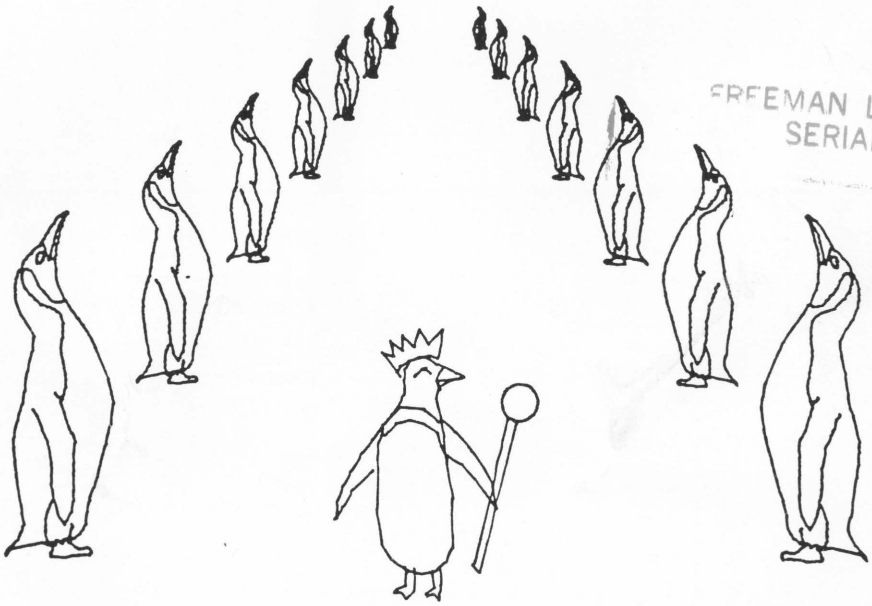
computers and people

formerly *Computers and Automation*

Sept.—Oct., 1986

Vol. 35, Nos. 9-10



PENGUINS, JUST FOR FUN

The Computer Almanac and Computer Book of Lists — Instalment 49

Neil Macdonald, Assistant Editor

12 RULES FOR DICTATING TO MACHINES (List 860901)

- Organize your thoughts before starting.
- Decide the purpose of your communication.
- Know what you are going to say.
- Begin by identifying yourself.
- Tell the kind of document and other needed details.
- Talk in a normal way and enunciate clearly.
- Be concise, not conversational, not verbose.
- Spell unusual or difficult words to avoid mistakes.
- Specify capitalization, punctuation, paragraph breaks, etc.
- Avoid background noises.
- Tell where the document when typed is to go.
- Keep always in mind possible confusions like "in view of" for "in lieu of".

(Source: from Russell K. Paul, Public Relations, Lanier Business Products, Inc., 1700 Chantilly Drive, Atlanta, GA 30324)

12 APHORISMS ABOUT THE ART OF THINKING (List 860902)

- Observing a person thinking well may lead to admiration, and then desire to do likewise.
- Everybody is conscious of spells during which his mind is thinking at its best.
- Children under nine or ten are poets and philosophers. One child said "Father, what is beauty? What makes it?" This is thinking.
- Ill-treated inspiration does not dare to return.
- We have a natural belief that an art of thinking exists.
- From "I can't think well today," it does not follow that I can't think well tomorrow.
- The chief obstacle to thinking is not stupidity but the phantasm "I cannot think."
- Social conversation with its usual polite requirements and indulgences produces insincerity that hinders thinking.
- Two notions side by side in the mind always hinder its working.
- It is exciting to hunt after thoughts or facts intended to throw light on an important question.
- Timid, easily abashed, people have a fatal capacity for letting in extraneous thoughts, mental parasites.
- The multitude has an antipathy against thinking well.

(Source: from Chapters 4 and 5 in "The Art of Thinking", by Ernest Dimnet, published by Simon and Schuster, Inc., New York, NY, 1928, 202 pp; slightly edited)

THE 4 PAPERS PRESENTED IN VOL. 1, NO.1, OF "THE INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS" (List 860903)

Constructs and Phenomena Common to the Semantically-Rich Domains / Beth Adelson, Harvard Univ., Cambridge, MA

An Intelligent Computer Vision System / Sushing Chen, Purdue Univ., Indianapolis, IN

Hierarchical Representation of Problem-Solving Knowledge in a Frame-Based Process Planning System / Dana S. Nau, Univ. of Maryland, College Park, MD, and Tien-Chien Chang, Purdue Univ., West Lafayette, IN

Toward General Theory of Reasoning with Uncertainty. I: Nonspecificity and Fuzziness / Ronald R. Yager, Machine Intelligence Inst., Iona College, New Rochelle, NY

(Source: Vol. 1, No. 1, "The International Journal of Intelligent Systems," copyright 1986 by and published by John Wiley and Sons, 605 Third Ave., New York, NY 10158)

Ω

Computing and Data Processing Newsletter

INSTRUCTION IN DICTATING TO MACHINES BEING LOST FROM VIEW

Russell K. Paul
Lanier Business Products, Div. of Harris Corp.
1700 Chantilly Dr. NE
Atlanta, GA 30324

Will speech recognition machines and voice-based input to computers stall in the marketplace? because the skills learned over 60 years of use of modern equipment for dictating to human transcribers have been neglected in the sparkling distractions of the powers of computers?

Many people have a strong reluctance or discomfort in talking to telephone answering devices. They simply refuse to leave messages on them.

Our experience in voice technology has taught us that people must be trained in verbal communication skills in order to be successful in using voice technology in the modern office.

The common sense rules are not difficult, but using them requires instruction, practice, and mastery. See List 860901 on page 2.

SCIENTISTS AT MIT CREATE THREE-DIMENSIONAL HOLOGRAM DIRECTLY FROM COMPUTER

News Office
Mass. Inst. of Technology
Cambridge, MA 02139

Scientists at the Massachusetts Institute of Technology's Media Laboratory demonstrated recently the first free-standing holographic image generated directly from a computer. The technique is expected to have widespread applications in five to ten years in such fields as design, medicine and architecture. The holographic stereogram, the product of lasers and optical systems, makes it possible to view a three-dimensional image as a suspended object floating in space in front of the observer, with nothing between the image and the viewer.

The MIT scientists used an automobile and a pelvic bone in their first demonstrations of the system, but many kinds of computer data can be "beamed up" into a three-dimensional display, according to Dr. Stephen A. Benton, the principal investigator. "At present, you have a two-dimensional image on a computer screen and you can rotate it to give a three-dimensional perspective, but you can never really see it as a solid," he said. "But now, using our system, the image is completely projected into space, suspended, floating in front of the observer. You get a real sense of what it's going to look like, and thus you have a much more effective interface with the computer."

Using the technology, Dr. Benton said, those designing products on a computer will be able to see what they will look like. In the case of cars, he said, computer-generated holograms can eliminate the need for carving clay models of proposed designs. As other examples, Dr. Benton said, architects should be able one day to show buildings in three-dimension, rather than on the drawing-board, and surgeons will be able to examine images of body parts before operating. He said the MIT researchers had approached the problem in two parallel steps. "We came up with a new way of making holograms to produce this kind of suspended image from computer data, and this has been paralleled by research in computer graphics, in which the image is processed in such a way as to give a realistic, undistorted three-dimensional image."

One of the experimental projections shows a three-dimensional, solid-looking image of an automobile nine inches long and four inches high floating in space in an alcove-like setting about three feet across, a foot high and several feet deep. Although the viewer

(please turn to page 12)

Vol. 35, Nos. 9-10
September-October, 1986

"Computers and People" (ISSN 0361-1442), formerly "Computers and Automation," is published every two months at 815 Washington St., Newtonville, MA 02160 U.S.A., by Berkeley Enterprises, Inc. Printed in U.S.A. Second-class postage paid at Boston, MA and additional mailing points.

Subscription rates, effective Sept. 1, 1986: U.S.A., $22.00 for one year, $43.00 for two years; elsewhere, add $7.00 per year.

NOTE: The above rates do not include our publication, the "Computer Directory and Buyers' Guide." To receive this, please add $20.00 per year to your subscription rate in the U.S.A., and $23.00 per year elsewhere.

NOTE: No organization in Switzerland or Morocco is authorized or permitted by us to solicit for, or receive payment for, "Computers and People" or the "Computer Directory and Buyers' Guide." All subscriptions to and payments for these publications should be sent directly to Berkeley Enterprises, Inc.

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

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

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 four weeks for the change to be made.

computers and people

formerly *Computers and Automation*

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.

Computer Field → Zero

There will be zero computer field and zero people if the nuclear holocaust and nuclear winter occur. Every city in the United States and the Soviet Union is a multiply computerized target. Radiation, firestorms, soot, darkness, freezing, starvation, megadeaths, lie ahead.

Thought, discussion, and action to prevent this earth-transforming disaster is imperative. Learning to live together is the biggest variable for a computer field future.

Front Cover Picture

The front cover shows a sample of art by Diana Wong, a student in a computer-assisted art class at California State University - Chico, CA. This illustration was produced making use of a group of programming routines, on an HP3000 minicomputer connected to a Tektronix display terminal.

Back Copies

For issues 1951 through 1979, only microfilm copies of *Computers and People* (formerly *Computers and Automation*) are available from University Microfilms International, 300 North Zeeb Rd., Ann Arbor, MI 48106. For issues 1980 to the present, only back copies are available from Berkeley Enterprises, Inc., 815 Washington St., Newtonville, MA 02160. The cost of these back copies is $5.00 each, plus postage and handling. Since we are not a technical journal, we do not supply free reprints of articles.

Signals in Table of Contents

[A]	–	Article
[C]	–	Monthly Column
[E]	–	Editorial
[EN]	–	Editorial Note
[O]	–	Opinion
[FC]	–	Front Cover
[N]	–	Newsletter
[R]	–	Reference

Type of Subscription

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

Editorial

The Time to Prevent Mischief Is Before It Happens

Edmund C. Berkeley, Editor

Almost everybody knows well this principle. But what do we do in the real world? We experiment with mischief, against our better judgement.

The organizations which can supply mischief produce it, and sell it to us. Then the complex of the stockholders, managers, labor, communities, and "infrastructure" push the supply of mischief. They increase the supply and market it. If it is clear that mischief is no longer a good marketable commodity in one country, the complex organizes to promote their wares in another country. They become a vested interest. And so on.

The mischief which may well be the most important, perplexing, and critical in this century (and probably for more centuries to come) is the visions of armaments, "defense", war, and especially nuclear war. Computer people and computer engineers substantially contribute to this mischief because their work improving computer applications worsens the mischievous game. But there is more to the trouble than that: the dreams, the mirages, the mystique of power-hungry or irrational leaders. Adolf Hitler, the Nazi dictator 1933-1945, was a prime example.

Following is some of the most effective summarizing of the central problem of armaments that I have seen. This was written before World War II and before nuclear weapons. It is from pages 271-272 in "Merchants of Death: A Study of the International Armaments Industry" by H.C. Englebrecht and F.C. Hanighen, published by Dodd, Mead, and Co., New York, in 1934. But in many ways it sounds as if it were written yesterday:

[Beginning of Quotation]

There remains then but one real way out, disarmament. The various futile conferences on disarmament have not been in vain if they have opened the eyes of the peace forces to the real problem which confronts them. Disarmament has not been achieved because of the international political situation. International politics in turn are determined by our whole civilization. Our civilization has permitted and even fostered war-making forces, such as nationalism and chauvinism, economic rivalry and competitive capitalism, imperialism and colonialism, political and territorial disputes, race hatred and pressure of population. The traditional way of establishing an equilibrium between these rival forces has been and is violence, armed warfare.

Disarmament is thus a problem of our civilization. It will never be achieved unless these war-making forces are crushed or eliminated. The problem of disarmament is therefore the problem of building a new civilization. All attempts at dealing with disarmament by itself, without consideration of the deeper issues involved, are doomed to failure. Minor agreements may be reached, limited to a short period of time, but the world will never cease being an armed camp until the basic elements of our present civilization have been changed.

The same holds true of the armament industry. A world which recognizes and expects war cannot get along without an enterprising, progressive, and up-to-date arms industry. All attempts to attack the problem of the arms makers in isolation -- by nationalization or by international control -- are almost certain to fail.

The arms industry is plainly a perfectly natural product of our present civilization. More than that, it is an essential element in the chaos and anarchy which characterize our international politics. To eliminate it requires the creation of a world which can get along without war by settling its differences and disputes by peaceful means. And that involves remaking our entire civilization.

Meanwhile those interested in creating a war-less world need not be idle and await the dawn of a new day. They can support every move made for the peaceful settlement

(please turn to page 26)

Artificial Intelligence Comes Of Age

Larry R. Harris, Ph.D.
President
Artificial Intelligence Corp.
and
Dwight B. Davis
Senior Editor
"High Technology" magazine

c/o Bantam Books
666 5th Ave.
New York, NY 10103

"When a computer draws conclusions that are beyond the grasp of all but a few experts in the field – tasks within the reach of today's machines – the lines between human intelligence and computer knowledge processing begin to blur."

Editorial Note: This article is based on Chapter 1, *AI Comes of Age*, of the book *Artificial Intelligence Enters the Marketplace* by the two authors stated above, copyright 1986 by them. The book is published by Bantam Books, 666 Fifth Ave., New York, NY 10103, 194 pp., and is currently available, important, and interesting. – *ECB*

The Attempt to Imbue Computers with Specific Human Traits

Many people approach computers with a mixture of distrust, skepticism, and even fear -- and for good reason. The typical computer is idiosyncratic and hard to use, knowledge about one machine is often useless when applied to another, and what initially seemed to be a very smart contraption sometimes seems more like a very dumb annoyance. Still, most people who work with computers eventually overcome their initial misgivings and, after a fair amount of effort, come to appreciate electronic processors as powerful and useful tools.

Now, a fascinating computer science discipline called artificial intelligence promises to topple many of the barriers that exist between people and computers. Artificial intelligence (AI) is just beginning to appear in commercial products after almost 30 years of research and development in university and corporate labs. The discipline encompasses a broad spectrum of fields and little agreement exists as to what, exactly, "artificial intelligence" means. But one common thread unites the developers of this technology: They are attempting to imbue computers with traits long regarded as specific, and unique, to human beings. The targeted traits include understanding and speaking natural languages, such as English, the ability to give expert advice about fields as diverse as medicine and geology and, generally, to exhibit behavior that we normally associate with human thought and intelligence.

Intelligent computers? Is nothing sacrosanct? Aren't computer scientists aware of all the bleak scenarios that have been drawn by science fiction writers and film makers about future computers running amok? Are some people so unable to separate fact from fancy that they actually believe that an electronic machine could ever match the intricate and mysterious workings of the human mind?

Such questions are common, but not everyone views artificial intelligence with trepidation or skepticism. As in the field of genetic engineering, potential for abuse exists in the pursuit of AI. But, just as genetic engineers believe the potential benefits of their research outweigh most of the risks, those developing artificial intelligence believe their work holds much promise. The public perception of this work, however, is often laden with misconceptions about the field and miscalculations about its likely role in our world.

No doubt the most common source of AI awareness is science fiction. For decades, writers have created future worlds populated by intelligent computers and robots; the resulting roles that humans play are usually less-than-happy ones. With assembly line robots already replacing people in some manufacturing plants, such fantasy worlds seem, at times, uncomfortably close to reality. Yet, deep down, people tend to draw a line between the proven ability of robots to perform rote, manual labor, and the potential for computers to one day handle tasks that require "true" intelligence. Yet, despite what most believe, that line is not inviolable.

The Meaning of Intelligence

Computers already routinely perform tasks that would have been labeled intelligent several years ago. However, "intelligence" is a poorly understood concept. Few contemplate what it means, and those who do rarely agree on a definition. Because so many definitions of intelligence exist, it's possible to keep demanding increasingly stringent requirements for the demonstration of machine thought, requirements that always stay one step ahead of what state-of-the-art computers can accomplish.

Computer Programs that Play World-Class Chess

For example, creating a computer that could play world-class chess was once a Holy Grail for those seeking to develop intelligent machines. Today a number of chess-playing computer programs demonstrate a mastery of the game that falls just short of the Grand Master level, but this capability is somehow no longer a definitive demonstration of machine thought. Because chess-playing computers use well-understood, human-programmed techniques to achieve their success, and because these techniques differ from the methods human players employ, few believe that such computers really "think."

Ironically, the claim that machines can exhibit intelligence has precipitated an unprecedented effort to understand the workings of the human mind. The field of artificial intelligence is unique in this regard because it provides proofs as well as theories. AI is hardly the first or the foremost discipline to examine questions about human intelligence and behavior. Psychology, philosophy, and linguistics are among the long-standing players in this game. But by their nature, these fields consist almost solely of developing theories about the workings of the human mind. And, despite the fact that there has been no adequate way to test the accuracy of these theories, many have become accepted "facts" within their respective realms.

Theories of Language that Did Not Hold Up

The early AI researchers, too, believed in the accuracy of many of these theories. These researchers were charged with implementing the theories on machines in order to imitate human behavior, but things didn't go as smoothly as expected. For example, when computer scientists first started to program machines to understand English, the scientists thought the task before them was relatively straightforward. After all, they had in hand theories of language and grammar that had been developed and refined over hundreds of years. All that needed to be done was to program these theories and rules into the computer and Voilá! A machine that understands English. But when the scientists programmed the accepted rules and theories they failed. Many theories simply didn't hold up when they were tested, a test that was never feasible prior to the availability of computers.

In fact, in almost every theoretical field involved with studying human thought and behavior, major holes were discovered as AI researchers attempted to use the theories to program machines. Many bedrock concepts were found to be devoid of the capability to fulfill their expected roles. As a result, AI has put a lot of pressure on the theoreticians in a number of fields. It's no longer enough to say "Here's how I think people do this," or "Here's how I think it can be done." With the advent of AI, the theories can be embodied in electronic machines, and be shown to work or to fail. As such, the new methods employed by AI scientists as they push the frontiers of machine intelligence have also brought new vigor to the many associated disciplines relevant to their quest. Consequently, along with producing computers that contain the seeds of thought, AI is helping create a fascinating -- and more accurate -- window into our own minds.

Spurred on by AI, the scientific and philosophical debate about the nature of human and machine intelligence has become extremely vibrant. But the debate sometimes draws attention from the very real advances now occuring in AI. Regardless of whether these advances garner the "intelligence" label, they are certain to have a profound impact on our lives. And they will achieve this impact, often in surprisingly simple ways, by making computers exhibit skills long presumed to reside exclusively in the domain of human beings.

A Whirlwind Evolution

The theory of computing dates back more than a century, to when Charles Babbage postulated an "analytic" machine and Ada Lovelace performed what is commonly regarded as the first machine programming. But it was only four decades ago that the first modern computer was constructed. Developed in secrecy by J. Presper Eckert and John Mauchly at the University of Pennsylvania during World War II, the ENIAC computer was publicly unveiled in 1946. In the 40 years since, computer scientists have amazed the world with an astonishing succession of rapid advances.

Foremost among the computer milestones have been developments in the actual device machinery, or hardware. In fact, so-called "generations" of computers have been measured primarily by specific hardware advances. The 1960s creation of the transistor replaced the first-generation vacuum tubes. The development of integrated circuits on a semiconductor chip in the 1970s; and the continued shrinking of electronic components through large-scale integration (LSI), which became practical only several years ago, led to further generations. The last, LSI, has brought us to the current "fourth generation" of computers.

This generation is characterized by a diverse group of computing machines. Inexpensive microprocessor chips no bigger than a fingernail already squeeze the equivalent of hundreds of thousands of transistors onto their tiny surfaces. Supercomputers costing millions of dollars have attained processing speeds of half-a-billion operations per second, and can perform in minutes tasks that once took months, or were totally intractable. In what promises to be the first step in the long-predicted arrival of computing for the masses, personal computers have begun to proliferate throughout our businesses, homes, and schools.

While hardware advances will continue to play a crucial role in the evolution of computers, the relative importance of software, the programs that orchestrate the operation of the machine circuitry, has grown dramatically. Just as hardware has progressed through various evolutionary stages, so has software seen several distinct generations. Software languages have evolved independently of the computer hardware advances, but, coincidentally, these languages are also now at the fourth generation. As it happens, the emerging fifth generation of each is critical to the advancement of AI.

Software Development

Computer hardware improvements promise to deliver incredibly powerful machines at a fraction of the cost of today's fastest supercomputers. As such, all fields of computer science stand to benefit, since more powerful machines mean faster processing, larger memories for storing information, and the ability to manipulate this information in ways not practical before. But of all the computer sciences, artificial intelligence stands to benefit the most from improved computers. Not only will the new machines permit the faster execution of complex AI programs, they will also serve as development aids that researchers can exploit as they delve further into the subtleties of machine thought. Despite the leverage next-generation hardware will provide, however, the immediate future of AI is primarily dependent upon the realm of software programming.

Software is critical in the development of AI because it's the computer programs that direct the machines in their attempts to perform humanlike tasks. These tasks include inference and deduction and the use of rules-of-thumb, or "heuristics." In performing such operations, the software must also rely upon an extensive storehouse of information programmed into the computer's memory.

Knowledge Processing

Combining the appropriate background information with various rules of reasoning permits computers to perform what is sometimes called "knowledge processing." Such processing represents a quantum leap beyond the "data processing" and "word processing" exhibited by most of today's computers. To a traditional computer, for example, the word "eagle" means nothing; it's just a series of letters (which aren't even recognized as letters by the machine) entered by a human operator. Most people, on the other hand, have numerous associations for -- or knowledge about -- the word: it's the name of a large bird, it's the symbol of the United States, it's an endangered species, it's a skillful hunter of other animals, and on and on.

To move beyond traditional processing and achieve knowledge processing, the computer must also "understand" something of the relationships between different pieces of data. The knowledge important for a computer to understand is heavily dependent upon the tasks the computer must perform. The word "eagle" has no place in a system dedicated to accounting tasks; it may be sufficient for a natural language understanding computer to merely recognize the word as a noun with a specific definition; and a system designed to contain and manipulate ornithological information may store volumes of data associated with the word.

Of course, hundreds of books already contain such detailed information, and we never presume to label them intelligent. But when a computer uses its processing rules to discern relationships that aren't explicitly stated in its memory, or when it draws conclusions that are beyond the grasp of all but a few experts in the field -- tasks within the reach of today's machines -- the lines between human intelligence and computer knowledge processing begin to blur.

The Roots of AI

As a field, artificial intelligence got seriously underway in 1956 when a group of computer scientists convened the first AI conference at Dartmouth College. The organizers included John McCarthy (who is credited with coining the term "artificial intelligence"), Marvin Minsky of MIT, Nathaniel Rochester of IBM, and Claude Shannon of Bell Laboratories. The youth of the overall computer field and the seemingly boundless potential of AI combined to create an air of heady confidence in the early years of the new discipline.

Although the computers of the 1950s seem hopelessly inept when compared to their modern counterparts, their capabilities seemed almost miraculous at the time. If these machines could perform numerical calculations at blinding speeds, why couldn't they perform a wide variety of feats better and faster than human beings? Many of AI's initial predictions for success were dashed, however, as the difficulties of creating intelligent-acting computers gradually became apparent.

The Setback from Rash Promises in Language Translating

Although various researchers did achieve impressive results in narrow AI fields, some widely publicized experiments failed grandly. Most notable of these was in the area of machine translation. It was thought that properly programmed computers could become automatic translators of, say, Russian to English. Instead, the early machine translators became subjects of ridicule, not respect, as their output often proved more ludicrous than accurate. The machines simply made direct conversions of words from one language to another without taking such factors as the words' content and possible multiple meanings into consideration.

Failures like those produced a general skepticism about all AI endeavors, a skepticism that seriously wounded the overall field. Virtually overnight, AI researchers found their funding evaporating and their professional support dwindling. Computer research continued full steam, but only in such "practical" areas as semiconductors, computer languages, machine architectures, and peripheral devices. AI developers had committed the cardinal sin of promising more than they could deliver, at least given the available technology, and they paid for their miscalculations. The research was abandoned by all but a few corporate and university laboratories, and the remaining AI researchers often labored under a reputation of being dreamers who simply couldn't accept the limits of machines.

Now, after years of obscurity, AI is back in vogue. For the first time, products that employ artificial intelligence techniques are successfully entering the commercial market. In fact, the accepted benefits of AI are now so great that companies are falling over themselves to place the "artificial intelligence" label on their products, deserved or not. As recently as two or three years ago, such a label would have probably driven away more customers than it attracted. Today, the association with AI is perceived as being a potential gold mine. No doubt many AI scientists find it amusing that, after enduring their long-standing exile from computer science's limelight, they are suddenly much-sought-after gurus of computing's new wave.

A Market Ripe for AI

A number of factors combined in the late 1970s and early 1980s to prepare the way for AI's emergence from the labs. The proliferation of computers throughout businesses and homes placed the machines in the hands of people not skilled in their operation. The very increase in the numbers of installed computers helped create markets large enough to offset the substantial development costs usually associated with artificial intelligence products. Computers also entered factories where, in concert with programmable robots and other systems, they became responsible for conveying parts, assembling products, controlling inventories, and projecting demands. Meanwhile, noncomputerized fields grew in complexity, creating new types of opportunities for computers to exploit.

Artificial intelligence is not by any means a narrow field, and AI researchers work in a variety of specialized disciplines. Two main branches of AI have exerted the most impact on solving near-term problems. One, natural language processing, lets people use their own language -- be it English, French, Japanese, or whatever -- to communicate with computers. This capability eliminates the need to learn awkward computer languages, and gives a much broader range of people access to computers. So far, natural language understanding is essentially limited to words typed into a computer's keyboard, but it will eventually be extended to include spoken words.

The other AI branch, and the one that has received the most publicity, is that of expert systems. As the name implies, these systems contain the knowledge of experts in specific fields, along with the reasoning rules these people employ to manipulate their knowledge and arrive at conclusions. Given this capability, expert systems can serve as assistants to people, giving suggestions and opinions, and explaining the derivation of these if asked. The best-known expert systems include some in such fields as medical diagnosis, geological surveying, and, appropriately, computer system configuration. Natural language and expert systems are complementary, in that they can work in concert to produce a very-easy-to-use, very "smart" computer.

A Daunting Task

The power that AI techniques bring to computers is impressive, and the research that developed these techniques was, and is, long and arduous. But the actual functioning of AI-based computers is not difficult to understand. Since the machines use many of the methods employed in mathematics, the general concepts underlying their operation are surprisingly simple to grasp. It's just when the computer brings its great speed and memory into play that these relatively simple procedures produce hard-to-believe capabilities.

Still, the programming of computers with the rules and information needed to perform humanlike tasks at anywhere near the level of a person is a daunting job. As a result, many working in AI look askance at some of the products now claiming to incorporate artificial intelligence. The indignation felt by some researchers about the misuse of AI labels by less-than-deserving products goes beyond mere snobbishness about "true" AI. Rather, they fear that a surfeit of outlandish claims for AI may cause another backlash against the field if the claims fall flat. Many researchers believe artificial intelligence has almost boundless potential, but even its most ardent proponents admit that the realization of many AI capabilities will be a fairly slow process.

The Japanese Fifth-Generation Project

Market realities have added some speed to this process, however. Most notable in this regard is Japan's 10-year Fifth-Generation Project. Scheduled for completion in 1991, the project -- which coordinates government, university and industrial research -- has a goal of using AI and new hardware designs to take computers a quantum leap beyond anything existing today. If successful, the Japanese could become extremely strong competitors in the computer industry, although it's unlikely that success would bring them total market domination, as some fear.

As with any frontier research effort, it's difficult to assess the likely end results of the Japanese project. But one result has already become evident: The Fifth-Generation Project has stimulated the inception of hundreds of other AI projects around the world. Much of this activity, like that in Japan, is government supported. This is true in the United States, where the Defense Department's major research funding agency has plans to spend about $600 million over the first five years of a broad project intended, in part, to develop machine intelligence. Few expect or want the United States to totally mimic the Japanese strategy of coordinating a university, industry, and government triad to work on well-defined, mutual goals. Even with the massive growth of government-sponsored research in this area, start-up entrepreneurial firms and established computer companies will remain a crucial element in the furthering of artificial intelligence in this country.

Both the United States and Japan, not to mention the European countries active in AI, will likely realize a good measure of success in future years. The potential AI market is huge, and is certainly large enough to support numerous participants. If one company or country gets a jump in some areas, other competitors will lead in different AI fields. This is not to say that the research and development activity in Japan should be ignored. Far from it. But the doomsday alarms sounded by some about the United States' AI prospects often ignore this country's strong lead in artificial intelligence, and the amount and quality of ongoing U.S. research.

The U.S. Shortage of AI Talent

Nevertheless, problems do exist in the United States. Most notable is a shortage of AI talent and the resulting strain likely to be placed on universities. When any research field suddenly becomes a darling of the commercial world, universities suffer an exodus of talent. This exodus is especially severe in AI, where the existing pool of knowledgeable researchers is extremely small and the potential commercial gains very large. By its nature, artificial intelligence work takes two to three years for a single project. This means long-term commitments are required of researchers, be they in academia or industry, and people with the necessary combination of commitment and skill are very hard to find. If AI's success is not to be limited, we need to seriously consider the shortcomings of our educational system, which must produce large numbers of skilled graduates to meet the market's growing demands.

Computers as Tools

As grand, or as threatening, as AI's destiny may be, that destiny is still no more than a multihued vision. But AI won't wait for its ultimate incarnation before it begins affecting people's lives. Its ability to bridge the gap between people and computers is already becoming evident. In a world where more and more people have to deal with computers regularly, AI is playing the ironic role of removing the "artificialness" of these human-machine interactions.

Making Computers Act More Like People

Once all is said and done, after all, the result of AI programming is to make computers act more like people. Whether this phenomenon is viewed as magic, as sacrilege, or as good science, it makes our interaction with computers much easier and much more powerful. In fact, "artificial intelligence" is an unfortunate misnomer for a field that really makes computers seem much more natural to us than conventional machines.

In the near term, AI's main effect will be to make computers more "invisible." That is, AI will present the user with a familiar face that disguises the workings of the underlying technology. At the same time, AI programs and methods will allow computers to break free from their traditional roles as number crunchers and as repositories of data. AI computers will store knowledge -- be it about birds, medicine, or grammar -- and this capability will make them more powerful tools than any existing machines.

Of course, the very power that can make AI-based computers so useful can also give rise to suspicion and fear. Few people today feel threatened by machines that perform physical labor, but this was not always true. And computers that encroach more and more into the realm of our mental processes may create much more of a challenge to our sense of individual worth and uniqueness. Will AI engender the formation of present-day Luddites? And, if so, will they be working for a just cause?

Computers Becoming a Tool for Dealing with Complexity

Probably the answer to both questions is no. If anything, our world promises to grow more complex, and computers will constitute our primary weapon in coping with this complexity. The questions about AI are no longer "Can it work?" and "Will it have a role?," but "What of many possible roles will AI play?" and "How independent will we permit computers to become in fulfilling these chosen roles?" For instance, an expert system that assists a doctor in making a diagnosis is one thing, but a system that independently performs a diagnosis and prescribes a treatment is quite another.

Properly managed, artificial intelligence promises to enrich and broaden our lives, not to make them less meaningful or less secure. AI will become so pervasive, and its power to effect change so broad, that it behooves people to understand at least the fundamental tenets of the field. Given that understanding, AI no longer seems so mystical or so threatening. And, with such understanding, people play informed roles in helping direct the applications of this amazing technology.

Ω

Newsletter – *Continued from page 3*

cannot get behind the image, the display provides a 180-degree viewing field, meaning that the object can be examined from the front and sides.

Dr. Benton said the technology used to create the holographic images was "complex," but that a combination of lasers, mirrors and computers basically reproduce the direction as well as the intensity of light. "We compute all the views," he said, "and fire them into a hologram, which later fires them back to our eyes."

Ω

Image Processing and Records Management

John E. Ashby, Jr.
Executive Vice Pres.
Teknekron Financial Systems
Berkeley, CA 94704

"Millions of pages can be made accessible to any number of users worldwide."

Reprinted with permission from *Recognition Technologies Today* for June, 1986, published by Recognition Technologies Users Association, P.O. Box 2016, Manchester Center, VT 05255.

Shuffling Paper

Paper, paper, paper. I don't know about anyone else, but I get tired of shuffling paper every day. Whenever I want a certain letter or contract, it's invariably missing. Either someone else is working with it, or it's been misplaced. Or it's hiding under that pile of papers on my desk and I don't know it.

Perhaps I want a contract that's in the Boston office. Then I have to call Boston and have someone FAX it to me. Either the FAX machine isn't working or the contract comes through so blurred, I can hardly read it.

I could ask that a copy of the contract be sent by overnight courier, but that's not terribly efficient if I need the information immediately. And overnight courier services can get expensive. Plus, I've just generated even more paper by having a copy of the original document sent to me.

What if we could do away with all that paper? What if we could simply call up an image of that contract on a screen? Or have a hard copy of the document in seconds?

Accessing a Central File

I agree that the paperless office will never be a reality. However, there is a way to rid ourselves of a lot of the paper we have to deal with every day. There is a way to access documents from one central file, no matter where you are in the company or what geographic location you're in.

It's an image processing technology called automated records management. Image processing takes in a pretty wide spectrum of technology, but in this article, I'll focus the subject to deal with records management.

Image processing is the technology of converting a paper document or film copy of a paper document to digital computer data. Once that document is converted to digital data, it can be treated as any other computer data. It can be stored, retrieved, transmitted, displayed or printed.

Automated Records Management

Automated records management is generally thought of as merely customer service or archival and retrieval. It's a lot more than that, however.

Records management is applying image technology to replace or reduce paper flow during the entire processing cycle. It should also deal with more than just machine-readable documents such as checks. A truly automated records management function should be able to handle mortgages, loan files and customer correspondence. It should deal with the type of document that can't be economically machine read and stored as data.

What company needs records management? Any organization that must deal with massive amounts of paper -- insurance companies, banks, hospitals, credit card companies -- needs an automated records management system.

These businesses can use records management for application processing, transaction processing, remittance processing, customer service and archiving.

Dramatic Reduction in Paper Flow

The most obvious benefit of an automated records management system is the dramatic reduction in paper flow. During application, transaction or remittance processing, paper moves from workstation to workstation and must be handled, stored and controlled. Sometimes 20 per cent of the labor force is devoted to these "empty" tasks.

Documents are generally filed in central cabinets remote from the user, and clerical interface is required to access those documents. Redundant filing systems tend to spring up over time, undermining file integrity and increasing duplicating costs. And the logistics of filing, reproduction and distribution are highly labor intensive.

Amount of Paper Handled

Here are some examples of the amount of paper handled each day by large organizations:

At a large East Coast insurance company, a staff of 300 file clerks needs to refer daily to an insurance policy file of 50 million individual documents.

In a large medical claims processing operation, 3,000 claims are received daily, resulting in a 90-day pending file of over 2.5 million documents.

The top 10 mortgage servicers in the United States have over 2.6 million loans in working files.

In a normal month, a large mortgage servicer will have about 21,000 to 35,000 copies of film to access.

An automated records management system would allow these businesses to employ a much smaller staff to maintain files, and requests could be handled in seconds instead of days.

Making the Management of Records Automatic

How can records management be automated? Today there are a number of systems that employ computer-assisted techniques to manage paper. Most fall short of freeing workers from chains of paper, though.

Ideally, when implementing such a massive, sophisticated system, engineers and users need to work jointly to build a detailed fact base to determine a company's needs. This is an involved consultation period that can last several months. After a full understanding of these needs, a system can be engineered that will give the user a practical, achievable solution and an attractive payback as well.

The result is a unique approach that integrates advanced data and word processing technology with a system that electronically captures, reads, stores and distributes document images anywhere within an organization. Several recognition technologies are used in capturing the data -- OCR, MICR and bar code reading can all be used in this ideal system.

Image-Processing Custom-Engineered

This automated records management system using image processing eliminates paper or microform processing after the initial handling, thus improving productivity, reducing labor and improving control and customer service.

What would such a system look like? While each one is custom engineered, every system has:

- A single central file for the automated storage of millions of documents of any type on a variety of microform formats or on optical disk

- Proprietary software for indexing the documents and for retrieving the electronic image or hard copy of any desired document within seconds

- Simultaneous access to any desired document by authorized users at any number of local or remote locations

- Proven hardware components integrated into a flexible, expandable system tailored to the client's unique current and future needs.

In other words, users can sit at the keyboard of a video terminal anywhere in the client's network and gain access to document images stored on microfilm, microfiche or disks and have, within seconds, any of these images displayed on the screen or converted to hard copy.

Keep in mind that each system is custom engineered, so any description will be only a general overview. However, I can give you an idea of what makes up an automated records management system.

Core Components

The core components include a control system computer, a digital disk mass storage image buffer, a high-resolution image display terminal with IBM 3278 capability and custom indexing and data base software.

The master controller is usually a microprocessor. It is chosen from any one of a number of vendors to fit the throughput requirements of the overall system. Standard configurations also exist for the IBM mainframe environment.

The microprocessor controls storage, retrieval, printing and image processing, while a video subsystem provides the means to connect image capture devices and hard-copy printers. Its modular architecture provides for great flexibility.

A digital disk mass storage image buffer is also included in this cabinet. The system needs to be designed so that vast amounts of image data can be stored on microform or optical disks and moved rapidly to video terminals to ensure that terminal image processing is completed within specified time-frames.

High-Density Disks

This is accomplished by using high-density magnetic disks. The approach is relatively simple in design. The total mass storage system is composed of image buffers connected to the network. It consists of interface and control electronics along with as many as eight Winchester disk drives of up to 660 million bytes each.

The response time for accessing images on one of these buffers averages five milliseconds. To provide even faster access, the buffer can also be used to buffer up succeeding images from microform or optical disk files on the basis of known access patterns.

High resolution image display terminals usually consist of monitors, keyboards, refresh buffers and terminal controllers that also function as IBM 3278 terminals. The monitor's 9-by-12-inch display area has a resolution of 1,000 TV lines. Larger terminals can display full 8-½-by-11-inch images or even larger for engineering drawings, maps and so forth. The terminal can be used in any of three modes -- video only, 3278 alphanumeric only or combined video with alphanumeric.

Central to this system is the ability to locate and display the proper image quickly and efficiently. The structure of the software allows users to retrieve documents with ease according to the same conceptual methods now employed in a manual environment.

Multiple Paths of Reference

This is accomplished through the use of multiple indexes, document abstracts, chained retrievals and assisted or approximate searches. Indexing software provides for multiple indexes to the image library, thus giving users considerable flexibility in document retrieval. A multiplicity of document images can be retrieved and stored with a single request. Various types of assisted or approximate searches can be made.

Two types of devices are used for storage and retrieval of microform data: the roll film device and the microfiche device. The roll film storage and retrieval device has the capacity to store 300 rolls of microfilm. The response time to user commands ranges from five seconds to 20 seconds, with an average response time of about 12 seconds. At a 50X reduction ratio, each roll of microfilm may contain between 8,400 and 14,000 page images.

The microfiche storage and retrieval device has six major components:

Carousel storage drum

Fiche selection mechanism

Optics

Camera

Control microcomputer

Support electronics

The carousel drum holds 780 microfiche. Depending on the reduction ratio, the total storage volume of each drum is 64,000 to 305,000 page images. These devices can be racked in a cabinet much like magazines or phonograph records.

Optical Disk Storage

The technology for storing images on optical or laser disks has reached the stage where optical disk storage systems are available for delivery as standard commercial products. There is a less than one second response time to obtain images from a disk that is already mounted. A single laser disk can store between 25,000 and 40,000 8-½-by-11-inch pages.

Each optical disk on-line archival system -- called a jukebox -- can hold up to 100 disks or four million 8-½-by-11-inch pages. Accelerated aging and stress tests have demonstrated a useful life for disks beyond 15 years.

Optical disks are currently "write once, read many times." These permanent records are appropriate in archiving document images but are restrictive in pure data processing environments.

Laser disk costs are declining rapidly as more vendors produce high-quality disks. Soon the media cost for microform and disk will be equivalent.

Laser disks and jukeboxes are particularly useful in records management and customer service applications where fast access to millions of records is required.

In many records management applications, the source documents for archival storage originate as paper or microform. The indexed document is automatically entered into the system for long-term storage on optical disk.

Varieties of Output

There are several types of output products. Images stored on the system can be converted to hard copy or microform through a variety of devices. Indexed software can access multiple stored images, transfer them to the disk buffer and then output them to hard copy in sorted order.

The user then has the capability to produce such items as documentation packets and account history files quickly and easily.

Millions of Pages Accessible to Any Number of Users Worldwide

Software can support remote video terminals and hardcopy printers. A single central file has millions of pages, each accessible to any number of users on a worldwide basis within seconds. The system provides a single source of accurate, up-to-date information for all users, regardless of their geographic location.

The benefits from an automated records management system based on image technology include improved control which reduces risk and gives better security and inventory control.

The operator efficiency achieved from this system means being able to remove the records area from functional work flows. Companies also experience improved file access time, easier file updating and mass production of file copies.

Much Better Control over Costs

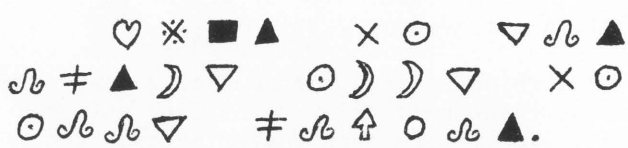
Organizations realize much better cost control, because the system eliminates microfilm costs, improves productivity, controls labor costs and allows file copies to be produced only when required for output.

The system has expansion capabilities so that the core system grows with increased activity. It can interface with accounting/production systems, and it can incorporate other records management functions.

What has been described in this article is an advanced system that allows companies to reduce their paper load and to increase their productivity. It's the only way available today for a business to keep from being buried, literally, in paper. While many people think of records management based on image processing as merely archival/retrieval systems, I believe that image processing technology can deliver significantly greater benefits when integrated into the entire processing life cycle. Ω

Games and Puzzles – *Continued from page 28*

MAXIMDIDGE 8609



SOLUTIONS

NUMBLE 8607: Nature leads the way.

MAXIMDIDGE 8607: Men may come and men may go – I go on forever.

Our thanks to the following person for sending us solutions: T. P. Finn, Indianapolis, IN – Numble 8605, Maximdidge 8605. Ω

Silver Bullet Opportunities

George Heilmeier
Senior Vice Pres. & Chief Technical Officer
Texas Instruments, Inc.
P.O. Box 2909
Austin, TX 78769

"We look for new technology, markets, and new products ... A silver bullet is something that we could not do before."

Based on a talk presented at the TI-MIX Symposium 1986 of the TI Users Group.

Changing the Ball Game

The electronics industry is one of the five largest industries in the world today. Back in 1960, it was roughly a $30 billion business. By 1990, we expect it to be $670 billion. At the same time the semiconductor revolution, which fueled the end equipment market, started as a market of roughly $1 billion in 1960 and by 1990 we feel it will grow to $70 billion. So we're all in the right business. Now what's the problem? Well, the problem is competition. It used to be that we had a positive balance of trade in high technology electronic equipment. That all changed about 1983 and now the balance is negative and seemingly getting worse. Now, there are many reasons for this, some are political, some are economic, some are cultural and some are financial. But when a technologist like myself gets involved in something of this nature, we tend to think in a different direction -- mainly, how can we change the ball game. There's some who think that the ball game cannot be changed. Those are members of what I'll call the "God is dead" school of technological thought. A prominent analyst on Wall Street observed that he didn't think, or he wasn't sure, that there were any really new electronic markets out there. The chairman of the board of one of our major West Coast semiconductor competitors observed that he didn't know what we were going to do with VLSI (very large scale integration), and VLSI is really the technological imperative of the semiconductor industry.

We technologists live in the future, and so we would like to associate ourselves with Charles Kettering, the former chairman of General Motors. Kettering observed, "I'm interested in the future because I expect to spend the rest of my life there." That's the way that we in the technical community at TI feel. We can't do anything about economic factors, financial factors, political factors or cultural factors. But we can change the ball game.

Silver Bullets

We look for new technology, markets and new products. These new markets and products are what I call silver bullets. Now what is a silver bullet? It's characterized perhaps as a new function, something that we couldn't do before. Or perhaps it results in a major cost or size reduction in an existing function or it may involve a major performance improvement at no additional cost.

Let's take a look at a few silver bullet opportunities that could very well represent billion dollar markets in the late 80s and beyond.

Symbolic Processing Engine in a Shoe Box

We believe that the symbolic processing engine in a shoe box will make possible the era of imbedded expert systems or imbedded artificial intelligence. Instruments will become smart instruments participating not only in their own diagnostics and maintenance but also in the interpretation of the data that they collect. We might find symbolic processing engines imbedded in all types of vehicles: airplanes, trucks and military vehicles. So the future of symbolic processing in large measure will be influenced by imbedding that technology and applications. It will not necessarily exist as stand alone resource.

Intelligent Data Base Management Box

You know when we really get down to it, our customers really don't want more complex instruction sets or more memory. What they really want is easy query and fast response. When I talk about the intelligent data base management box I'm talking about a box that can extract information from data. It can personalize that information. It can enable you to browse or pursue goals over time and protect the security and the validity of the information in the data base.

Speech Understanding Box

Today's speech understanding capability is relatively rudimentary. What I'm talking about is large vocabulary speech understanding which is speaker independent, continuous, and performs in a robust environment.

Image Understanding Box

Another silver bullet opportunity is the "image understanding box." Today's image processing systems are very limited. They have difficulty when an object is reoriented or when the light direction is changed or when a shadow from an adjacent part appears suddenly. When I talk about image understanding I mean the ability to recognize images in complex environments including human faces.

Design Synthesis Box

Today's design automation is a wonderful tool when you know what it is you want to design, but it doesn't participate very extensively in the area of design synthesis. This is a whole new area I believe will become very, very important to us in the 90s.

Automatic Programming Box

The automatic programming "box" is almost self-explanatory. It deals in the context of telling the machine what it is you'd like the machine to do as opposed to telling it how you'd like it to do it.

Incremental Computer

And finally, the incremental computer, the cost-effective, massively parallel, computational box. A great deal of work is ongoing in this area today, but so far we haven't succeeded in building systems with large numbers of processors with processing power growing linearly with the number of processors. It must also grow in a way that is entirely transparent to the user so that computational power can be added incrementally.

The Challenge of the Technology of Very Large Scale Integration

There are two technologies that underlie those seven silver bullet, billion dollar market opportunities of the 90s. Those two technologies are Very Large Scale Integration (VLSI) and Artificial Intelligence (AI). I'd like to walk you through some of the aspects of these two technologies. Let's begin with VLSI. What do we mean by VLSI? VLSI means at least a half million devices on a single chip, but it also means a new level of capability to solve systems problems of all kinds. You may have seen charts hanging on the wall of your chemistry and physics classes in high school -- the periodic table of elements. For us, that table really only has one element -- silicon. Silicon is the basic building block material of the VLSI era. A great deal of progress has been made since the invention of the integrated circuit back in the late 50s. We've made each element a lot smaller. We've made chips bigger so that we could put more elements on them and we've learned how to package those chips more efficiently. As a result, we've increased the number of devices per chip by 10 times per decade over the last three decades. Now if automotive technology would have made the same kind of progress of the last 20 years or so that semiconductor technology has made, we would be looking at a Rolls Royce that cost less than $3, got over 3 million miles per gallon, delivered enough power to drive the Queen Elizabeth II, and six of them would fit on the head of a pin.

We in the VLSI business have four challenges. We must learn to deal with minimum feature sizes of less than one micron. A micron is a millionth of a meter and during my presentation here tonight, your fingernails will grow more than one micron. We also must deal with design complexity and facilities and people. These are our four challenges.

Let's take a look at "minimum feature sizes less than one micron." Today's 1 megabyte DRAM (dynamic random access memory) chips are designed with one micron minimum feature sizes. That means that the minimum feature size is about the size of a yeast cell. That's roughly a hundred times smaller than the size of a human hair. By 1988, we expect to see some prototypes of devices made with half-micron minimum feature sizes, and that will mean that the minimum feature size will be somewhere between the smallest bacteria and yeast cell. We build integrated circuits on silicon wafers. This is a typical six-inch silicon wafer containing several hundred 4 megabyte DRAMs. Now let's take a look at relative sizes. A 4 megabyte DRAM chip is about the size of a third to a quarter of a dime. It contains 4 million bytes of memory. The minimum cell size is about the size of a red blood cell.

More Challenges

The second challenge we must face is design complexity. Back in the mid-60s when we first started to fabricate medium scale integrated circuits, the complexity of those circuits was equivalent to roughly a street map of Highland Park. Today's microcontrollers have a complexity that's equivalent to a street map of the entire Dallas/Ft. Worth area. One micron technology similar to that which we use in our 4 megabyte DRAM has a complexity that's roughly equivalent to a street map of the state of Texas and the four surrounding states. In the 90s, when we finally approach one-quarter micron technology, the design complexity will be similar to that of a street map of the entire North American continent.

Our third challenge is facilities. The cleanest operating rooms today are Class 10,000 clean rooms. By that I mean that there are no more than 10,000 particles in a cubic foot, none of which is larger than two-tenths of a micron. Today's clean rooms in which we fabricate VLSI circuits are Class 10 clean rooms. That means that there are no more than 10 particles per cubic foot in those rooms, and these particles are smaller than two-tenths of a micron.

I want to say something for emphasis. Requiring that there be no defects larger than a tenth micron or so in a VLSI chip is like requiring that there be no potholes on any street in the Dallas/Ft. Worth area for traffic to flow. That's the kind of level of technology that we deal with in VLSI at TI today.

The final challenge that we face is people. It's interesting to note that the U.S. integrated circuit industry is roughly a $6 billion industry and there are roughly 2,250 key people that drive that industry, designer and process engineers. By comparison, professional sports is roughly a half billion dollar business, and the number of key people in professional sports in the United States is just about the same as the number of key people in our microelectronics industry.

Comparison with Life Processes

Let's take a look at where we are in producing semiconductor memories today. Our industry produces roughly 10^{14} bits per year. Now, compare that to the human brain. The human brain contains approximately 10^{14} synapses. If you have teenagers at home you know that number is grossly exaggerated. But what it means is that roughly our IC industry produces the equivalent of one human brain per year. We have a "tongue-in-cheek" challenge in the semiconductor industry. We need to develop a manufacturing process for VLSI that's similar to mother nature's process from the standpoint that the bytes produced per direct labor hour is equivalent to mother nature's process, and job satisfaction is equivalent to that of mother nature's process. Who knows what the world of semiconductor manufacturing may look like a decade or two downstream for us?

What about biochips? I get about one phone call a week from people in the investment community asking me about biochips. You can't pick up a trade journal today without finding at least some mention of biochips or bioelectronics in one form or another. To be sure biological materials are plentiful and they're certainly cheap, but will they make good semiconductor devices? The answer to that is "no" because they lack one of the fundamental properties that determine the speed of semiconductor devices, and that's charge carrier mobility. So don't look for biochips to play a major role in the electronics of the near future. There are, however, a lot of things we can learn from biological systems, such as computation without long interconnects, computation without isolation between each device, functional devices versus collections of transistors, resistors and capacitors, and finally, nonlocalized storage concepts. Biological analogies will be an important area of research for us in the microelectronics business for the rest of this decade and beyond.

We're making history in VLSI at Texas Instruments. In the last six months, we prototyped the world's first 4 megabit DRAM. Just when everyone was counting out U.S. semiconductor manufacturers in the DRAM race, TI produced the world's first 4 megabit DRAM device ahead of the Japanese. We also released the design of the world's first 32-bit microprocessor, whose architecture is specifically oriented toward artificial intelligence, and we've continued our string of first-pass design successes that began in 1984. We have not had an integrated circuit at Texas Instruments fail to be functional on the first pass since the middle of 1984.

The Promise of Computational Plenty

Let's go back to this issue of what do we do with VLSI. Up until the late 60s, for the most part, we used whatever computational resources were available to crunch numbers. Beginning in the late 60s we began to take more and more CPU cycles and apply them to the user interface. This is a trend that I expect to continue and by the 1990s perhaps 90 percent of our CPU cycles will go into the interface and 10 percent will be used for numerical processing. This is made possible by the fact that computational power increased by 22 percent per year up until the late 70s. It's now on a much steeper rising curve, 44 percent per year. Now what will this mean? Well, we'll be using VLSI to solve the interface problem. That means we'll be making free format access to computer power a reality. We'll do without computer-ese and we'll begin to build adaptive systems, that is systems that adapt to the level of the user instead of forcing the user to adapt to the system. What will VLSI mean? VLSI will mean an age of computational plenty, the availability of low cost computational power. It will mean systems that adapt to the user instead of vice versa and it will also mean the era of artificial intelligence.

Before I move to the second technology that I feel is going to be one of the major technologies of the late 80s and beyond, I'd like to conduct a brief review of where we've been in digital computing. It all started back in 1945 at the University of Pennsylvania with the world's first electronic digital computer, the Eniac. That computer was used for engineering calculations. It was used to compute ballistic trajectories for the United States Army, and the engineering wave of computation continues today. Big names in the engineering computational field today are names like Cray and Control Data. Back in the early 60s we entered what I'll call the "machine productivity wave" and that existed side by side with the "engineering wave." That started with the introduction of the IBM 360 system and continues today with super minis and personal computers. I believe we're just beginning to see the advent of the third wave, the people productivity wave. That's the age of transparent complexity and artificial intelligence.

People Efficiency Improved with AI Systems

Let's compare general purpose computing with AI systems. In the general purpose computing world for the most part, our goal is people efficiency. The kinds of problems that we address are generally problems that require frequent repetition. Some examples are things like word processing, spread sheets, filing and data base management of one kind or another. In contrast, AI systems address the issue of people effectiveness. The kinds of problems that AI addresses are problems which are characterized by ambiguity, complexity and uncertainty. Applications are things like decision analysis, investment analysis, problem diagnosis, policy administration and maintenance. Back in 1953, one of the first major computer conferences was held in Los Angeles and Dr. Haroon of the National Bureau of Standards was one of the principal speakers. He issued four challenges to the computer designers of the day. He said that a computer would have to cost less than $1,000. He also said that we would have to solve any problem in real time or in one second, which ever was less. He said the system had to be so compact that it could be painted on a convenient surface and finally, he said that we needed to communicate with computers at a level that was equal to the intelligence of the user. We've delivered on computers that cost less than $1,000. We've delivered on computers that can solve problems in far less time than one second. We've made computers compact. We still haven't been able to paint them on any surface, but we've made them so compact so that they virtually are existent on plane surfaces. The challenge that remains is to be able to communicate with its user with an intelligence equal to its user. That is the goal, one of the goals, of artificial intelligence technology.

AI = Intelligent Behavior

What is artificial intelligence? There are many definitions. As a science, it's the understanding of the mechanisms of intelligence. As engineering, it's building systems which exhibit intelligent behavior. General purpose or conventional data processing essentially takes a mechanistic view of problem solving characterized by perception, analysis, and adaptivity. These two worlds are separate today, but over the next decade or so we'll see the merger of the general purpose computing area and the symbolic computing area.

Let's talk about artificial intelligence at TI. Many of you are familiar with the satellite symposium that we held last November. Thirty thousand people attended that symposium; it was probably the largest symposium of its kind ever held. In a survey of participants, we learned that 3,600 or 12 percent had specific applications under development. We started our AI activities at Texas Instruments in 1978. The Japanese started their Fifth Generation or AI program in 1982. In 1980, we prototyped the first version of NaturalLink. In 1982, we applied AI to seismic interpretation in our geophysical prospecting business. We also started the Explorer program in 1982. By 1983, we were using AI in the context of software prototyping and NaturalLink was introduced. In 1984, Explorer was introduced and we started the LISP chip, the 32-bit AI microprocessor. We also introduced Personal Consultant. In 1985, we introduced Arborist, PC Scheme, held the AI Satellite Symposium, and won two of the three major AI programs in DARPA's strategic computing initiative.

High Cost of Expertise

Let's look at some applications of artificial intelligence. The kinds of problems or applications that lend themselves to AI are applications that are characterized by a high level of complexity, are symbolic rather than arithmetic in nature, and there is a scarcity, or high cost, of human expertise. Some of the applications of interest are those involving expert systems. I believe that expert systems are going to be a pervasive technology. They're going to be imbedded in all sorts of systems from sophisticated scientific instruments to equipment that you'll find in your kitchen. Natural language interfaces to computers, no more computer-ese, no more protocols, just a natural dialogue with your computer. It can also mean the extraction of information from data. Another application will be in the domain of image understanding, and add to that speech understanding, as well. AI also will play an important role in robotics and finally, I think AI has the capability to create a revolution in software productivity.

It is of interest to contrast conventional software development with the way software is developed in an AI environment similar to that which exists on the Explorer. In the conventional manner, we start with a requirement specification and we grow that into a set of functional specifications, a set of design specifications, and then finally, an implementation. In contrast, in the AI programming world, we start with a requirements discussion. We do some experimentation in the presence of the user so that he can begin to get a feel for the functionality of the system. Then we produce an incremental prototype. What we have found is that the new way of software productivity enables us to improve productivity by roughly 5 times. We've been able to see the ultimate user interact with the system and watch it mature. Design changes are still acceptable during the development stage and finally the sustaining effort appears to be much less.

Let's take a look at the kinds of things that are going on in our development activities today, activities that will probably see the light of day within the next several years. In hardware systems, one of the barriers to imbedded applications for artificial intelligence is the cost, size and performance of today's LISP machines. We're working on a single chip version of the Explorer processor that will not only greatly reduce the size of the existing Explorer CPU, but result in at least a 500 percent increase in performance. This processor chip will be done in CMO technology which will enable us to reduce the CPU power by 200 times. It will also enable us to reduce the CPU volume by 40 times. This is probably the most complex chip we've ever done at Texas Instruments. Its complexity is roughly 10 times the original Motorola 68000 processor.

Expertness as Time Changes

What about expert systems? There's been a great deal of talk about expert systems. They've turned out to be exceedingly useful in numerous applications, but there's one problem. The expert system software tools that are out there today do not comprehend dynamic environments and so they can't build expert systems that deal with emergency procedures or dynamic factory control. In other words, they can't cope with situations that are changing in time. So what are we doing about it? We're building expert system toolkits that will enable us to work in dynamic environments. Those environments include factory automation as well as things that might assist operators of various types of equipment, such as aircraft. They include process control and scheduling.

Another problem is data base management. Today's data base management systems do not deal very well with multiple representations of data, that is, the merger of alpha-numerics, text, maps, graphics. We're working on object-oriented data base management systems that comprehend multiple representations of data. That is the kind of data

(please turn to page 27)

Conversation With a Computer —Part 2

John Shore
c/o Viking Penguin Inc.
40 West 23rd St.
New York, NY 10010

"We don't know how to write programs that handle errors as well as people do."

This article is based on Chapter 6 of *The Sachertorte Algorithm* by John Shore, copyright 1985 by John Shore, published by Viking Penguin Inc., 40 West 23rd St., New York, NY 10010; it is reprinted with permission. Part 1 appeared in the July-August, 1986 issue of *Computers and People*.

Finding Errors Through Consistency Checks

Bugs can also be detected by consistency checks that are included in the software itself, like the consistency checks in my father's instructions. Unfortunately, such consistency checks consume extra resources -- both memory and execution time -- and most programmers prefer to spend these resources on additional capabilities. Besides, although every programmer knows how hard it is to write correct programs -- something I'll talk more about soon -- most programmers believe that their own programs are correct. As a result, in much of the software that's in widespread use, software consistency checks are sparsely distributed and fairly general in nature. Although many program bugs eventually result in the violation of a hardware or software consistency check, considerable damage may occur first. Furthermore, the connection between a failed consistency check and the actual bug is often so remote as to be unhelpful.

Once the computer detects an error, something must be done about it. One approach is to restart the offending program, or even the entire system, in the hope that the problem won't reappear -- either because it was a transient hardware error, or because it was a software bug that surfaced through a series of unlikely coincidences. But the most common approach is to have the computer system stop executing the offending program and report the error to the user.

When a word-processing or spreadsheet program stops abruptly with messages that blame parity errors, register overflows, subscript checking, protection violations, bus errors, illegal instructions, and the like, the user-interface is shattered -- the messages have nothing to do with writing a letter or analyzing a budget. Such messages are certainly helpful to the programmer who wants to fix the word-processing or spreadsheet program. They are even helpful to the computer-savvy user who can exploit them in working around the problem. To the novice user, however, they are discouraging.

What Would a Person Do?

When asked to set fire to the logs in your fireplace, a friend will oblige cheerfully. Asked to set fire to your house, the friend will at least say, "Are you quite sure?" Is this kind of behavior too much to ask of a computer? Why can't a computer react to 'rm book *' more cautiously than to 'rm book*'? In fact, it can. A computer can notice that you've asked for all your files to be deleted, and it can react by asking for confirmation, but only if it's programmed to do so.

Fortunately, in most cases it's relatively easy to have a computer program detect and double-check on commands that would delete all of your files, and many current systems behave in this way. Indeed, systems tend more and more to protect users from electronic damage caused by their mistakes. Many text-editing and word-processing programs double-check with you before deleting the entire contents of your document, remind you to save copies of changes you make, and the like. Many of them have a general "undo" command that's able to reverse the effects of most single commands. It's easy to see how this might be possible -- conceptually, the program could maintain two working copies of your document, one being the current version and the other being the version just prior to your most recent command. The effect of the "undo" command is to revert from the current version to the prior version.

What Computers Can Do

It's also possible for computers to do a better job handling the consequences of software bugs and hardware failures. Instead of screaming

JOB ABORTED ... FATAL REDUNDANCY CHECK

they can be made to behave more like we did on our way to my mother's surprise party. Again, suitable programming is required. Typically, it's missing. To an extent, its absence reflects a lack of knowledge; we don't know how to write programs that handle errors as well as people do. But the absence of effective errorhandling software also reflects a lack of effort; designers typically choose to spend the available CPU cycles and memory capacity elsewhere.

What about the small, repairable, but frustrating damage I did by typing 'deep' when the text-editing program was in command mode? The resulting exchange of two words isn't the kind of major change that merits confirmation, and the "undo" command wouldn't work because the computer interpreted 'deep' as a sequence of four commands. From the program's point of view, I typed in a sequence of four correct commands, so it seems hard to argue that "the computer should know better" or even that it could know better.

But a person might well suspect that I was confused about the mode. For one thing, the sequence of commands that exchanges two words happens also to spell a common English word. This can even happen for an intentional sequence of commands, but it's enough of an unlikely coincidence to raise suspicions. Having become suspicious, a person might look at my document in the vicinity of the cursor position, realize immediately that the word 'deep' makes sense in that context, and conclude that I could be confused about the mode. Given this evidence, it would be reasonable to check with me before making this change.

Can a computer do this? Partially. Checking a dictionary to decide if a string of characters composes a common English word is an easy programming task. But it's another matter to write a program that can decide if it makes sense to insert that word at a given point in a document. Fair accuracy might be attained just by determining whether the word is the right part of speech required by the context -- adjective, noun, verb -- and people in the field known as computational linguistics know how to write programs that could do this. For example, because the word "deep" can serve as an adjective, such a program could analyze a sentence like

"He took a breath, fast becoming faint,"

and conclude that it's reasonable to insert "deep" before "breath." But the same program would conclude that it's reasonable to insert "deep" into the sentence

"The condor baby fell out of the nest,"

right before "condor". You know that such an insertion makes little sense, whereas reversing the positions of "condor" and "baby" makes a lot of sense. For that matter, reversing "breath," and "fast" in the previous sentence makes as much sense as inserting "deep" before "breath".

These examples show that high accuracy would require programs that can analyze not only grammar, but meaning. This is extremely difficult; there do not exist programs that can analyze the meaning of unrestricted English with anything close to the sophistication of a native speaker.

We Speak Such Different Languages

Every language is a means of expression consisting of a vocabulary and a way of using it. These characteristics are true of all languages, whether natural languages that have evolved to support communication among ourselves or artificial languages that we have created for communication with our machines. The importance of natural language arises from life; the importance of artificial language arises from technology. Mathematics is an artificial language with ancient roots, but it is not as rigorously defined as is commonly supposed; many aspects of mathematical notation are not formally defined, and a correct interpretation of the notation relies on the reader's common sense and general understanding. These informalities do not apply to the artificial languages that we use to communicate with computers. Indeed, the development of computer technology elevated formally defined artificial languages from a theoretical tool to a practical necessity, and it stimulated important advances in their theory and applications. Artificial language permeates the information age.

Syntax and Semantics

It has proved useful to describe languages in terms of two different properties: the form of correct expressions in the language and the meaning of those expressions. The technical terms for these properties are "syntax" and "semantics". Syntax is a set of rules for forming correct expressions. English syntax, for example, includes the rules for forming grammatically correct English sentences. An example is the familiar rule stating that simple English sentences can be formed by a noun followed by a verb followed by an object, a rule that leads to sentences like

"John drinks coffee."

The syntax of a language determines whether an expression has a correct form, but it is the semantics that assigns meaning to the expression. After you read that John drinks coffee, you know that a person named John sometimes pours a liquid into his mouth and swallows; you know that the liquid is made from the roasted and ground seeds of a particular plant that is grown primarily in South and Central America; you know that the liquid is probably hot and probably dark; and you know that John probably ingests caffeine when he drinks the liquid. All of these conclusions illustrate the semantics of one simple English sentence.

The concepts of syntax and semantics apply not only to the natural languages we use when we converse with each other, but also to the artificial languages we've created to converse with computers. Here I don't just mean artificial languages we use in writing software. We also use artificial languages when we type in commands to a word processor or a spreadsheet program, and we use artificial languages when we read their responses. When we tell UNIX intentionally to 'rm book*' or unintentionally to 'rm book *' we are using an artificial language. ...

Syntax Errors

Computers can communicate in terms of these artificial languages because people can write computer software that determines the meaning of language expressions. Ideally, software to do this for a particular artificial language is based on a complete description of the language's syntax and semantics. When you type in an expression using the artificial language, a program attempts to parse the expression, thereby dividing it into its various sytactic components -- analogous to finding the noun, verb, and object of a simple English sentence. If you happen to enter an expression that doesn't fit any of the language's syntactic rules -- a common mistake known technically as a syntax error -- the program should detect that such an error has occurred and reject the offending expression. How gracefully it does so depends on the program. I gave an example of a syntax error as part of the magic trick example in Chapter 2 when I suggested that you might respond to the prompt

PLEASE ENTER A NUMBER BETWEEN 1 AND 10:

by typing in 'SEVEN' or VII'. In order of decreasing grace, the various likely responses were

ILLEGAL NUMBER, TRY AGAIN:

or

FORMAT ERROR

or

SYNTAX ERROR ... ILLEGAL TERMINAL SYMBOL

or

WHAT?

or even a shrill beep without any message at all.

Semantics – Intentional and Otherwise

Once the software has determined that an expression is syntactically correct, it proceeds to apply the corresponding semantics. In the case of the magic trick, the software proceeds to perform the trick with whatever number you entered. In the case of 'rm book*' or 'rm book *', it proceeds to remove the specified files. It's obvious that problems will arise if there are bugs in the programs that are supposed to take the actions deemed by the semantics. Such bugs can arise from programming mistakes as well as from misunderstanding about the intended semantics.

There is, however, a more subtle source of problems: not all syntactically correct expressions are semantically meaningful. This can happen with natural languages; for example, reversing the subject and the object in the coffee example yields

"Coffee drinks John,"

which is syntactically correct but semantically meaningless. Similar problems can arise in artificial languages, and if they occur without being detected, they can result in unexpected behavior.

To use a common example, most computer systems have a command that will display the contents of a file on your terminal screen, the assumption being that the file is filled with human-readable text. To view such a file -- for example, a file called "book" -- you can use a program that I'll call "see"; typically you type in something like 'see book'. But not all files are filled with human-readable text. Some files contain bit patterns that represent CPU instructions and other terminal control codes, and these bit patterns cannot be interpreted as English characters. For example, if you happen to type 'see magictrick', where "magictrick" is the name of a file containing the CPU instructions of the magic-trick program, strange things are likely to happen. The bit patterns in magictrick are meaningful when interpreted by the CPU as instructions, but not when interpreted by your terminal as English characters. But many of the bit patterns for the CPU instructions in magictrick happen to coincide with bit patterns that are normally used to control your terminal, so when you send all of these bits to your terminal by typing 'see magictrick', your terminal tries to obey but proceeds to do wild and crazy things. Typically, random characters splatter all over the screen while the terminal beeps and flashes. When it's all over, you may not be able to resume normal conversation with the computer without turning your terminal off and on again.

The 'see book' and 'see magictrick' commands are both syntactically acceptable -- to use a natural-language analogy, they both have verb-object forms. But only the first is semantically meaningful. Many computer systems nevertheless go right ahead and process the second command as though it were semantically meaningful, the result being behavior quite unrelated to the normal semantics of 'see'. This is a bit like taking the sentence "Coffee drinks John" and trying to interpret it literally, despite its being semantically meaningless.

Another example from the magic-trick program is the consequences of your responding to the prompt

PLEASE PICK A NUMBER BETWEEN 1 AND 10:

by entering '0' (zero). In contrast to its treatment of 'SEVEN' and 'VII', which the magic-trick program rejected on syntactic grounds, the program accepted the '0'. It shouldn't have, as the prompt itself suggests, but it did. The program then processed the zero according to semantics inherent in the magic trick that the program was intended to demonstrate, semantics that didn't apply to zero. It did so till the bitter end, which in this case arrived with a hardware consistency check:

FATAL ERROR... REGISTER OVERFLOW AT AF45
712 547 234 232
777 234 342 455
209 487 439 332
>

This outcome and the outcome of 'see magictrick' are not what either you or the programmer really want. In both cases you are asking for something that's not included in the intended semantics, and in this sense the outcomes are your fault. But the programmer can prevent such inappropriate processing and, for not doing so, deserves some of the blame. Here again, the computer is doing exactly what it's instructed to do.

Artificial vs. Natural Languages

The artificial languages that we've programmed computers to understand are much smaller, much simpler, and much less expressive than the natural languages that people understand. These contrasts arise in part from difficulties in writing the computer programs that process artificial languages. These programs make progress, like other programs, by executing long sequences of CPU instructions. Step by step, the programs have to scan the input language expression, divide it into relevant pieces, determine which syntax rules apply (if any), and take whatever actions are implied by the language semantics. All of this requires that the language be defined completely by the designer and well understood by the programmer.

As the syntax becomes more complicated and the semantics more sophisticated, it becomes harder for the designer to define the language and harder for the programmer to write language-processing programs that work properly. The programs also become bigger and slower -- the bigger, more complicated, and more sophisticated the language, the more instructions are required to process language expressions. And this requires more memory and either more processing time or a faster computer.

In the case of a typical personal computer system, say an Apple IIe or an IBM PC with between 64K and 512K of main memory, the artificial languages are limited severely by the computational resources available. (Keep in mind that the available resources are needed for more than language analysis.) Conversations with such systems are characterized by rigid, unforgiving syntax and simple semantics. On much larger systems, however, people have written programs that can actually converse in English, albeit severely restricted subsets of English. Moreover, these programs converse by exploiting both syntax and semantics, in contrast to programs like ELIZA, which operate almost entirely at a syntactic level. Unfortunately, the language subsets are too small to be really useful, and the computer systems are too big to be commonly available -- tens to hundreds of times bigger and faster than typical personal computers. Even so, the fastest programs take many seconds to process even relatively simple sentences (which is considerably slower than human conversational speed), and they can take much longer.

In between these two extremes are programs that provide a somewhat flexible, forgiving syntax together with limited semantics. These programs are advertised as being able to converse in English, but it's more appropriate to describe their capabilities as English-like. Nevertheless, they can provide a much better user-interface than what is typical today. They are starting to become available for personal computer systems -- typically as a user-interface for data base management programs -- and you will be seeing more of them.

Tomorrow's Conversations

The quality of our conversations with computers will improve. Intellectual advances will lead to better user-interface designs, better artificial languages, better methods for processing useful subsets of natural languages, and better methods for handling errors. Technological advances will make greater computational resources available at lower cost. The capabilities of today's large, institutional computers will be available in tomorrow's desktop, personal computers, so that we will be able to exploit the fruits of our intellectual advances.

One thing will not change. Your computer may make it easier for you to instruct it in accordance with your intentions, but it won't divine those intentions. The computer is a machine whose ability to communicate with its users results from basic principles of operation and from the manner in which it is programmed. Computers not only do exactly what they're told -- barring a malfunction, they don't do anything without being told. Your computer won't read your mind. Neither will it love you or honor you. It will, however, obey you, for better or worse.

Ω

Editorial – *Continued from page 6*

of international disputes; they can help to reduce the exorbitant budgets of war and navy departments; they can work for regional limitation of armaments and back all treaties which tend to avoid competition in arms; they can oppose nationalism and chauvinism wherever they show themselves, in the press, in the schools, on the lecture platform; they can strive to bring order into the chaotic economic and political conditions of the world. ...Wars are man-made, and peace, when it comes, will also be man-made. Surely the challenge of war and of the armament maker is one that no intelligent or civilized being can evade.

[End of Quotation]

Some significant changes and some definite progress have however occurred since World War II. For 41 years no "great war" and no nuclear war has occurred. Horror of nuclear war is almost universal. Countries are choosing to stay out of the armament complex, such as New Zealand, Denmark, and Japan. More than 10 million inhabitants of the Earth have declared their areas nuclear-free zones. More than 30 countries have changed from colonial to independent. The infantry and the marines of the United States armed forces, and the Congress of the U.S., will probably never undertake again a war in South East Asia, after the historic defeat of the U.S. in Viet Nam. Many leaders who are sane have become convinced that they and their families will die in a nuclear war. Accidents at nuclear reactors such as Three Mile Island in Pennsylvania and Chernobyl in the Soviet Union have rendered pro-war arguments weaker than ever.

In World War II there were places to hide. In World War III there will be none. Ω

Heilmeier – *Continued from page 21*

base management system that will be extremely useful for things like design automation data bases, medical data bases, real estate data bases, and business management data bases of all types. A conventional data base deals with a relatively small set of simple structures, things like an employee number or an employee name or a salary. The operations that can be performed on that data set are relatively fixed and limited. For example, you might have a representation or an operation that enables you to ask for things that are greater than or equal to or less than a certain amount. In object-oriented data base management systems, we can merge text, trees, graphs, two-dimensional objects and their descriptions and maps. And in the LISP environment, the operators will be fully user definable. Let's take a look at some examples.

Multiple Representations of Information

In the typical semiconductor design data base, one has a set of specifications that are alphanumeric in nature. One also has a circuit diagram, a logic diagram, and finally, a chip layout. In almost all areas of design automation one finds the need for multiple representation. With the data base management system that can comprehend multiple representations, we can interface to expert systems to essentially perform operations on those multiple representations without transformation. Another example might be a real estate application where one deals with some specifications, floor plans, maps, and some graphics. One might be able to specify a set of constraints and walk through that multiple representation data base to extract just the right house that has the right driving distance and the right monthly cost and the right number of bedrooms, etc. One more example might be medical records. Medical records consist of x-rays, laboratory reports, perhaps electrocardiograms, and some patient forms. An object-oriented data base will enable a doctor to essentially reason about these various objects in his data base. He can ask questions about the electrocardiogram and reference that to previous patient records.

We think that the two major technologies that will drive the billion dollar silver bullets of the late 80s and beyond are VLSI and artificial intelligence. TI is a world class competitor in these technologies today and the bottom line for all of us in TI, is that we intend to keep it that way.

Ω

Opportunities for Information Systems

– Instalment 5

SOUND TO SPELLING – II

Edmund C. Berkeley, Editor

Suppose we have a machine as good as the human ear which will recognize the English sound (consonant, phoneme) "f" as in "fat" or "muffin" or "leaf" or "phrase" or "graph" or "toughen". What is the technique to be used for us to spell that sound?

In the state of the art at present this difficulty has not been surmounted. Instead, all the words (or vocabulary) that a person is to use in a given situation are spoken (each 3 or 4 times) separately into the microphone of the recognizing machine. This machine analyzes the separate recorded sound patterns, compares them, "decides" upon a pattern, and pulls out from a file the properly spelled word. I tried this just recently on a nearby speech recognizing machine, for some 20 words, one of them being "communications". I was impressed. I was told that the machine could distinguish over one thousand words currently, and many more soon. But each user of the machine had to "train" it to his own speech habits.

But of course this does not yet come near to the skill of a good human speller. Furthermore he can spell two words that sound just alike in two different ways depending on their meaning, such as "phrase" in the context of words or grammar and "frays" in the context of battles and struggle. A good speller can handle many other problems, as in the varying pronunciations of "telegraph, telegraphic, telegraphy" where the second "e" is pronounced as a schwa (an unstressed neutral vowel) in the first two words, and "e" as in "leg" in the third word.

One algorithm which might be used is a table with perhaps over 3,000 input entries of spoken words and word-elements (sequences of phonemes) including affixes that might be uttered by a speaker of standard English. There would be occasional additional entries of subject matter, meaning, etc. The output of the table would give the spelling of the word or word-element. As this table is constructed, it could be much shortened by including many simple spelling rules, like the change of "take" to "taking", "like" to "liking", "poke" to "poking", . . .

The substitution of "spelling pronunciations" in some cases, like changing "Wenzdy" to "Wednesday", would be needed.

The techniques used by a good human speller, the kind of person that used to win old fashioned spelling bees, need studying. I have known persons who could look at a wall and see in their mind's eye just how a long or apparently baffling word was spelled, and spell it with no hesitation, like "antidisestablishmentarianism" or "paradimethylaminoazoorthocarboxylic".

If the machine is to become an "automatic stenographer" and to produce usable typed English, then the person dictating must take on additional chores. These include "training" the machine on unusual terms or acronyms that may occur in the dictation, specifying punctuation and paragraphing, and arranging a way in which spoken words of unanticipated instruction will not be transcribed but will be used instead to give directions to the machine. The machine is simple in theory but difficult in reality; small models are easy, but powerful models are in the future. Ω

Games and Puzzles for Nimble Minds and Computers

Neil Macdonald
Assistant Editor

NUMBLE

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, expressed in numerical digits, is to be translated using the same key, and possibly puns or other simple tricks.

NUMBLE 8609

```
          H A B I T
       *      H A S
      -------------
        R N Y O A R
      O H Y A R B
    O N A I T O
    ---------------
  = O R N S N I R R
```

44052 10635 27

MAXIMDIDGE

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, plus a few more signs. The spaces between words are kept. Puns or other simple tricks (like KS for X) may be used.

(please turn to page 16)