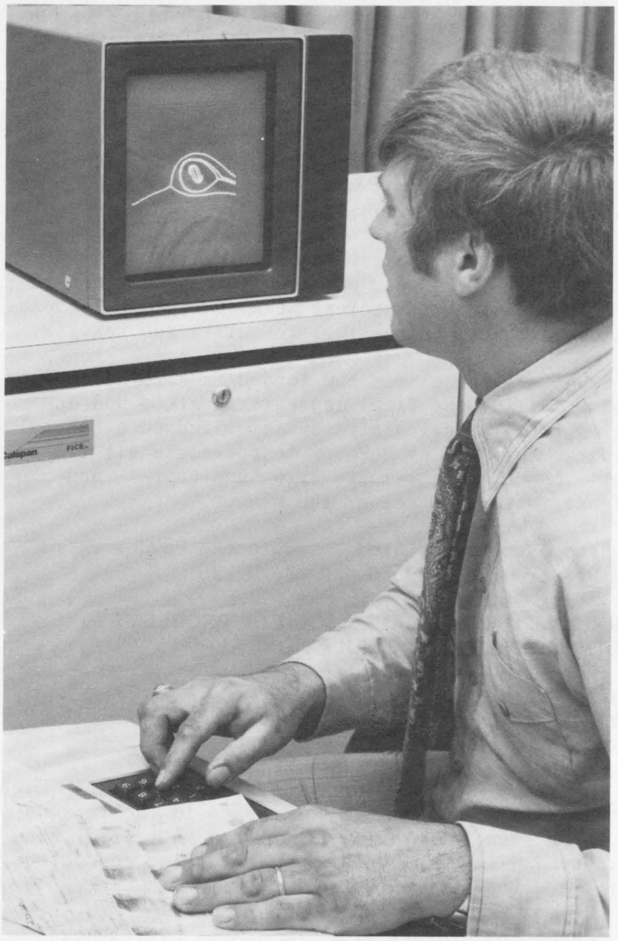
The directors are using the systems dynamics approach to modeling, one of the simplest ways to simulate real life situations on a computer. According to Davisson and Uhran, this technique is one of the few available which allow one not only to model rates of change, but also to determine what causes change over time. Without the ability to conduct such studies on their own small computers, many institutions will find themselves mere spectators to scientific controversy, they said.

An essential aspect of the project will be careful field testing of the translator program at a number of small colleges using different computing systems. These institutions will serve as initial field test centers for implementation of the system and provide student feedback concerning the program's reliability. When the project is completed, the directors hope to make a complete kit available to institutions throughout the United States.

COMPUTER-BASED SYSTEM IDENTIFIES AND MATCHES FINGERPRINTS

Ralph Wallenhorst
Calspan Technology Products, Inc.
Buffalo, NY 44221

The first fully automatic system for classifying fingerprints has been demonstrated by Calspan Tech-



An engineer at Calspan Technology Products operates the first fully-automatic system for classifying fingerprints. The fingerprint initially is described in digital form by a companion unit, an automatic fingerprint reader. Using this data, the classifier locates the cores, deltas, and "type lines" – shown on the monitor – which define the fingerprint's pattern. Then the unit assigns the print to one of the 116 classifications established by the National Crime Information Center of the FBI.

nology Products of Buffalo, N.Y. The system is for use in large identification bureaus as a replacement for manpower-intensive manual classification.

"This computer-based system is more consistent than visual classification and will greatly decrease workloads," asserts Claron W. Swonger, company vice-president. Importantly for identification bureaus, the system directly implements the same classification rules which have been proven and used for decades by fingerprint experts.

Prior to classification by the new equipment, the print must be read by the FINDER Automatic Fingerprint Reader. This system automatically converts the print into digital form and locates the tiny minutiae — endings and branchings of ridge lines — which will later be used to distinguish one fingerprint from another. FINDER also produces data on ridge-valley direction and enhanced, digitized, fingerprint images.

The data is then transferred to the company's new FICS, short for the Fingerprint Classifier System. The system, using FINDER data, locates the cores, deltas, and "type lines" which define the fingerprint's pattern. These pattern types include right and left loops, various types of whorls, arches, tented arches, and accidentals. The FICS system also counts ridges between key points in the fingerprint, as does a human fingerprint classifier. By this method, the classifier assigns each fingerprint, by means of a standardized letter code, to one of the 116 classifications established by the National Crime Information Center of the FBI. If the classification is a borderline case, alternate classes are indicated.

An overall numerical classification for the fingerprint card is then made by designating the particular combination of classifications for the ten individual prints on the card. Mr. Swonger explains that there are hundreds of thousands of such combinations in actual practice, each corresponding to a bin in which the identification bureau files all its cards with the same overall numerical classification.

To determine whether an individual is already in the fingerprint file, his card need be searched only against those cards already in the appropriate bin or few bins indicated by the classifier system. This can be done manually or automatically by utilizing special searching and matching equipment currently available for matching new prints at high speed against sets already in the files.

The standard automatic matching equipment for use in large city and state identification bureaus is somewhat different in technology from the prototype matcher being built by Calspan for the FBI. Because of the large number of fingerprint cards in the FBI's arrest file — more than 21 million — the FBI matcher will be required to search for a match at a rate of 800 sets of filed fingerprints per second.

COMPUTERIZED RESEARCH SYSTEM HELPS LAWYERS PROVIDE BETTER LEGAL SERVICES

Amos Landman
Ruder & Finn Incorporated
110 East 59th Street
New York, NY 10022

Before long, a lawyer in Tuscon, Arizona, will be able to query a computer in Dayton, Ohio, and research one of the finer points of securities law in Tokyo, Japan.

This was the prediction of Jerome S. Rubin, president, Mead Data Central, Inc., New York City, before the International Bar Association in Stockholm, Sweden.

Mr. Rubin's company operates the LEXIS computer-assisted legal research service. The service now has U.S. federal cases, the case law of ten states, securities tax and trade regulation law, and many other materials — the full text of 900,000 items so far — stored in the computer's memory. Mr. Rubin predicted that the service will ultimately have libraries of international law and of the law of other nations.

Nearly 17,000 American lawyers, judges and accountants, as well as some 6,000 law students, have been trained in the use of LEXIS. Subscribers to the legal service include various state and federal courts and agencies, some of the best-known law firms in the U.S., all of the "Big 8" accounting firms, bar associations and law schools.

By using the service, thousands of lawyers can perform legal research more thoroughly and quickly. In many cases, lawyers can do legal research that cannot be done with traditional methods. For ex-

(please turn to page 21)

GAMES AND PUZZLES for Nimble Minds – and Computers

Neil Macdonald
Assistant Editor

It is fun to use one's mind, and it is fun to use the artificial mind of a computer. We publish here a variety of puzzles and problems, related in one way or another to computer game playing and computer puzzle solving, or to the programming of a computer to understand and use free and unconstrained natural language.

We hope these puzzles will entertain and challenge the readers of *Computers and People.*

NAYMANDIJ

In this kind of puzzle an array of random or pseudorandom digits ("produced by Nature") has been subjected to a "definite systematic operation" ("chosen by Nature") and the problem ("which Man is faced with") is to figure out what was Nature's operation.

A "definite systematic operation" meets the following requirements: the operation must be performed on all the digits of a definite class which can be designated; the result displays some kind of evident, systematic, rational order and completely removes some kind of randomness; the operation must be expressible in not more than four English words. (But Man can use more words to express it and still win.)

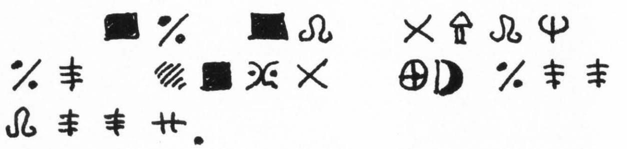
NAYMANDIJ 7610

```
0 7 0 9 6 4 5 0 3 9 2 7 9 5 8 7 7 4 4 4
9 7 5 0 8 6 4 4 3 2 5 5 5 8 2 8 8 6 8 7
4 2 4 2 4 8 0 7 8 5 4 3 8 5 2 1 1 9 1 9
1 3 1 4 6 0 6 8 5 5 4 7 3 4 5 5 9 2 2 4
5 7 8 8 9 0 5 8 5 7 2 0 0 4 5 8 5 4 9 8
9 7 4 1 1 6 9 3 5 0 3 1 8 0 5 4 3 4 2 3
8 8 2 6 3 6 4 3 9 5 4 2 1 5 8 1 2 6 3 9
7 7 7 3 5 6 0 4 5 6 5 5 5 7 8 4 8 4 9 3
2 6 9 3 7 9 3 6 5 3 9 4 6 5 8 4 2 9 7 2
9 8 8 0 1 4 0 6 2 1 4 9 1 5 4 9 0 8 9 6
```

MAXIMDIJ

In this kind of puzzle, a maxim (common saying, proverb, some good advice, etc.) using 14 or fewer different letters is enciphered (using a simple substitution cipher) into the 10 decimal digits or equivalent signs for them. To compress any extra letters into the 10 digits, the encipherer may use puns, minor misspellings, equivalents like CS or KS for X or vice versa, etc. But the spaces between words are kept.

MAXIMDIJ 7610



NUMBLES

A "numble" is an arithmetical problem in which: digits have been replaced by capital letters; and there are two messages, one which can be read right away and a second one in the digit cipher. The problem is to solve for the digits. Each capital letter in the arithmetical problem stands for just one digit 0 to 9. A digit may be represented by more than one letter. The second message, which is expressed in numerical digits, is to be translated (using the same key) into letters so that it may be read; but the spelling uses puns, or deliberate (but evident) misspellings, or is otherwise irregular, to discourage cryptanalytic methods of deciphering.

NUMBLE 7610

```
            T R U E
          x L O V E
          ---------
          S V L S V
        O O R G E
      E O T I G
    V E I U U
    ---------------
=   V T E V L L L V
```

23464 10751 2583

We invite our readers to send us solutions. Usually the (or "a") solution is published in the next issue.

SOLUTIONS

NAYMANDIJ 769: Sequence column 3.

MAXIMDIJ 769: The second half of anything goes faster than the first half.

NUMBLE 769: Safety breeds danger.

Our thanks to the following individuals for sending us solutions: Byung Sun Choung, San Diego, CA: Numble 768 – Frank DeLeo, Brooklyn, NY: Naymandij 768; Maximdij 768; Numble 768 – T.P. Finn, Indianapolis, IN: Maximdij 768; Numble 768 – Abraham Schwartz, Jamaica, NY: Numble 768 – Scott Troutman, Winston–Salem, NC: Naymandij 768; Maximdij 768.

COMPUTER GRAPHICS AND ART

COMPUTER GRAPHICS and ART is a new international quarterly of interdisciplinary graphics for graphics people and computer artists. This new periodical is aimed at students, teachers, people from undergraduate and graduate institutions, researchers, and individuals working professionally in graphics. Its topical coverage is broad, embracing a variety of fields. It is useful, informative, entertaining, and current.

Here Are Some Samples:

Learning Through Graphics
by Dr. Al Bork, University of California, Irvine, California
A ten-year forecast for computers, education, and graphics by a leading authority.

Art of the Technical World
by Dr. Herbert Franke, Munich, Germany
Computer art as the bridge between the two realms of art and leisure.

Expanding the Graphics Compatability System to Three Dimensions
by Richard F. Puk, Purdue University, Lafayette, Indiana
Design considerations for a user-oriented 3-D graphics system.

A Personal Philosophy of Ideas, New Hardware, and the Results
by Duane Palyka, University of Utah, Salt Lake City, Utah
The frame-buffer from Evans and Sutherland allows the artist to treat the computer as a paint and brush medium.

How to Build Fuzzy Visual Symbols
by Alex Makarovitsch, Honeywell Bull, Paris, France
A new approach to computer art and graphics by a computer scientist.

The State of the Art of Computer Art
by Grace C. Hertlein, Editor
Comparisons of early computer art and today's newer art. What is art? What is art in computer art?

Inexpensive Graphics from a Storage Cathode Ray Tube
by Charles J. Fritchie and Robert H. Morriss, Tulane University, New Orleans, Louisiana
Illustrations and photographic techniques used to achieve graphics from a storage tube CRT.

An Investigation of Criteria for Evaluating Computer Art
by Thomas E. Linehan, Ohio State University, Columbus, Ohio
The new aesthetic of computer art requires a departure from the previous, formalist-traditionalist doctrines for evaluating art.

Send your manuscripts, papers, art, and ideas to:

Editor, *COMPUTER GRAPHICS and ART*
Berkeley Enterprises, Inc. – Chico Branch
555 Vallombrosa – No. 35
Chico, California 95926

List of Coverage for Up-Coming Issues

- Applied Arts and Graphics
- Architectural Graphics
- Cartography Systems
- Computer-Aided Design
- Computer Assisted and Managed Instruction Utilizing Computer Graphics
- Computer Graphics in Physics, Chemistry, Mathematics, etc.
- Computer Programs for New Applications
- Display Systems and Graphics
- Fine Art and Media Explorations
- Graphics in Business
- Hardware Systems and Graphics
- Interactive Graphics Languages and Systems
- Languages for Computer Graphics and Graphics Primitives
- Software Systems and Graphic Requirements
- Statistical Packages and General Graphing
- Syllabi for Computer Graphic Courses

– –(may be copied on any piece of paper)– –

To: *COMPUTER GRAPHICS and ART*
Berkeley Enterprises, Inc. – Business Office
815 Washington St.
Newtonville, Mass. 02160

( ) Please enter my PERSONAL (U.S. and Canada) subscription to CG&A ($10 per year).
( ) Please enter my FOREIGN PERSONAL subscription to CG&A ($13 per year).
( ) Please enter my LIBRARY/DEPARTMENTAL subscription to CG&A ($15 per year).
( ) Enclosed is my PERSONAL CHECK AND/OR PURCHASE ORDER for CG&A.
( ) Enclosed is $2.50 for a sample copy of CG&A (applicable toward a subscription).

FULL REFUND IN 30 DAYS IF NOT SATISFACTORY

Name______________________________
Title______________________________
Organization______________________________
Address______________________________

HERE IS YOUR OPPORTUNITY FOR FEEDBACK TO US:

( ) I hope to submit for publication in CG&A material on the following topics:______________________________

( ) I am interested in reading materials by the following authors:______________________________

( ) I am particularly interested in coverage of the following subjects:______________________________

( ) I would like to receive materials on other Berkeley Enterprises, Inc. publications: ( ) *COMPUTERS and PEOPLE* ( ) *The COMPUTER DIRECTORY and BUYER'S GUIDE* ( ) *People and the PURSUIT of Truth* ( ) *The Notebook on COMMON SENSE and WISDOM* ( ) *WHO'S WHO in COMPUTERS and DATA PROCESSING* ( ) Books

( ) I am interested in: ( ) black and white computer art reprints at low cost (a bonus for subscribing to CG&A) ( ) 77 page FORTRAN IV art manual ( ) 45 page interdisciplinary graphics bibliography by G. Hertlein

( ) Additional Comments (attach another paper if needed):______________________________