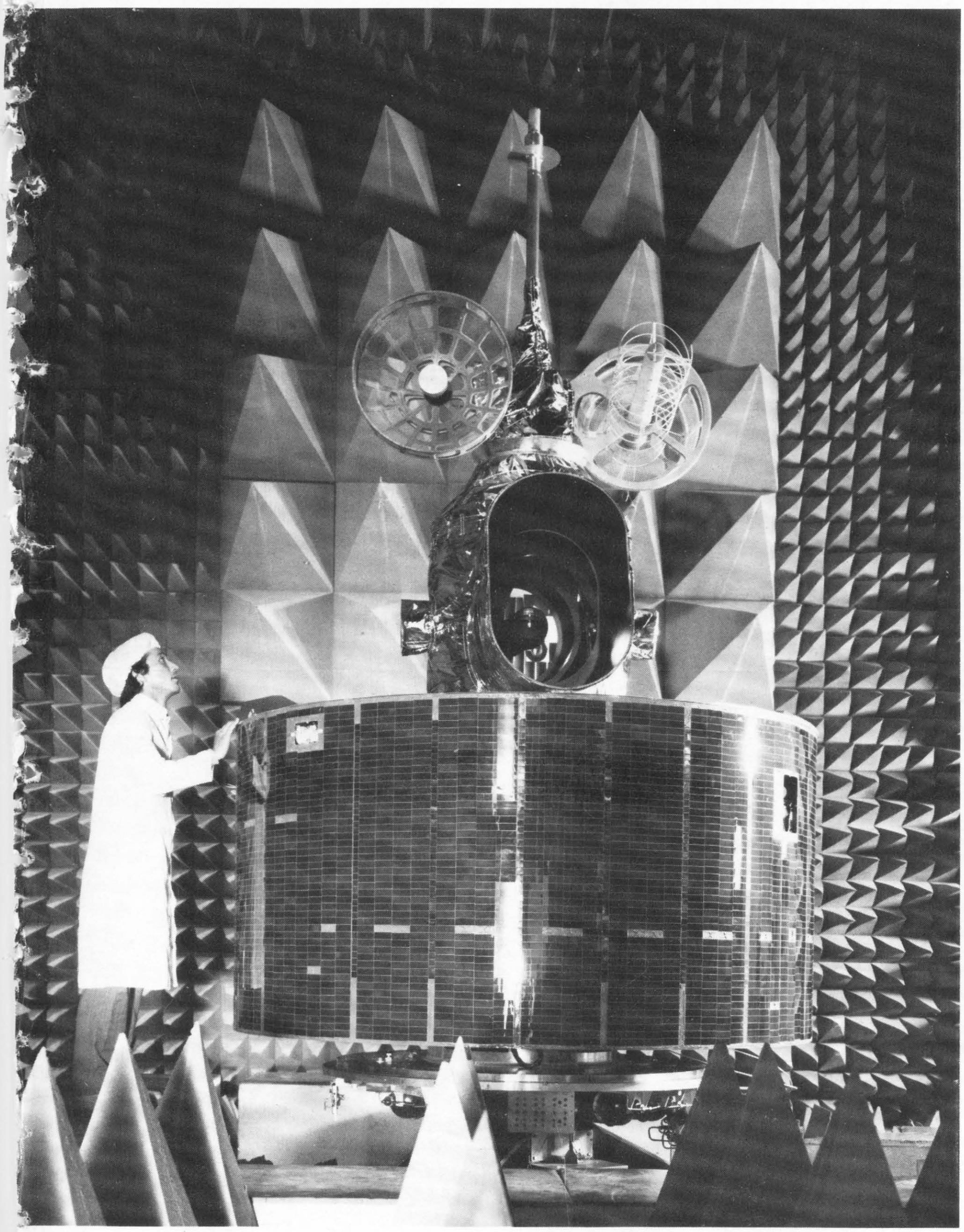
computers and people

Sept.—Oct., 1981
Vol. 30, Nos. 9-10
formerly *Computers and Automation*



U.S. BUILT WEATHER SATELLITE LAUNCHED BY JAPAN

The Computer Almanac and Computer Book of Lists — Instalment 21

Neil Macdonald
Assistant Editor

229 CEMENT-WORDS OF MATHEMATICS (List 810901)

1. Place, Position a. Locations: at, before, ahead of, behind, beyond, over, above, under, below, underneath, in, inside of, out of, outside of, on, off, against, touching, around, near, beside, next, opposite to, between, among

1. Place, Position b. Regions: top, bottom, left, right, front, back, center, side, edge, corner, end, middle, neighborhood, surrounding

1. Place Position c. Directions: to, toward, from, up, down, along, across, through, into, out of, onto, forward, backward, sideways

2. Shape, Form, Structure: flat, plane, round, circle, point, dot, line, curve, angle, fork, bend, square, space, spacing, ball, globe, ring, loop, oval, coil, hole, bump, bulge, knob, stick up, project, cover, crack, fold, knot, hanging, hollow, block, lump, chunk, solid, surface

3. Size, Magnitude: big, great, large, little, small, size, long, high, tall, deep, far, thick, fat, full, wide, short, low, shallow, near, thin, empty, narrow, length, height, depth, distance

4. Comparison, Degree: more, most, less, least, equal, unequal, enough, -er (as in "taller"), -est (as in "tallest"), great, greatest, than, as much as, as little as, not so much as

5. Indefinite Numbers and Measurements: few, several, little, much, many, some, a handful of, a lot of, a number of, a quantity of, a great deal of, -s (as in "cats", meaning more than one); also other endings for the plural, not very, fairly, rather, somewhat, partly, almost, nearly, very, entirely, altogether, quite

6. Definite Numbers: one, a, an, two, bi-, three, tri-, thir-, four, for-, five, fif-, six, seven, eight, nine, ten, -teen, -ty, eleven, twelve, dozen, twenty, score, hundred, thousand, billion, zero, no, none, naught, once, twice, half, third, quarter, plus, and (not all meanings), times, minus, less, -th (as in "sixth"), divided by

7. Order: first, second, third, next, last, -th (as in "sixth", meaning sixth in sequence), arrange, pattern, map, collect, scatter, one after another, in order, in sequence, one by one, two by two, even, uneven, together, scattered, hit or miss, spaced, diagram

8. Variation and Approximation: by and large, on the average, about, roughly, approximately, range over, depend on, vary with, various * - see page 24

172 CEMENT-WORDS OF LOGIC (List 810902)

1. Reports on Statements, Truth, Values: yes, true, right, no, false, not so, wrong, truth, falsehood

2. Connectives of Statements or Terms: and, or, not, if...then, if and only if, only if, or else, and/or, except, unless, with, without, in other words, that is, namely, provided, suppose, let, assuming, also, together with, neither...nor, nor, therefore

3. Name, Meaning: name, label, meaning, idea, be called, be named, stand for, mean, definition

4. Assertion: be, is, are, am, have, has, do, exist, happen, occur

5. Properties, Classes, Abstractions: kind, sort, type, class, example, instance, member, -ment (as in "employment"), -th (as in "width"), -dom (as in "freedom"), -ness (as in "sweetness"), -ty (as in "virility")

6. Relation Connectives: of, to, for, by, with, from, in, as to, in regard to, about, covering, in relation to, -ee (as in "employee"), -er (as in "employer"), -or (as in "visitor")

7. Variables: I, me, you, he, she, it, they, them, we, us, such, as, which, who, this, these, that, those, do (as in "Do you...?", "I do."), so (as in "I'll do so."), to (as in "I'd like to."), thing, body, person, one, A,B,C...(as in "A buys a car, and sells it to B"), the former, the latter, John Doe, Richard Roe...

8. Operators on Variables: all, each, every, both, either, any, not all, some, a, an, at least one, one, none, no, not any, neither, the

9. The Relation of Equal or Unequal: same as, equal to, identical with, different from, unequal to, other than

10. The Relation of Like or Unlike: like, unlike, similar to, dissimilar to

11. The Relation of Membership or Inclusion: in, part of, piece of, bit of, subdivision of, included in, excluded from, outside of

12. The Rest or Remainder: and so forth, and so on, etc., and the rest

13. Miscellaneous Properties and Relations: conflicting with, contradiction, consistent with, inconsistent with, overlapping with, correlated with, corresponding to, matching with, pairing with, gaps, loopholes, incomplete, complete, symmetric with, assymetric to, essence of, gist of, core of, kernel of, essentially, substantially, basically * - see page 24

16 COMPUTER STORES IN THE SAN FRANCISCO, CALIF. AREA (List 810903)

A.I.D.S. Inc. / 301 Balboa St., San Francisco, CA / 94118 / (415) 221-8500
Artificial Intelligence Designs Specialists, Inc.; business, medical, education; Apple

Bay Area Computer Data / 1258 Mission (nr. 9th), San Francisco, CA / 94103 / (415) 864-2283
Small business; "discount computers"

Better Business Machines, Inc. / 55 New Montgomery, San Francisco, CA / 94105 / (415) 982-1414
Hewlett-Packard personal computers

California Media Electronics Inc. / 2240 Morello Ave., San Francisco, CA / ? / (415) 671-9050
"Single and multi-user word processing and accounting...custom programming"

Computer Connection / 214 California St., San Francisco, CA / 94111 / (415) 781-0200
"Computer programmers welcome to use equipment without cost"; business, personal, school, educational; training; word processing; Apple

Computerland of San Francisco / 117 Fremont (financial district), San Francisco, CA / 94105 / (415) 546-1592
Business, home; hardware and software; word processing systems; supplies and accessories; Apple, Hewlett-Packard

Computerland of the Castro / 2272 Market (Castro District), San Francisco, CA / 94114 / (415) 864-8080
Business, home; hardware and software; word processing systems; supplies and accessories; Apple, Hewlett-Packard

Data Machines / 1228 Folsom, San Francisco, CA / 94103 / (415) 864-7203
Small business; software, hardware; service; leasing; rentals; microcomputers

Micro Plus / 1400 Taraval St., San Francisco, CA / 94116 / (415) 665-4990
Business, personal; legal, medical, dental; RCS 8000

Mr. Calculator Inc. / 55 3rd Ave. (at Market), San Francisco, CA / 94118 / (415) 543-1541
Small business, personal; peripherals; CBM 2000, PET

National Computer Sales / 140 Valley Dr., Brisbane, CA / 94005 / (415) 956-6300
Small business computers

Queue Computer Corp. / 1044 University Ave., Berkeley, CA / 94134 / (415) 845-5300
Business; word processing

Radio Shack / 701 Clement, San Francisco, CA / 94118 / (415) 752-5099
TRS-80

Sunset Electronics / 2254 Taraval, San Francisco, CA / 94116 / (415) 665-8330
Educational, business, personal; software; supplies; Ohio Scientific

Texas Instruments Supply Co. / 100 California St., San Francisco, CA / 94111
Business, home; minicomputers; software; TI-99/4, DS-990

The Physician's Microcomputer, Inc. / 311 Balboa St., San Francisco, CA / 94118 / (415) 668-6878
Computer systems for medical offices

(Source: Michael Goldstein's notes)

19 COMPUTER STORES IN THE SAN JOSE-SANTA CLARA, CALIF. AREA (List 810904)

The Computer Emporium / 5821 Cottle Rd., San Jose, CA / 95123 / (408) 227-5414
Apple

Affordable Computer Systems / 3331 El Camino Real, Santa Clara, CA / 95051 / (408) 249-4221
Apple, Commodore, North Star, Altos

Advanced Computer Products / 542 W. Trimble Rd., San Jose, CA / 95131 / (408) 946-7010
Personal, small business computers, prepackaged software

American Standards Programming / 5458 Castle Manor Dr., San Jose, CA / 95129 / (408) 255-6170
Program troubleshooters

Computerland of Almaden / 5035 Almaden Expwy., San Jose, CA / 95118 / (408) 267-2182
Business, home, education; Apple

Computerland of the Santa Clara Valley / 2037 El Camino Real, Santa Clara, CA / 95051 / (408) 246-4500
Business, home, education; Apple

General Programming Inc. / 2000 W. Hedding St., San Jose, CA / 95128 / (408) 248-5322
Custom business systems; Mini's and up; COBOL, BASIC, RPG

International Data Equipment & Accessories Inc. / 3350 Scott Blvd., Santa Clara, CA / 95050 / (408) 988-6444
Computer supplies and furniture

Metra Instruments / 2056 Bering Dr., San Jose, CA / ? / (408) 297-8530
Computer systems for industry, school, business; Compucolor, TI, Intercolor

The Systems Store / 2520 Mission College Blvd., Santa Clara, CA / 95050 / (408) 988-1988
Small systems, software, maintenance; Cromemco

Logical Concepts Corp. / 20395 Pacifica Dr., Cupertino, CA / 95014 / (408) 446-4566
sales, service, training, rentals

The Computer Post / 781 W. Hamilton Ave., San Jose, CA / 95125 / (408) 866-6246
Home, business, education; training, system analysis, maintenance, leasing; Apple

Interim Computer Systems / 447 S. Bascom Ave., San Jose, CA / 95128 / (408) 292-3134
Complete business systems; Intertec, Atari, Diablo, NEC, Centronics

Radio Shack - A Division of Tandy Corp. / 1228B S. Bascom Ave., San Jose, CA / 95128 / (408) 297-2603
Small business, home; TRS-80

(please turn to page 24)

Vol. 30, Nos. 9-10
Sept.-Oct., 1981

Editorial Offices	**Berkeley Enterprises, Inc.** 815 Washington St. Newtonville, MA 02160 (617) 332-5453
Advertising Contact	**The Publisher** **Berkeley Enterprises, Inc.** 815 Washington St. Newtonville, MA 02160 (617) 332-5453

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

Subscription rates: United States, $16.50 for one year, $32.00 for two years. Elsewhere, add $6.00 per year.

NOTE: The above rates do not include our publication, "The Computer Directory and Buyers' Guide." To receive this, please add $18.00 per year to your subscription rate in the U.S., and $22.00 elsewhere.

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

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

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

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

computers and people

formerly *Computers and Automation*

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.

Front Cover Picture

The front cover shows a "weather eye" called GMS-2, Japan's newest weather satellite. The satellite undergoes tests in an anechoic chamber in a Hughes Aircraft Company facility in El Segundo, Calif. The satellite was launched in Japan in August from the National Space Development quarters at the Tanegashima, Japan, space complex. It is to deliver weather pictures and other data covering 65 million square miles of the western Pacific region.

Key

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

Notice

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

Notice to Our *D Subscribers

The 1979-80 *Computer Directory and Buyers' Guide* is expected to be printed by mid-1981. In the meantime, the 1978-79 *Directory* may be consulted. Copies are available.

Jabberwocky and Technology

Clarence L. Barnhart
Clarence L. Barnhart, Inc.
Reference and School Books
P.O. Box 250
Bronxville, NY 10708

Thank you for sending me the General Instrument Corp. news release using the term "opto-isolator". This seems to be a fairly new electronics term, not yet recorded in standard general dictionaries or in technical ones such as the "McGraw Hill Dictionary of Scientific and Technical Terms" (1974). I did find it entered, however, in John Markus's "Electronic Dictionary" (McGraw-Hill, 1974), which defines it as follows:

"A coupling device in which a light-emitting diode, energized by the output signal, is optically coupled to a photodetector such as a light sensitive output diode, transistor, or silicon-controlled rectifier. The optical coupling through air, optical fibers, or some other form of light pipe provides essentially perfect electric isolation between input and output while transferring signals, data pulses, or control voltages. Also called optical coupler, optical isolator, (etc.)."

I quote this definition in its entirety because it is sufficiently straightforward to give me an idea of the function of an "opto-isolator", though I cannot claim to know the meaning of all the terms appearing in the definition. Yet as I read through the news release which you sent me, which was all about opto-isolators, I did not have the vaguest idea (just as you did not, I gather) what these blessed thingumbobs were -- until, of course, I stumbeld upon the definition cited above.

The trouble with this news release (and others of its kind) is not so much that it uses highly technical terms, but that it piles them one on top of the other so that even specialists are probably confused, though they may be unwilling to admit it. There is, I feel, a certain perverseness in ignoring simple grammatical sentence structure in favor of a kind of telescoped jabberwocky such as the following excerpts from the news release:

"These devices are suitable for high speed logic interfacing applications such as line receivers."

"The output of the detector is an open collector, Schottky clamped transistor with current sinking capability of 25mA."

"This feature provides a minimum common mode transient immunity of 1000V/usec. to give error-free coupling data despite large ground differences or induced noise."

Assuming that the technical terms are irreplaceable in these sentences (a large assumption, since some of them , like "line receiver", "open collector", and "transient immunity", I could not find entered or defined in any source, including the "Electronics Dictionary"), could not something be done to improve the grammaticality and clarity in them? A judicious use of the hyphen, for example, to set off attributive phrases from the nouns they modify would do wonders for comprehension.

Would not "applications of high-speed logic interfacing" be clearer than "high speed logic interfacing applications"? And why not say clearly "a Schottky clamped transistor that is an open collector and has a current-sinking capability of 25mA" instead of the convolutedly obscure "an open collector, Schottky clamped transistor with current sinking capability of 25mA"?

Not only does simple sentence structure clarify meaning and facilitate understanding, it also humanizes technology and science, whose language is all too often mechanical and impersonal, like the language of computers.

I hope we still have a long way to go before we let computers teach us how to communicate -- shall I say interface -- with one another.

But I fear that technical writing that sounds less than human, that seems to be moving farther and farther from common speech, which is the source of language, brings us a step closer to the dehumanized society of Newspeak and 1984.

Perhaps a word about this in your magazine might elicit some useful comments from your readers.

Clarence L. Barnhart

Amen.

Edmund C. Berkeley

Computer Program Patents and the First Patent for Software

S. Pal Asija
7 Woonsocket Ave.
Shelton, CT 06484
(203) 736-0774

"I claim that I am the owner of the first pure software patent (4, 270, 182) disclosed, claimed, prosecuted, rejected, and eventually issued as an algorithm."

Outline

1. Controversy
2. Various Cases
3. "A patentable invention does not become unpatentable merely because it includes a computer program"
4. No Flood of Patents
5. A New Way to Invent and Patent
6. Effort to Patent on My Own
7. Think in a New Direction
8. The Tenacity of Chester Carlson
9. Patent Office Rejection
10. Issuing the Patent
11. Forgivingness
12. The Algorithm that is Patented

Controversy

In view of the fact that a copyright merely protects the expression of an idea and a patent protects an embodiment of an idea, (and neither protects the idea itself), a copyright is suitable for protecting a specific code only and a patent is necessary to protect the underlying algorithm.

The controversy about software patents first surfaced before the Court of Customs and Patent Appeals in 1968 with "In re Prater" (159 USPQ 583 & 162 USPQ 541) dealing with reduction of data from spectral analysis. On rehearing the apparatus claims (means for plus function language) were again allowed but process (method) claims were disallowed as reading on "a mental process augmented by pencil and paper markings" (as opposed to strictly computer process/program).

Various Cases

The United States Supreme Court in 1972 decided the Benson case (from Bell Labs) in which a patent for converting BCD (Binary coded decimal) to pure binary utilizing reentrant shift register was denied. The court concluded "The mathematical formula involved here has no substantial practical application except in connection with a digital computer." However the Supreme Court carefully left a ray of hope for advocates of computer program patents by saying, "It is said that the decision precludes a patent for any program servicing a computer. We do not so hold." (Gottschalk v Benson 175 USPQ 673)

In 1976 the Supreme Court heard the Johnston case in which a patent for bank statements showing sub-totals of like types of expenses of a customer based on an additional digit entered on the check by the customer was denied on the ground of Obviousness. Thus this case did not advance the law of computer programs as statutory subject matter. (Dann v Johnston 189 USPQ 257)

The Flook case was decided in 1978 in which a patent for a formula to cause alarm was denied; because the court felt that the nominal "post solution activity" merely requiring the generation of a number to set alarm limit did not transform unstatutory subject matter to statutory. (Parker v Flook 198 USPQ 193)

Most recently the U.S. Supreme Court decided the Diehr case in which a computerized process patent for curing synthetic rubber was granted. The court held that a patentable process does not become unpatentable subject matter when serviced by a digital computer. The court also admonished the Commissioner of Patents and Trademarks not to dissect a process into old and new elements for determination of statutory subject matter, which should be independent of the novelty and obviousness determinations. In other words, the determination of whether or not an invention comprises statutory subject matter should be made by viewing the invention as a whole and not by subject matter of the point of novelty only.

The Bradley case, also decided by the Supreme Court the following week (March 9, 1981), was tied 4 to 4. According to the Supreme Court rules, a majority is needed to overturn. Thus, a patent was granted in this case, also. However, it is reasonable to assume that had the chief justice participated in the opinion, he would have voted the same way as in Diehr case, which would have also given the same result.

"A patentable invention does not become unpatentable merely because it includes a computer program"

The Diehr and Bradley cases do not stand for the proposition that computer programs are patentable subject matter, but only that a patentable invention does not become unpatentable merely because it includes a computer program. The CCPA, in the meantime, decided numerous cases (generally in favour of the applicant) attempting diligently to interpret the wishes of the Supreme Court according to its best understanding of the law of computer software then prevailing.

I claim that I (and not Goetz or anyone else) am the owner of the first pure software patent (4,270,182) disclosed, claimed, prosecuted, rejected, and eventually issued as an algorithm. Most patentees who claim to own software patents are erroneous in their assertions for one of the following three reasons:

1. It is not pure software. It entails substantial pre and/or post solution activity on some hardware other than electronic general purpose digital computer.

2. The claims are written as machine claims, or "Means for plus function" language claims or equivalent system claims.

3. During prosecution the non-statutory subject matter issue (35 USC 101) was never raised, primarily because it was not perceived by the examiner that the applicant was trying to protect software. This can be ascertained by reviewing class/sub-class definition.

Mr. Goetz's patents, "Sorting System (3,380,029)" and "Autoflow (3,533,086)", do not have even a single pure software claim. Nor did it receive 101 rejection. It is naive to think that the Patent Office was issuing software patents and at the same time arguing before the CCPA & the Supreme Court that computer programs are not patentable.

No Flood of Patents

My patent does not mean there will be a flood of computer patents, even though the patent office has recently created a separate sub-class for computer program patents. A flood of software patents is nevertheless unlikely to occur because most computer programs do not qualify as unobvious to those of average skill in the art of computer programming. Examples of unobvious computer program patents that may be issued in the future are as follows.

1) A program to convert handwriting to OCR (Optical Character Reader) readable hard copy.

2) A program to transform voice input to hard copy.

3) A program to synthesize/simulate voice from alphanumeric information in the computer

4) Heuristic programming algorithms.

The patents for hardware and software of necessity must go hand in hand. Otherwise, all hardware patents become worthless in the marketplace because competitors can choose to transform some/most of the hardware functions into software. It would be foolish to argue that unpatentable subject matter (software) is infringing a patented product.

A New Way to Invent and Patent

The rest of this article describes why and how I invented and patented a new way to communicate with a computer in a human-like manner. It includes a summary of how the patent works, as well as its limitations and capabilities. I also defend my claim that this is the first pure software patent ever issued.

My first exposure to computers was at the University of Southampton in 1963. There I learned how to program the FERRANTI Pegasus II computer in machine code. I was facinated by this giant man-made brain. It was then the thought crossed my mind that someday, the affluence of western civilization will be measured, not by a roast in every pot, but by a computer in every home. I further reasoned that before this can become a reality, there must be an easier way to communicate with computers; preferably, a way comparable to person-to-person communication.

For some years, I could not do anything to realize my dream of inventing a way to communicate with computers in a human-like manner, but I did have several other ideas for inventions. In 1968, I went to see a patent attorney in the United States, regarding a patent for a sleep-sensing switch. At that time, I was in humble financial circumstances; and I was told that the patent would cost approximately $1200.00, over a period of three years.

Effort to Patent on My Own

I attempted to secure a patent on my own. I soon discovered that it was almost impossible for a lay person to get a patent with any worthwhile protection. Then I made up my mind to go to law school someday.

In the meantime, the attorney and I worked out an arrangement whereby he would criticize my responses to the patent office, at a price I could afford. Since inspiration favours the prepared mind, I also made it a point to read everything I could get my hands on re the topics of inventing, computing, and patenting. About this time the Court of Customs and Patent Appeals decided the Prater and Wie cases I and II, which surfaced the computer program patent controversy.

Think in a New Direction

One thing I had learned about inventing, was that often to invent, one must think in a different direction, rather than imitate. For example, Hoover invented the vacuum cleaner, not by imitating the movement of the arms and broom, but by considering all methods of achieving that function. So, it became apparent to me that, for humans to communicate with computers in a human-like manner, the computer must not imitate human thinking and communication processes. Instead, I needed to think in an entirely different direction. I decided not to investigate artificial intelligence and heuristic programming routes any further. I enrolled at the nearest law school in Cincinnati in 1970; and I commuted three evenings a week from Dayton, Ohio, where I held a full time job with NCR as an engineer. During my law school days, many times by inspiration, if not revelation, I conceived various aspects of the word-matching algorithm which later was to form the basis of my invention.

A couple of years later, the Supreme Court decided the Benson case, in which a patent for converting a binary coded decimal to pure binary using re-entrant shift register, was denied to an inventor/employee of Bell Telephone Labs. This was interpreted by legal and computer communities as denying patents for computer programs.

The Tenacity of Chester Carlson

My dream seemed shattered once again. But I recalled the tenacity of Chester Carlson (another patent attorney) in pursuing his xerography idea. I filed a disclosure for my invention in the Patent office in 1973 against the better judgment of the legal community.

After completing all requirements for a practicing patent attorney in Minnesota, in 1974 I filed a patent application for my invention with the Commissioner of Patents and Trademarks in Washington, D.C. While copyright would have been easier to obtain, I opted for patent because, by now, I understood that a copyright merely protects the expression of an idea, while a patent protects the embodiment of an idea, and neither protects the idea itself. I chose the tougher road because the patent would make it almost impossible for someone else to circumvent my algorithm, while copyright could be circumvented merely by recoding the same algorithm. For this reason, I refused to include, and argued against, the patent office requirement that a program be included in the patent application.

Patent Office Rejection

The patent office rejected my application on the grounds that the Supreme Court in Benson had stated that computer programs are not patentable subject matter. I argued to the contrary. During the prosecution, the patent office twice suspended prosecution of my application, pending the U.S. Supreme Court decisions in the related Johnson and Flook cases, in 1976 and 1978, respectively. The Flook decision was another big blow to my case. In addition, the examiner gave me final rejection twice, and withdrew the final rejection both times, but one of them only after considering my appeal brief.

I was not altogether sorry for all these difficulties and delays in the patent office. I conjectured that, like the Laser Patent, the delay might turn out to be a blessing in disguise. During this delay, several developments, such as high level languages and low cost microcomputers, made the use of my invention more likely and brought everybody that much closer to the dream of a computer in every home.

Issuing the Patent

After reading the appeal brief, the examiner became fully convinced that I was prepared to take my case all the way to the U.S. Supreme Court. So, instead of losing a case directly on the issue of computer program patenting, the Commissioner of Patents issued the patent on May 26, 1981, with the following three unique and unprecedented FIRSTS:

(1) First pure software patent ever issued.

(2) First algorithm that allows man/machine communication in a natural language.

(3) First system that forgives a user's errors.

In retrospect, the success can be attributed to much synergy between my legal and computer background, and a strong desire to help others rather than to get rich. While it may be true that God helps those who help themselves, it is even more true that God helps those who help others.

Forgivingness

The special data processing software of this patent is entitled "AUTOMATED INFORMATION INPUT STORAGE AND RETRIEVAL SYSTEM", and carries the trademark name of "SWIFTANSWER", which is an acronym for "Special Word Indexed Full Text Alpha Numeric Storage With Easy Retrieval". While most computer systems are very fussy about the rules of grammar and computer programming, my system is very forgiving to a user who makes errors in spelling, punctuation, syntax and grammar. Furthermore, this system requires no knowledge of computer programming on the part of the user.

The Algorithm that is Patented

Even after reviewing the tables of capabilities and limitations of the system, one wonders how the computer is able to respond to an imperfect question from a human being, without really understanding the question. The essence of this may be gleaned from the following simplified version of the procedure (algorithm) the computer undertakes. For more detailed information, the reader should refer to patent number 4,270,182, or the concomitant file wrapper.

1. The full text of the database in a natural language is entered and stored in an electronic digital general-purpose computer.

2. A list of alphabetized "meaningless" words is compiled and entered into the computer. Examples of meaningless words are "a, an, the, she, where, him", etc., and all punctuation. All words not in the list of meaningless words are regarded by the computer as meaningful words. The computer does not comprehend the meaning of these words, but it knows that those words have meaning for people. The list of meaningless words varies somewhat as a function of the nature of the data base and application.

Table 1

The First Pure Software Patent:
CAPABILITIES

1. Runs on any general purpose digital computer.
2. Any computer language.
3. Any natural language.
4. Questions can be spontaneous.
5. Questions with errors in grammar, punctuation, syntax and spelling are not automatically rejected.
6. Communication can be as accurate as with a human being.
7. No-pre or post solution activity or hardware; i.e., pure software patent.
8. The system will work with any state of the art input and output devices, including voice mode, when and if successfully developed.
9. Affordable.
10. Has multiple uses, as in computer-aided instruction, library research, and police investigations.
11. Reframing of questions is permitted in case of no response, or unsatisfactory response.

3. The entire text is divided into paragraphs, pages or other logical information units, and each unit is given a unique number.

4. An alphabetized synonym directory of certain words in the text is entered and stored.

5. The frequency of each meaningful word is mapped against each paragraph and arranged in alphabetical order.

6. The user inputs the question from which meaningless words are temporarily discarded. Synonyms are added for the remaining meaningful words.

Table 2

The First Pure Software Patent:
LIMITATIONS

1. The answer must exist in the database.
2. While the user need not know the computer language, the programmer who sets up the data base and the program, must.
3. The user must ask the question in the same natural language as used for the text of data.
4. The system cannot correct the question. The most likely answer is presented along with an imperfect question.
5. Interpretation of the answer is left to the user.
6. Accuracy limited to about 98%.
7. The computer is unable to comprehend the question.
8. The system must be set up in advance. The response time is a function of the size of the data base and system configuration.
9. The system is not as efficient as indexed/nomenclatured retrieval methods.
10. No artificial intelligence or heuristic programming capability is assumed or required.

7. Meaningful words of Step 6 are compared against meaningful words of Step 5. Some clever algorithms are used for word matching, including left and right shift, proportional matching, synonym matching, stem matching, etc.

8. The paragraphs are arranged and presented according to an algorithm comprising a weighted measure of the number and frequency of meaningful words and their synonyms, coming from the question appearing in the paragraphs thus cited.

9. The computer presents the question and the paragraphs containing the answer in the order of Step 8. The computer leaves the interpretation of the answer and/or its validity to the user.

10. In the unlikely event of no answer, or unsatisfactory answer, the computer requests the user to reframe the question in different words.

(please turn to page 15)

Raster Graphics: The Dominant Visual Image of the 1980's

Conrac Division (Edmund Van Deusen and others)
Conrac Corp.
600 North Rimsdale Ave.
Covina, CA 91722

"Both technologies, computer and television, have benefited from each other's accomplishments."

Being the first chapter (and a bit of the second) in "Raster Graphics Handbook", published in 1980 by Conrac Corp., copyright ⓒ 1980, $20.00. Reprinted with permission.

Table of Contents of the Handbook

Outline of This Article

Two Separate Technologies

Two separate technologies are converging to make "raster graphics" the dominant visual image of the 1980's--in business, science, industry, education, and the home.

Alphanumeric raster-scan CRT displays already form the principal communication link between computer users and their hardware/software systems. As computer installations expand into the millions, worldwide, there is an accelerating trend toward the use of monochrome and full-color graphic images to enhance the information transfer at this vital "video" interface.

Accelerating changes are also occurring in the television industry. Computer-generated captions, graphics, and special effects have already transformed the appearance of newscasts and commercials. Computer processing is now altering the form of the television signal itself. TV cameras and receivers are analog devices, but the quality of the raster display can be significantly improved (or restored) by digital techniques. Computer-controlled, "all-digital" broadcast studios and teleproduction facilities will soon be a practical reality.

Synergistic Interchange

Both technologies, computer and television, have benefited from each other's accomplishments. The basic display device for computer-generated raster graphics is the CRT monitor--a non-broadcast version of the standard TV receiver. Decades of mass production experience have refined the receiver CRT and raster-scan circuits to a high level of reliability and cost effectiveness. Adaptation to computer graphics has required only incremental improvements. (In the case of lower-performance "alphagraphic" applications, the CRT package has been actually downgraded to provide further savings in material and assembly costs.)

Until the past few years, however, the full potential of raster graphics has been beyond the reach of computer-system designers. All of the major benefits offered by the raster-scan technique--photographic detail, cinematic animation, an unlimited range of colors--require the support of systems with large-scale, random-access memories and, in the case of animation, high-speed computational capabilities devoted exclusively to the display task. Only a few specialized applications, such as medical instrumentation and flight simulation, could justify the hardware investment.

Technical Competition and Successes

Technical competition within the computer industry has released graphics-system designers from these restrictions. Large-scale integration has dropped the price of digital memory by a factor of a hundred to one. Hardware function generators are available to perform the highly repetitive tasks which characterize most display programs. Control and timing functions can be accomplished by versatile, low-cost microprocessors. The result, predictably, has been a surge of development interest in raster graphics, helping to make computer graphics the fastest growing segment within an expanding computer industry.

Television system designers have been equally quick to take advantage of the technical advances in computer hardware. Earlier digital processing had been performed "on the fly," with only a few lines of raster data stored in memory. The new, lower-cost memory modules now allow complete frames to be captured--economically--and manipulated under computer control. Weather maps have become animated, teachers are able to add their own notations to instructional materials shown on closed-circuit monitors, and advertising agencies can rearrange the visual elements within a commercial until exactly the desired effect has been achieved.

Most important, the gap between television and computer-based graphics hardware has become almost non-existent. Interfacing can be accomplished with off-the-shelf conversion modules. This means that all of television's new digital (and analog) "black boxes" can be readily adapted to computer graphics, expanding its potential in ways which are just being explored.

The Role of Conrac Corporation

A major purpose of the Conrac Raster Graphics Handbook is to facilitate this continuing, synergistic interchange between the television and computer technologies. The Conrac Division of Conrac Corporation has, in fact, served as a catalyst and technical leader in both fields for a number of years. Conrac supplied graphics-display units for one of the earliest computer-controlled utility systems. For over a quarter of a century Conrac has also been the country's leading supplier of high-performance television monitors. An estimated three-fourths of all broadcast and teleproduction studio monitors are Conrac units, setting the industry's standards for color fidelity and picture quality.

Raster graphics combines the company's two historical strengths. Few suppliers of raster graphics equipment can match Conrac's breadth of experience. No other company supplies such a variety and range of monitors for graphics display. Yet the monitor is only part of a total graphics system. There will be, moreover, a number of occasions when Conrac quality and performance are not essential to the application. The scope of this Handbook reflects these facts. The text draws on the knowledge and background of the Conrac engineering staff as these resources apply to the entire field of raster graphics, independent of Conrac's specific products.

The principal emphasis is on the design of computer-based systems. Where applicable, television design concepts, equipment, and standards are incorporated into the discussion. When possible, a unifying terminology has been used, drawing from both disciplines. The authors have also felt free to include ideas and equipment descriptions from other fields which seem, on the surface, far removed from raster graphics. Included in this category would be circuit concepts and hardware developed for video games, cable-television networks, and facsimile systems.

A Broad Overview

The Handbook presumes a familiarity with computer technology, and is addressed to individuals who are actively engaged in the design of raster graphics systems. The goal is to present the broadest possible overview of the current state-of-the-art in this rapidly evolving technology and, at the same time, point the way toward developments which will influence the design of graphics systems in the foreseeable future.

The text is also organized so that it can serve as a valuable checklist for readers who are evaluating the capabilities of systems now being offered. The entry cost into raster graphics can range from a few hundred to over a half-million dollars. Clearly, there are economic tradeoffs which may or may not limit the future potential of the selected system. There are, too, a variety of performance features which may have little value technically, but are necessary in terms of subjective "appeal." Technology aside, the human factor remains an important variable, perhaps the most important variable, in every raster-graphics equation.

Raster Graphics – A New Definition

The emergence of a higher-performance, more cost-effective raster-scan CRT display has required a redefinition of terms. "Computer graphics" had its start with X-Y plotters. The technology was soon extended to include vector-graphic (line-drawing) CRT systems based on refresh-vector hardware and, at a later date, random-scan storage tubes. Only the endpoint coordinates of the lines needed to be stored, so memory requirements were held to a minimum. Refresh-vector writing speeds made it possible to "animate" the display and to create interactive systems which allowed the operator to control the display in "real time" through such input devices as lightpens, joysticks, and digitizer tablets. Again, however, only the endpoints needed to be recalculated with each refresh cycle, minimizing the need for high-speed computational capabilities.

Pixels

Meanwhile, a separate "image processing" technology had been following its own evolutionary path. Instead of dealing with lines and points, randomly positioned anywhere on the display surface, image processing was based on a rectangular array of picture elements or "pixels." Digital information defining the state of each pixel was stored in a random-access (or rotating) memory and used to generate a television-type, raster-scan CRT display--in monochrome or full color.

The source of the pixel data might be a computer, as in the case of a CAT (computer-aided tomography) scanning system. Or the source data could be the digitized output of a TV camera mounted on a satellite, or of a spot scanner in a photo-analysis laboratory. In every case, new information or insights could be gained by using computer techniques to alter the contrast or color, or to perform more sophisticated procedures such as correlations, convolutions, and digital filtering of the pixel data. The image processing systems required large amounts of memory, enough to store up to several bytes of data for each of hundreds of thousands of pixels. But the displays were relatively static and, considering the nature of the applications, there was little pressure for high-speed alterations in the displayed images. A delay of seconds, or even minutes, was entirely acceptable.

The division between vectorgraphic "computer graphics" and raster-scan "image processing" has now been bridged by the new raster display technology. Low-cost memories have made it economically feasible to assemble high-resolution, fine-detail raster systems that all but eliminate the objectionable stairstepping of vectorgraphic lines when displayed on a raster-CRT screen. Microprocessor-based system architectures have also made it possible to alter selected display elements at full animation rates. The result is "raster graphics"--combining the full-color, pixel-by-pixel control potentials of image processing and the line-drawing capabilities of vectorgraphics in a single display system.

Every Type of Signal Generation

Raster graphics encompasses, in effect, virtually every type of computer-generated or computer-processed display: alphagraphic, vectorgraphic, and continuous-tone imagery. The only type of raster-scan display excepted from this definition would be purely alphanumeric systems with fixed character spaces and a limited choice of character fonts.

A specific raster graphics system may be optimized for imaging or vector-oriented graphics. Often as not, however, both capabilities are incorporated into the system design. Line-drawing software, for example, may be employed to generate a pixel-data image of an object in the system's display memory. The displayed color or intensity of the pixels may then be defined--or redefined--by image-processing hardware, such as programmable lookup tables. Or conversely, a system designed primarily for image processing may use a graphics "overlay" to add captions and/or to "key" the processing procedures to specific display areas on the CRT screen. Figure 1-1 illustrates this cascading and combining of graphics and image-processing functions.

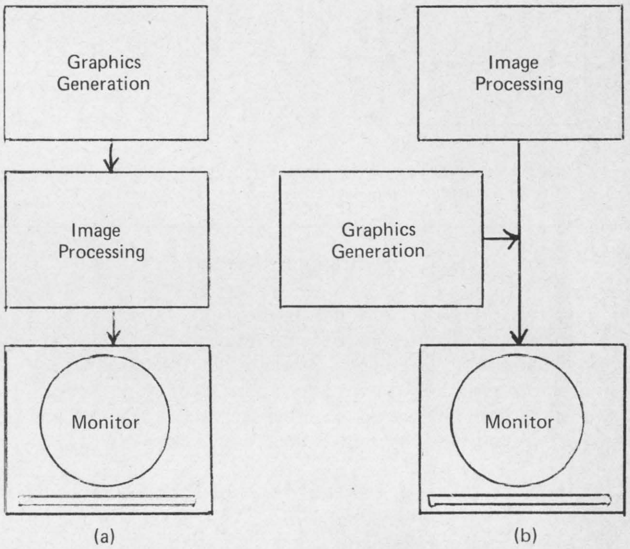


Figure 1-1: Multiple graphics and imaging functions – performed sequentially (a) or combined on monitor screen (b).

Sophisticated Techniques

Cascading and combining can also be applied to multiple display sources--both analog and digital. Television system designers have, in fact, developed a variety of sophisticated techniques for this purpose. The output of the graphics generator shown in Figure 1-2 could be superimposed on a "live" or recorded video picture, for example, or video images may be inserted into selected areas, such as outlined letters, within a graphics display. (The term "video" will be limited in this Handbook to the raster-scan output of a video camera, disk, tape recorder, or similar analog source.)

The same principles, in many cases the same equipment, can be applied to computer-based raster graphics systems to increase their scope and versatility. Business charts from previous financial periods could be recorded, for example, on video tape and combined on the monitor screen with current-period displays generated by the graphics system. Frequently used graphic overlays and formats could be added in analog form at the monitor input rather than digitally regenerated in display memory each time they are needed. Background patterns or images could also serve as a visual analog of background music--relieving tedium and helping to maintain the operator's attention.

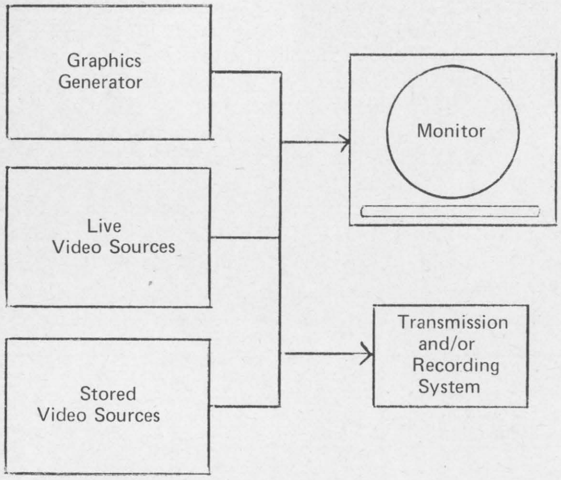


Figure 1-2: Television switching/mixing system with multiple display-signal sources.

The Graphics System

There is not, and may never be, a strict definition of the functions and hardware that constitute a "graphics system." Too much depends on the relative importance of the graphics-generation process, compared to the overall purposes served by the computer installation.

At one extreme is the "standalone" graphics system shown in Figure 1-3. Nearly all of the system's computing capabilities and most of its memory resources are dedicated, we can assume, to the display task. A number of commercially available systems oriented to a specific application, such as image processing or computer-aided design, fall in this category. A more general purpose standalone graphics system would be limited, in most cases, to the processing of "display lists" of coordinated data generated either manually or by a separate computer system.

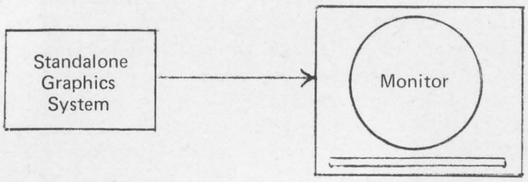


Figure 1-3: Standalone graphics systems with dedicated mini-computer or microprocessor.

At the other extreme is the configuration illustrated by Figure 1-4. Nearly all of the graphics-generation functions must be performed, in this case, by a host computer's hardware and software. The "graphics system" is reduced to a monitor and the display-generation elements of the host interface. The graphics-output section of a process-controlled computer could be implemented in this form, for example, if the graphics are limited to relatively simple flow charts with alphanumeric notations indicating the state of process variables.

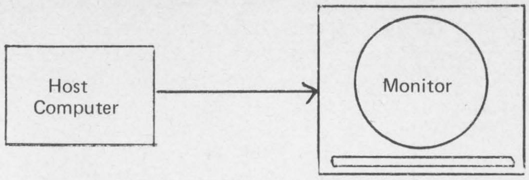


Figure 1-4: Direct interface between host computer and monitor.

The "midway" Systems

The majority of raster graphics systems fall midway between these two extremes. With the exception of the special-purpose installations noted above, standalone graphics systems rarely have the computing power to perform all of the extensive "modelling" and "viewing" functions required to convert source data to graphics form. The generation of a raster graphics display can, at the same time, impose a severe burden on a host computer's processor and memory resources. The answer has been to "off-load" the host computer by transferring a part or all of the repetitive graphics functions to a separate graphics subsystem with its own processor hardware and graphics software.

One option is to add a graphics capability to the functions of a microprocessor-based intelligent CRT terminal, as illustrated in Figure 1-5. The resulting "graphics terminal" is generally limited to alphagraphic or, at most, vectorgraphic displays. This is more than adequate, however, for a number of applications, such as the generation of business, process control, or educational graphics. Interaction with the system may be through the standard terminal keyboard or through such added accessories as lightpens and digitizer tablets.

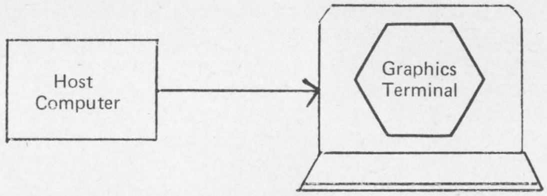


Figure 1-5: Intelligent terminal with "graphics" capabilities.

Graphics Controller

Another approach (Figure 1-6) is to concentrate display-generation functions into a "graphics controller"--physically separated from both the host computer and the CRT monitor. The monitor may be, in fact, only one of several display or recording devices controlled by the graphics subsystem. The host computer may also be only one of several data sources for the graphics controller. Display information may be supplied, for example, by mass storage devices or data communication links connected directly to the controller. By definition, however, the host computer retains complete control over the graphics-generation process, including the operation of the graphics controller and its accessories.

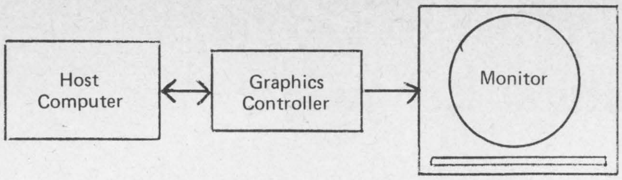


Figure 1-6: Programmable interface between host computer and monitor.

We will be using the graphics-controller configuration as our descriptive model in the following chapters. But the raster graphics "system" will not be limited to the graphics controller, or even to the controller-monitor package which characterizes a number of commercially available systems. Instead, the system definition will encompass all of the functional resources, including those of the host computer, which contribute to the generation of a raster graphics display. Graphics system software could include, therefore, program modules executed by the host computer, by the graphics controller, and by programmable hardware in the display monitor itself (Figure 1-7). The same would be true of accessories and peripherals which may be physically connected to the host computer, graphics controller, or the monitor input (Figure 1-8). As long as they contribute to the graphics-generation function, they will be considered a part of the graphics system.

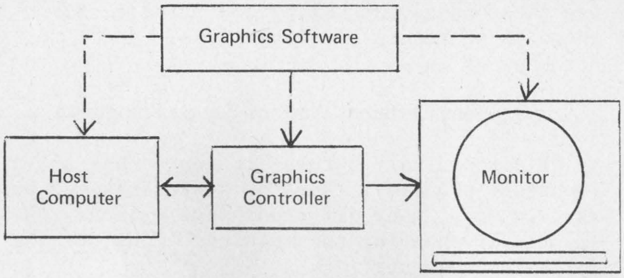


Figure 1-7: Distribution of graphics-system software functions.

Wide Range of Options

From all of this it can be seen that the purpose of this Handbook is not to define an optimum raster graphics system, but to emphasize the wide range of options and alternatives available to the systems designer--starting, in the next chapter, with the basic question of whether raster graphics is, in itself, the best solution to the specific display problem.

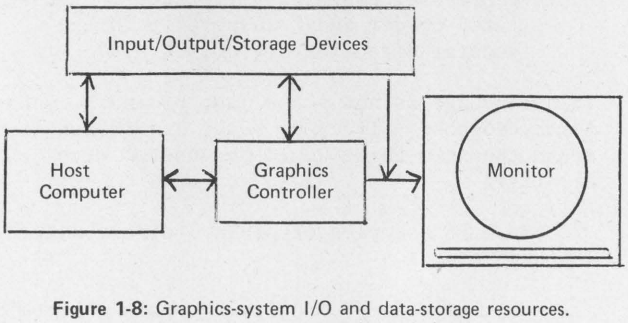


Figure 1-8: Graphics-system I/O and data-storage resources.

Two Important Decisions

Two questions must be considered before any decision is made regarding the design or purchase of a raster-scan CRT system. First, is a CRT the most appropriate display device? Second, and equally important, is the raster-scan technique the best way to present information on the CRT screen?

The CRT has been creating luminous images for approximately a hundred years. It was, in fact, the first electronic vacuum tube, predating the term "electronics" by a half-century. During the greater part of this period, particularly during the past thirty years, there has been an intensive search for an electronic-display device that could replace the CRT--or more precisely, a device that would overcome the major shortcomings of the CRT in its present form. The CRT is bulky, with a very high ratio of depth to display area, is relatively fragile, poses high-voltage and radiation hazards, and is among the least efficient of all display methods in terms of energy consumption.

Despite these faults, the CRT has maintained its position as one of the most versatile, cost-effective devices available to the display-system designer. Considerably more than half of all the electronic displays now in service (including television receivers) are CRT's. Ω

Asija – *Continued from page 10*

Occasionally, there have been claims from patentees who thought they were owners of computer program patents. But the introductory discussion traces the defects in their claims, and supports my claim that I (and not any one else), am the recipient of the first pure software (no pre-solution or post-solution activity, and no hardware other than the general-purpose computer) patent disclosed, with claim prosecuted, rejected, and eventually issued as such. Ω

Thorsen – *Continued from page 19*

Table 3

COMPOSITE OF TWO FILES:
STK.INV.FIL.DES and STK.INV.FIL.PRIC

Stock No.	Stock Descr.	Unit.Price
PL21	plug	.70
SO68	socket	1.20
BU42	bulb	.40
BA33	battery	1.50

Ω

Telling a Computer What to Do in Plain Ordinary Natural Language

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

" 'Plain ordinary natural language' is a way of saying something so that almost everybody would understand it . . . would think it was clear, direct and simple."

Outline

1. My Main Thesis
2. What is "Plain Ordinary Natural Language"?
3. Many Different Ways of Saying Something
4. What is a Program?
5. What is a Computer Program?
6. Telling a Computer Anything -- Obstacle 1: Language
7. Obstacle 2: How to Think about Programming
8. Subdivisions of a Computer Program
9. Explanations of the Subcontexts
10. Some Examples of Natural Language in a Computer

1. My Main Thesis

You can tell a computer what to do in plain ordinary natural language.

In other words, it is not necessary (though it may of course be helpful, convenient and efficient) to learn and use a strange rigorous computer programming language (such as FORTRAN, COBOL, or BASIC, or any one of 500 others).

This assertion is not on the face of it evident. Nor is it accepted as a true statement by a great many persons in the computer field, who are likely to call it absurd.

But the proposition or assertion can be demonstrated. There are examples of programming using ordinary natural language. And there is also reasoning that leads to this proposition, based on definitions, logic, and facts.

2. What is "Plain Ordinary Natural Language"?

"Plain ordinary natural language" is a way of saying something so that almost everybody would understand it, and would think that the way it was said was clear, direct, and simple.

In other words, the phrasing is not:
verbose
affected
highbrow
perplexing
strange
devious, etc.,
and does not have other properties that conflict with easy understanding.

There is an obvious exception to this statement: certain names. You have to use the names that things are known by. In the case of motor cars, some of the names you have to use are clutch, transmission, carburetor. You have to use these names, no matter how strange, if there is no reasonable other name. And I do not know of any.

3. Many Different Ways of Saying Something

"Plain ordinary natural language" has an important property: ideas or statements may be said in many, many different ways without substantially changing the meaning. Consider the following requests:

> Pass the sugar, please.
> Please pass the sugar.
> Can I have the sugar?
> May I have the sugar?
> Sugar, please.

Almost everyone would understand all of these statements to mean the same request, and would probably respond by passing the sugar.

In contrast, consider the following:

> Would you be so gracious as to elevate the vessel containing the sweet, granulated powder and transport it in the general direction of yours truly?

This language is not plain, not ordinary, unnatural, verbose, affected, ... , and though it means the same thing would be greeted with hilarity.

Some more examples of plain ordinary natural language are:

> Multiply item 3 and item 4; this is item 5.
> Item 5 is item 3 times item 4.

Find the product of item 3 and item 4;
call it item 5.

In contrast:

> Now, if you take the item known as 3, and then determine the product of it and the figure designated as the 4th one, you will obtain the value identified by the label 5.

Having clarified the concept of plain ordinary natural language, let us now seek to clarify the concept of "a program."

4. What is a Program?

A program is a way to do something. A program is a set of instructions or actions in sequence. A program is a procedure.

Some examples of programs are:
- how to go shopping in a store for what you need
- how to cook an egg that is to be soft boiled
- how to play a piece of music
- how to knit a sweater
- how to make a good decision
- how to study geology

Some programs are so well learned in the ordinary course of human life that it is hard to think of the exact steps and their sequence, because long ago they have sunk into habits. Examples: how to tie your shoelaces; walking. Some programs are so novel that you have to learn step by step just what is to be done. Examples of this kind are: how to balance your checkbook; how to order stock for a retail store; how to get information easily and quickly out of large files.

We shall here deal with a simple collection of the steps that are needed to make a computer do what you want it to -- in much the same way as you learn the steps that are needed to make a motor car do what you want it to do.

5. What is a Computer Program?

A computer program is a sequence of operations that a computer can do, organized to accomplish what you want.

In the 1930's a computer was a human clerk who computed, and the sequence of operations was expressed on a piece of paper (called a "worksheet") where columns (called (1),(2),(3), (4), ... etc.) were headed by:
- the number of the column
- the name of the data that was in it
- the formula which told the computing clerk how to fill in that column

For example, (1) might hold the day of the week, (2) might hold the hours worked, (3) might hold the rate of pay for that kind of work, and (4) might hold the cost, equal to (2) multipied by (3), etc.

In the 1980's a computer is a machine that computes, and a program to make that computer do what you want it to do, if expressed in plain ordinary natural language, is:

1. Fill in (1) with day of week.
2. Put into (2) the hours worked.
3. Enter in (3) the rate of pay.
4. Multiply (2) by (3); this is (4).

And so on.

Now, even though computers have been in the business environment for more than 30 years, the languages in which the establishment of computer programmers expresses these simple instructions are difficult and complicated.

Also, there are difficult and complicated choices to be made in the design of sequences of operations or steps, because some ways of proceeding are far more costly than other ways. For example, if you search for a name in a telephone book, do you want to start at page 1 in the A's, compare your name Pennypacker with each successive listing, and so on and so on, until three days later you find your name Henry H. Pennypacker in the P's?

Of course, the computing clerk without even being told, would search more efficiently.

6. Telling a Computer Anything – Obstacle 1: Language

In the early days of computing machines, it was hard to program computers. The first language for programming consisted of strings of 1's and 0's, usually called machine language. These corresponded to the setting of switches, 1 for "on" and 0 for "off". The second language for programming consisted of easy strings of letters and numbers, like BUT 7887 for a telephone number. This was usually called assembly language, and enabled programs to be put together or assembled much more easily. Then came so called higher level languages, with names like FORTRAN (for "formula translation"), COBOL (for "common business oriented language"), and BASIC (for "basic" in a certain teaching situation at Dartmouth College, subsequently mnemonicized as the initial letters of "Beginners' All-purpose Symbolic Instruction Code"). Much development occurred in machines, languages, access to computers, cost reductions, etc.

With the coming of microcomputers, and a tidal wave of new computer users who want easy simple programming, the objective of "plain oridnary natural language" as a language for programming computers is within reach.

7. Obstacle 2: How to Think about Programming

When we start to use plain ordinary natural language to tell a computer to do something, we find a second obstacle: how can we think about programming a computer, without the heavy baggage of a programming language, even one as good as the language BASIC?

There are, in fact, only 13 subdivisions (or 12 or 14 or thereabouts) of programming ideas, 13 subcontexts, which we need to think about, understand, and arrange to program in "plain ordinary natural language". And these subdivisons (or subcontexts) correspond very well with the subdivsions of ordinary computing by human clerks.

8. Subdivisions of a Computer Program

Every computer program has or may have the following subdivisions (we shall call them "subcontexts", divisions of the "context" of computer programming):

1. A NAME by means of which the computer program is known, can be talked about, filed, and pulled out of file.
2. A way by which DATA or information can be inserted or input into the program.
3. A way by which data from a FILE or TABLE can be inserted into the program.
4. A way by which directions for COMPUTING, and REASONING, can be given to the program.
5. A way by which CONDITIONS can be decided.
6. A way of BRANCHING from one sequence of instructions to another sequence of instructions.
7. A way of CYCLING or LOOPING over and over again through a sequence of instructions, with minor changes, until the cycles are finished with (otherwise we have an "infinite loop" as it is called, and the program will never come to a stop).
8. A way of calling SUBROUTINES or RULES for use in various kinds of cases, and then RETURNING to the MAIN program.
9. A way of outputting results from time to time, to a TERMINAL or a PRINTER or a DISPLAY or a FILE or a TABLE.
10. A way of STOPPING.
11. A way of FILING the program so that it can be used again.
12. A way of RUNNING the program on cases and examples.
13. A way of MODIFYING the program so as to remove errors (called BUGS) and to take in changes in the requirements that the procedure is to meet.

9. Explanations of the Subcontexts

The first one of the subcontexts is the NAME of the program. In a section of computing clerks in the 1930's, the NAME might be "figuring payroll" or "recording overtime" or "reordering stock", etc. The procedure much of the time would be in the memory of the clerks, with such assertions as "this is the way we always do it" and "here is a worked example of the way we do it."

The second subcontext is the way in which DATA goes into the program. As a new order comes in the mail, or a shipment comes into the delivery room, or an article goes out on to the shipping platform, DATA is produced and is acted on.

The third subcontext is data coming from FILES or TABLES. The file may be a set of the addresses of suppliers, etc. The table may be a table of the commission percents of salesmen. Etc.

The fourth subcontext is the rules for COMPUTING and REASONING, the ways in which calculations are performed, classifications are established, etc.

The fifth subcontext is the way in which CONDITIONS can be decided. Conditions involve the ideas IF, SUPPOSE, ASSUMING, IN CASE 3, IN ALL REMAINING CASES" and so on. They involve signals, indications, on-ness, off-ness, flags, true, false, and similar ideas.

The sixth subcontext is a way of BRANCHING from one sequence of instructions to another sequence of instructions. A college treats freshmen, sophomores, juniors, and seniors by different sets of rules. A computer can likewise treat different cases by different sets of instructions.

The seventh subcontext is CYCLING or LOOPING over and over again through a sequence of instructions, with minor changes, until the cycles are finished with. This is really a common idea and common procedure, though we do not notice it very much, because it is so much a habit sunk into our experience. When we count, we do this, using the regular sequence of whole numbers and stopping when we reach the last item of a collection. When we add a column of numbers, using an adding machine, we do this, stopping when we reach the last number to be added, and then we push a button or lever or key or knob, signaling "I want the total now" and the adding machine prints the total.

The eighth subcontext is a way of calling SUBROUTINES or RULES for use in various kinds of cases, and then returning to the MAIN program. For example, there may be a RULE which provides for different sales taxes in different localities, and the subroutine will be called if the locality is any one of a set of places.

The ninth subcontext is a way of outputting results from time to time, to a TERMINAL, or a PRINTER, or a DISPLAY, or a FILE, or a TABLE. For every computer program becomes worth its effort and trouble when it delivers a solution to a problem to the place where that solution is to be used.

The tenth subcontext is a way of STOPPING, and by this we mean stopping the program in a tidy way. For it is always possible (at least in theory) to pull the plug and disconnect a computer in such a way that starting it again is very difficult. The telephone system, which is a giant computer of a special nature, is an exception: it may be very difficult to stop it. In many programming languages, the word STOP does stop the program.

The eleventh subcontext is a way of FILING the program so that it can be used again. For clearly, if you are going to the effort of working out a good program, it makes sense to file it in some reasonable way, so that you can use it again and again.

Here is one of the ways in which we lose sight of a very necessary behavior: human beings when they often do some procedure grow habits: the more you practise typing, the better you type; the more you carry through a clerical job, the more habitual it becomes. This happens without thinking about it. Nature is a clever schemer. But in the case of computers, you do have to think about keeping useful programs, and it can be disastrous when you can't find them once more.

The twelfth subcontext is a way of RUNNING the program on cases and examples. The program is a good and effective instrument. To use this instrument is to "run" it; to use a sewing machine is to run the sewing machine, to sew with it, using thread and cloth and directing the seam in the way it should go.

Often in the case of a program some of the data which it uses is changed from a starting value to a final value, and the run winds up with final values. Before the program can be run a second time, a "restarting" procedure is needed to reset the original values. Before you start out once more in a motor car, it is a good idea to inspect the tires, the gas gage, the oil, etc., so that your next "run" will not encounter trouble.

The thirteenth subcontext is a way of MODIFYING the program so as to remove errors and make changes. This technique is almost the same as writing the program in the first place, and the regular method is to call a utility program which is named the EDITOR, and this is exactly what it is. It takes in the program you ask it to take in (or a blank space where you can write a new program), and arranges things so you can "write" what you want. It regularly has two "modes" called "text" mode and "command" mode. "Text" mode is like writing in black on a memo. "Command" mode is like writing in red ink instructions for the typist.

10. Some Examples of Natural Language in a Computer

For some time, programmers have made use of ordinary natural language here and there in computer programming (subject to their inveterate habit -- like all human beings -- of abbreviating).

In BASIC, the word REMARK is a term which precedes a line of explanation in a computer program, and which has no effect on the running of the program, but does give some documentation. But the standard abbreviation is REM.

The utility program named the EDITOR is regularly called the EDITOR, or ED for short. The commands it uses are often the first letters of the words telling the command, like E for EXIT.

In COBOL, the following statement is perfectly valid (except that you have to position each character in a fussy way according to rules):

```
MULTIPLY QUANT-ORDERED BY UNIT-PRICE
    GIVING EXTENSION
```

In the system called DJINNI-5, the program reported in Table 1 (dealing with inventory and invoicing) works perfectly, as shown in Table 2. The files appear in Table 3.

You can tell a computer what to do using plain ordinary natural language -- if not quite yet, then very soon.

Table 1

The Input Program Using DJINNI-5

```
A>TYPE INV.ENG

CALL THIS PROGRAM INVOICE.BODY

STARTPGM:

INPUT THE DATA STOCK.NUMBER$

ENTER THE QUANTITY.ORDERED

LOOK UP THE STOCK.NUMBER$ IN THE STK.INV.FIL.DES
 AND FIND THE STOCK.DESCRIP.$

LOOK UP THE STOCK.NUMBER$ IN THE STK.INV.FIL.PRI
C AND DETERMINE THE\

CORRESPONDING UNIT.PRICE

FIND THE PRODUCT OF THE QUANTITY.ORDERED AND THE
 UNIT.PRICE; THIS IS THE\

EXTENSION

PRINT WITH HEADINGS STOCK.NUMBER$, QUANTITY.ORDE
RED

PRINT WITH HEADINGS STOCK.DESCRIP.$, UNIT.PRICE,
 EXTENSION

PRINT "MORE ITEMS? (Y/N)"

ENTER THE RESPONSE$

IF RESPONSE$ = "Y" GO TO STARTPGM

STOP
```

Table 2

An Output Run Using DJINNI-5

```
A>CRUN INV

CRUN VER 2.35
PROGRAM INVOICE.BODY
STOCK.NUMBER$
? S068
QUANTITY.ORDERED
? 200
STOCK.NUMBER$       QUANTITY.ORDERED
S068                 200
STOCK.DESCRIP.$     UNIT.PRICE              EXTENSION
SOCKET               1.2                     240
MORE ITEMS? (Y/N)
RESPONSE$
? Y
STOCK.NUMBER$
? BU42
QUANTITY.ORDERED
? 3
STOCK.NUMBER$       QUANTITY.ORDERED
BU42                 3
STOCK.DESCRIP.$     UNIT.PRICE              EXTENSION
BULB                 0.4                     1.2
MORE ITEMS? (Y/N)
RESPONSE$
? N
```

 (please turn to page 15)

Society, Computers, Thinking, and Actuaries

Edmund C. Berkeley
Fellow of the Society of Actuaries
President, Berkeley Enterprises, Inc.
Editor: "Computers and People", 1951 . . .
"The Computer Directory and Buyers' Guide", 1955 . . .
"The Notebook on Common Sense, Elementary and Advanced", 1971 . . .
"People and the Pursuit of Truth", 1975 . . .
"The Fork River Anthology: Poetry and Sense", 1981 . . .
Berkeley Enterprises, Inc.
815 Washington St.
Newtonville, Mass. 02160

"The century of the 1800's produced extraordinary new concepts, new recognitions: . . ."

Based on an address to the Actuarial Research Conference at the University of Manitoba, Winnipeg, Canada, August 1981.

It is a great pleasure to me to be invited here to talk to you all.

I think it may be worthwhile to begin by describing to you my background, the point of view from which I talk.

Clerk in life insurance company actuarial and methods departments, 1930 to 1948.
Taker of actuarial examinations, 1930 to 1941.
Fellow of the Actuarial Society of America and the American Institute of Actuaries, 1941.
Member of the U.S. Naval Reserve on active duty, 1942 to 1946; last rank, Lt. Comdr.
In the Computation Laboratory of Prof. Howard H. Aiken, 1945-46.
Founder of the Association for Computing Machinery, 1947, and 1st secretary 1947-53.
Left insurance company employment 1948, and worked in own business with 1 to 8 employees from 1948 to the present.
Have operated or owned (or both) minicomputers and microcomputers, 1961 to the present.
Have held contracts for investigation from the Office of Naval Research from 1964 to 1972.
Have written 14 books with total sale over 100,000; translations have been in French, Japanese, Russian, Swedish, Polish, etc.
Have lectured on computers in the United States Canada, Australia, Great Britain, the Soviet Union, and Japan.
Have not retired and never will.

My interests have been: logic; mathematics; actuarial knowledge; linguistics; cryptanalysis; common sense; wisdom; literature; computers; computer programming; chess; poetry; mineralogy; growing plants indoors, including orchids; foundations of mathematics; Boolean algebra; the writing of fiction; teaching; the improvement of the world; the promotion of peace; etc.

My work full time in the actuarial field lasted from 1930 to 1948, with that interlude in the U.S. Navy.

As a result of my long and valued apprenticeship in the actuarial profession, I learned a great deal. I admired the senior actuaries with whom I came into contact. They were kind to me, explained to me, and taught me. I learned in particular:

- a sacred ideal of telling the truth
- the value of never misrepresenting facts
- the importance of estimating figures ahead of calculation
- a code of professional honor
- the importance of being right always in actuarial matters

Proposition 1

Proposition 1: More than 50 percent of programming by human beings will vanish as computers take over computer programming.

In other words, the specification of the detailed steps that a computer is to follow will no longer be necessary as the computer becomes able to understand the meaning of such commands in English as:

- Find the sum of the column of figures X.
- Find the scattering of the triples X, Y, Z.
- Given a file of transactions T, and a main file M, update the file M.
- Suppose a mortality table with force of mortality MU changes to MU2. How does the set of functions F change to F2?

Suppose we review a little history. About 1955 a computer was mainly a number cruncher, a prodigy able to calculate a million times faster than a human being. About 1965 it was mainly a data handler, a processor of data. In the 1970's it was becoming an extraordinary word processor. In the 1980's it will become a processor of ideas. And in the 1990's it will become a processor of systems of ideas.

In the actuarial field, the computer travels the career path: actuarial department clerk; actuarial student; assistant mathematician; assistant actuary; actuary; vice president.

Why do I make this prediction? Just as there is no apparent limit of the attainments of a good human mind, like Aristotle, or Galileo or Einstein, so there are no limits of attainments of the good computer mind. Anything that a human mind can think, a computer mind can think also.

Proposition 2

Proposition 2: Every defined intellectual operation will be performed by computer faster, better, and more reliably than by a human being.

A human brain and a computer brain are very much alike. Here are some of the resemblances:

- They operate electrically
- They store information
- The stored information is organized for access
- They operate at high speed (of course, high speed for a human mind is about 10 to 100 bits per second, while high speed for a computer mind is about 10 billion bits per second)
- They operate reliably a great deal of the time
- They operate with precision a great deal of the time
- They receive information via sensing organs
- They output information via acting organs
- They perform arithmetic by successive steps
- They perform logic by successive steps
- They are machines of great complexity made up of simple parts of only a few kinds
- Both are programmed by a culture

There are of course enormous differences. Speed: a factor of a million or more. Reliability: a factor of a million or more. The culture of human beings is more than 10,000 years long; of computers, only about 30 years. The learning of a human being; more than 20 years. Learning by a computer: on the order of 1 minute.

Over the last 30 years, wherever computer and human being are properly set to solve the same defined problem, the computer is faster, better, and more reliable. This is certain to continue in the future.

Proposition 3

Proposition 3: All the language of thought will become calculable like mathematics.

Until the 1800's, informed human beings in general agreed that Euclid had said the last word on geometry; that mathematicians like Newton had said the last word on arithmetic, algebra, and calculus; that Aristotle had said the last word on logic and syllogisms; that Latin showed the way grammar should be; and that Philosophy and Metaphysics could be handled by rational arguments using words with no concrete examples.

The century of the 1800's produced extraordinary new concepts, new recognitions:

- There were more geometries than Euclid's
- There were more algebras than the algebra of ordinary numbers
- There were more logics than Aristotle's
- It ought to be possible to make a general computing machine that would do general computing

This last concept was that of Charles Babbage, a notable professor and actuary.

Now in 1981 it is clear that every branch of knowledge from Anatomy to Zoology is organized in the same simple way: a mathematical system. It has a collection of ideas or concepts or special terms. It has a collection of relations that exist between them. And if you understand the special terms, and the relations between them, then you understand that branch of knowledge -- just as if it were a mathematical system. Here is the framework by which a computer when properly programmed can deal understandingly with the system of any branch of knowledge, so that you do not have to think with only words or symbols written on paper.

In this way all the language of thinking about any field of knowledge can become calculable like mathematics.

Proposition 4

Proposition 4: Computers will be able to compute judgement.

Judgement means a decision that one course of action is to be preferred to another.

A common environment when judgement is needed is the situation in chess. A player looking at the board has to conceive of a move among many possible moves, evaluate it, and make the choice.

Computer programs that play chess have been developed for more than 20 years; several play chess at a level better than 95% of the human players who are members of chess clubs. This is good evidence.

Recently a computer program for playing backgammon has been developed, and has become the world champion. This program is called BKG 9.8 and its creator is Dr. Hans J. Berliner of the Computer Science department of Carnegie Mellon University in Pittsburgh.

The key to his accomplishment, according to some articles he has published, is the construction of a new kind of function, named a SNAC function. This function is the sum of a number of terms. Each term is a product of three numbers:

- the first number is a C or cost index
- the second number is an A or application coefficient
- the third number is an F or factor entering into the relative value

The function has to be smooth (S stands for smooth) and has to be nonlinear (N stands for nonlinear). The function computes value. And the choice of move is that move which has the most value.

This enables Berliner to represent knowledge not only by rules for a situation but also by evaluation of alternatives. Judging by the winning of the backgammon championship, the SNAC function is a valuable addition to algorithms.

Proposition 5

Proposition 5: Computers will be able to understand any natural language used by human beings, and probably the first one will be English.

The understanding of a language X means that the ideas which are expressed in sentences of X can be represented in some other language Y which is more suited to a person who knows Y.

For example, take the French sentence "deux et trois font cinq." Suppose I do not know French. Then if I have a translating scheme that "deux" means two, "trois" means three, "cinq" means five, and "et" and "font" in this context mean plus and equals, I know that the sentence is asserting "two plus three equals five."

For another example, take "he shuffled the deck." Immediately I think of a game of cards, an operation of rearranging them, and I understand. But if I have "he shuffled along the deck", my context jumps from a deck of cards to the deck of a ship; I think of an old man moving along the deck without lifting his feet.

Understanding may mean even more than these two kinds of understanding. It is an interesting and important subject even among human beings.

A great deal of work has been done by organizations in this area. One is the World Translation Co., Ottawa, Ontario, Canada. A good article describing their accomplishments is "Natural Human Languages Automatically Translated by Computer: the Systran II System" by David Burden, in "Computers and People", May-June 1981. The speed is on the order of 1000 Russian words translated into English in about 15 seconds.

Proposition 6

Proposition 6: Actuaries as well as "computer professionals" should become the main guides for society in the wise use of mathematics, probability, and computers for the benefit of man, provided actuaries widen their horizons and choose to do so.

Mathematics, probability, statistics, the study of risks, the study of methods to control risks -- all of these subjects are already a function of actuaries. For over 2500 years insurance has been used as a way of spreading the cost of risks. Also, underwriting establishments have evolved, with inspectors to examine insured properties. The purpose is to prevent insured persons from unfairly paying more premium than they should pay, as a result of paying losses due to deliberate abuse of the insurance contract by persons who commit arson, shipwreck, suicide, etc.

Now it is time to add to the computations of actuaries and their clerks, the computations of machines, not only punch card machines but electronic computers, not only the algebra of numbers, but the algebra of propositions, signals, algorithms, lists, procedures, etc.

I remember 1936 in the Prudential in Newark when I realized that the presence or absence of an X punch in a punch card was an application of Boolean algebra -- and I would never have realized it if I had not found out about Boolean algebra in college. These novel branches of mathematical reasoning are a natural extension of the professional interests of actuaries from 1850 to 1950.

Actuaries have a multi-century tradition of profession, pride, and honorable service to society. Now is the time for actuaries to go the rest of the way, and deliberately take into their territory the wise application of computers in the long wide corridors of time.

Proposition 7

Proposition 7: The pursuit of truth, common sense, wisdom, and humanity is an appropriate additional mission for the next 100 years of the actuarial profession.

As we actuaries survey the world of human beings in 1981, we can see many very alarming conditions. There are two in particular.

Before I go ahead with the discussion of these two, let me review for a moment the conditions in 1931, my first full year working in the life insurance business as an actuarial clerk in the Rates and Values Section of the Mutual Life Insurance Co. of New York at 34 Nassau St. in New York, along with Manuel Gelles and Morris Monsky, actuaries. A normal interest rate was 4%. I could live comfortably on my salary of $1500 a year. I had no fear riding the subways in New York at 5 cents per ride. The subway cars were clean, not vandalized with hostile inscriptions. The big depression was in full swing, but it did not bother me, because I had a salary and was relatively safe in a safe business. The government did not tell lies systematically, nor did the newspapers, except about Bolshevism. There was no TV. And so on.

Today the world is really hugely alarming, for two main reasons: (1) Nuclear weapons in the control of the United States and in the control of the Soviet Union can destroy all human life in a half hour at any time. (2) The capacity of the people of the United States to think and decide sensibly is being eroded by a great increase of mind-altering techniques including lying throughout the media and the government. Take one example: Reverend Jerry Falwell, an evangelist on TV, who is president of Moral Majority, Inc., stated: "You can't be a good Christian and a liberal at the same time." That is a lie; and Jerry Falwell knows it is a lie. If he does not know it is a lie, then he is stupid; and he is not stupid.

There are other reasons as well for the assertion of Proposition 7. But let us classify it as possibly true, maybe even probably true, though unproved, and go on to my last three propositions.

Proposition 8

Proposition 8: Possible implementations of this mission ((the pursuit of truth, common sense, wisdom, and humanity)) need to be a subject for actuarial study and discussion.

Actuaries do have social responsibilities. Some come to them because they are members of society like everybody else. Some come to them because their special professional knowledge gives them status as experts. One social responsibility that is much talked about by consulting actuaries is responsibility to the client. Another that is much talked about by actuaries in mutual insurance companies is the responsibility to policyholders, such as to pay claims fairly and quickly; to charge a low net cost; to act in the interest of their policyholders, which for some companies may inlcude a large percent of the people of the U.S.

Another important social responsibility is to help prevent or decrease disasters of all kinds, and to help make the hazard of disaster less and less. I see no social need for excluding any disasters whatever from four categories:

- disaster from war
- disaster from nuclear weapons or nuclear energy
- disaster from genocide
- disaster from extinction of human beings

Actuaries, like other people, often pass on these hazards to "somebody else", "the government", etc., but that is not right. It is not what the Rockefeller Foundation did in those early days of fighting hookworm. It is not what was done recently in the United Nations campaign to make the smallpox virus extinct, when the U.S. and the Soviet Union together contributed 180 million doses of vaccine.

Medical doctors led by the Physicians for Social Responsibility have begun to lecture, petition, educate, lobby, ... for a freeze on nuclear weapons and the arms race. How can actuaries refrain from saying "Here is an area of social responsibility which we must study and discuss and take action about"? You may actually save the lives of your children and grandchildren, and many, many more.

Proposition 9

Proposition 9: The most virulent problem of society currently is control over nuclear weapons and the elimination of nuclear fission energy.

Actuaries know exceedingly well a certain kind of mathematical problem: how to calculate the expectation of life.

When talking about this problem to ordinary people, I often call it the "eventually" problem. If my chance of surviving each year of my life is anything less than 1, eventually I shall die. If my chance of winning in a Nevada gambling game is less than 1, eventually I will lose all the money that I gamble.

The annual probability of another great war like World Wars I and II is perhaps 1 in 33, a war with no weapons barred. If so, the odds are excellent that mankind will not survive more than another 500 years.

We as a species have found out so much about the world of nature, and so much about the world of deceiving people, that any normally cautious person cannot bet on the survival of mankind. No species that we know about has ever before found out so much.

The highest peak of this iceberg of a problem is nuclear weapons and nuclear fission energy. Thomas J. Watson, Jr., president of IBM Corporation, gave the Commencement Address at Harvard College, June 1981, on this subject, "What Future Lies Ahead?" I suggest that everyone read it. It should be obtainable from the Public Relations Office, Harvard University, Cambridge, Mass. 02138. He says again (as President John Kennedy said before he was assassinated) if nuclear war occurs "The living will envy the dead."

Proposition 10

Proposition 10: It is not right for actuaries and managers in insurance companies to exclude the risks of nuclear weapons, nuclear fission energy, war, and genocide, in (a) policy contracts, (b) thinking and discussion, and (c) political and professional action.

By "right" I mean "socially responsible". "Morally right" is something else; a slaveowner in the pre-Civil-War atmosphere of Mississippi probably thought it was morally right to sell a black husband in New Orleans and his black wife and babies in Memphis, Tennessee.

It is plain and clear nowadays that the interests of all the people all over the world require a world where the population is not too great, the resources are shared reasonably and widely, cannibalism is eliminated, slavery is eliminated, public health is maintained everywhere, and patriotism means a loyalty to the best interests of all mankind. It is time for the humanitarian concern for whooping cranes and for whales to reach to all men and women.

The excluding of risks from insurance contracts dodges, bypasses the concerns that thinking people and actuaries both must have over the dangers of war, nuclear weapons, nuclear fission energy, and genocide. Make no mistake: genocide and the desire for it is very easy to spread. When Rome sacked Carthage in 146 B.C., nine tenths of 700,000 people were slain, and the rest sold into slavery.

Adlai Stevenson spoke of "our fragile spaceship Earth". Let us bestir ourselves and deserve our good fortune to live on this magnificent life-supporting planet. Ω

CACBOL – *Continued from page 3*

WW Component Supply Inc. / 1771 Junction Ave., San Jose, CA / 95112 / (408) 295-7171
Complete supplies for computers; peripherals

Nycom / 4500 El Camino Real, Los Altos, CA / 94022 / (408) 948-4500
Business, educational, scientific, personal; hardware, software, service

Byte Shop / 1122 B Street (between 2nd and Foothill), Hayward, CA / 94541 / (408) 537-2983
Small business, personal, software, classes, service; Apple, Atari, Cromemco, IBM, NEC, Texas Instruments, many others

Computerland of San Jose / 1077 S. Saratoga-Sunnyvale Rd., San Jose, CA / 95129 / (408) 253-8080
Small business, personal; Apple II, Dynabyte, North Star, Cromemco, Altos, Texas Instruments, Atari; software; repair

Professional Business Systems / 6038 Calle De Rico, San Jose, CA / 95124 / (408) 448-6674
Data processing systems, sales, installation, service, training

(Source: Michael Goldstein's notes)

27 COMPUTER STORES IN THE LOS ANGELES, CALIF. AREA (List 810905)

American Word Processing Co. / 18730 Oxnard, Los Angeles, CA / ? / (213) 705-2245
Data processing and word processing; supplies and accessories

Chess and Games Unlimited / 10868 W. Pico Blvd., Los Angeles, CA / 90064 / (213) 474-5544
Home computers; chess, backgammon, games

Comet Computers / 1339 W. Garvey Ave., Los Angeles, CA / ? / (213) 962-7051
Business, home, personal, education; microcomputers and accessories

Computer Aide / 1105 Olancha Dr., Los Angeles, CA / 90065 / (213) 258-8800
Software for education; Apple

Computerland of South Bay / 16720 Hawthorne Blvd., Lawndale, CA / 90260 / (213) 371-4624
Small business; "All major microlines"; data processing; supplies and accessories

Computer Place / 2530 W. Sepulveda Blvd., Torrance, CA / 90505 / (213) 325-4754
Business, industry, education, home; software

Computerworld / 5848 N. Sepulveda Blvd., Los Angeles, CA / 90049 / (213) 786-7411
Professional, personal; Alpha Micro, Cromemco, Vector Graphic, Apple, Atari, Texas Instruments, Hewlett-Packard; software

Computique / 11986 Wilshire Blvd., West Los Angeles, CA / 90025 / (213) 820-5761
Apple; calculators

Data Equipment Supply Corp. / 8315 Firestone Blvd., Downey, CA / 90241 / (213) 923-9361
Business, legal, real estate, medical, educational, games; peripherals; training, courses; Commodore, PET, Apple, TRS-80

Data Systems Marketing / 6820 La Tigera Blvd., Los Angeles, CA / 90045 / (213) 641-2050
Home, business; Texas Instruments; microcomputers; training; terminals and printers

EduComp / 4445 South Santa Fe Ave., Vernon, CA / 90058 / (213) 582-9422
Education, home, business, personal; Texas Instruments TI-99/4

HW Computers / 19511 Business Center Dr., Northridge, CA / 91324 / (213) 886-9200
Business, home, education; software

Instant Replay Equipment Co. / 5520 Westlawn Ave., Los Angeles, CA / 90066 / (213) 390-4417
Buy, rent, lease terminals, printers, microprocessors, software

Jade Computer Products / 13440 S. Hawthorne Blvd., Hawthorne, CA / 90250 / (213) 973-7330
Computer parts, boards, peripherals, systems, software and supplies

Khalsa Computer Systems Inc. / 500 S. Lake Ave., Pasedena, CA / 90057 / (213) 684-3311
Small business, personal; software; classes; peripherals and supplies; service

Micronomic Computer Systems / 10871 W. Pico Blvd., West Los Angeles, CA / 90064 / (213) 474-3389
Sales - service - software - supplies

National Computer Sales / 18013 Sky Park Circle, Irvine,, Los Angeles, CA / ? / (213) 603-9933
Small business computers

Olympic Sales Co. / 216 South Oxford Ave., Los Angeles, CA / 90004 / (213) 381-3911
Educational, small business, hobbyist; Apple, Hewlett-Packard, Texas Instruments, Ohio Scientific, Mattel, APF, Atari, Commodore

Professional Computers Inc. / 10885 Washington Blvd., Culver City, CA / 90230 / (213) 836-5005
Business; Ohio Scientific

QT Computer Systems / 15335 S. Hawthorne Blvd., Lawndale, CA / 90260 / (213) 973-2619
Apple, S-100 specialists

Rainbow Computing Inc. / Garden Plaza Shopping Center, 9719 Reseda Blvd., Northridge, CA / 91324 / (213) 349-5560
Apple, Atari 400 and 800, Digital

RPS Electronics Inc. / 1501 South Hill, Los Angeles, CA / 90015 / (213) 748-1271
Business; Texas Instruments

SCS Retail Store / 9522 Jefferson Blvd., Los Angeles, CA / 90011 / (213) 836-7746
Buy and sell small - mini - micro - computers; peripherals and systems

Sunshine Computer Co. / 20710 Leapwood Ave., Carson, CA / 90746 / (213) 515-1736
Business, engineering, education, personal

Terminal Brokers Inc. / 4265 Marina City Dr., Marina Del Rey, CA / 90291 / (213) 822-3900
Computer terminals, terminal stands, cables, telephone modems, business systems

The Computer Store / 820 Broadway, Santa Monica, CA / 90401 / (213) 451-0713
Personal, business, word processing; Apple, Atari, Computalker, Dynabyte, Hazeltine, Integral Data Systems, Micropolis, Microsoft, Mountain Hardware, Nippon Electric, Programma, Sanyo, Softape, Structured Systems, Summagraphics, Teletype, Texas Instruments, Vector Graphics,

Thorpe Datasystems Inc. / 22968 Victory Blvd., Los Angeles, CA / ? / (213) 703-6900
TRS-80 support; hardware; software; service

(Source: Michael Goldstein's notes)

---

* - from pages 2 and 3
(Source: Appendix 3 of "The Computer Revolution" by Edmund C. Berkeley, published by Doubleday & Co., Inc., Garden City, New York, 1962, 249 pp) Ω

Computing and Data Processing Newsletter

PAT WALLACE: NAVIGATIONAL SUPPORT FOR OCEANOGRAPHIC RESEARCH

George Jones
Western Electric
222 Broadway
New York, NY 10038

Pat Wallace spends 70 percent of her time on the high seas. She provides navigational support for oceanographic research. She's an engineering associate with the Government Systems Group of Western Electric, helping the U.S. Navy conduct underwater studies. She majored in electrical technology and before her present job served a hitch in the Air Force.

"I'd never been on a ship before. Going from the Air Force to working as a civilian on Navy ships was a big transition. But I love it. Nothing is more beautiful than the sea."

Pat Wallace, an engineering associate for Western Electric, plots navigational information on a map in her job on board Navy ships.

Since joining Western Electric seven months ago, she has completed three cruises varying from one to two months. She operates electronic navigational systems, plotting the course of the ship, and using and maintaining computer systems. The navigation systems include long range radio, satellite, and an acoustic transponder system -- a network of transponders on the ocean floor that echo navigation signals.

The information is put on a digital plotter and integrated into the ship's computer. She is responsible for calibrating, maintaining, and repairing the equipment, including the computer. So far Wallace has avoided the main occupational hazard -- seasickness -- though on her last trip a storm made the ship roll 30 degrees, and even the medics were sick. The biggest adjustment for Wallace is advance planning when she doesn't know how long she'll be at sea: "I pay my rent a few months in advance, have a neighbor take in my mail, and water my plants."

75 YOUNG COMPUTER WIZARDS AT NEW YORK UNIVERSITY

New York University
Public Affairs Dept.
Press Office
21 Washington Place
New York, NY 10003

The Courant Institute of Mathematical Sciences, a world-renowned research center in computer technology, physics, and mathematics, played host to some 75 high school students of the New York area this past summer.

Ronnie Schnell, 14 years old, of Spring Valley, NY, first introduced himself to computers at a community college course at age 9: "I was just curious." He did work during the summer "testing out Pseudo-Parallel SETL." He graduated from Pomona Jr. High School recently; then worked part time on "breakproofing" an information utility against raiders.

The young computer people were taught during the summer six computer languages: FORTRAN, PASCAL, BASIC, ARTSPEAK, SNOBOL, and COBOL; and computer hardware. They had unlimited access to all of the Courant Inst.'s high speed computers. They often worked Saturdays and evenings.

Henry Mullish, Sr. Research Scientist, who has administered the program for several years, says many kids come from New York's two top high schools, Bronx Science and Stuyvesant. About a third are of Asian parentage. Many are children of immigrants - Russian Jews and Israelis. And he says "The kids are usually very beautiful."

COMPUTER GUIDANCE HELPS FEED 8000 STUDENTS

Robert A. Reid
IBM Corp.
Data Processing Div.
3424 Wilshire Blvd.
Los Angeles, CA 90005

Managing food services on a campus with 8000 students is a challenge to Al Deskiewicz, Jr., amid food quality considerations, student grumblings, and an ever-tighter budget for food. He is manager of the food operations of the student union at the Univ. of Idaho in Moscow, Idaho.

Two years ago, the food facility's entire inventory of 1100 items was hand calculated and entered on worksheets; there was no duty roster of personnel, no work schedules; and accounting records were 45 days old.

Now a computer schedules duty hours for 75 employees, processes payroll, tracks inventory, generates purchase orders, vendor analyses, keeps accounting records current, and issues reports to management.

Benefits from these operations include a saving of $10,000 since employee duty hours were automated; payrolls produced in 3 hours instead of 24 hours, and inventory has been reduced to 900 items.

Sometimes a student fails to report for a shift, or cancels at the last minute because of an exam. Because of computer access via a display terminal, Deskiewicz can identify a substitute with similar skills, available to work that shift at short notice.

Deskiewicz regularly reviews costs of all items, and assigns them acceptable price ranges. When cost exceeds that range, he can substitute a lower cost item or select another vendor.

For programming, he used work-study students and computer science department projects for class work. This might set a world record for least-cost achievement in programming!

THE DESIGNING OF RESEARCH TO ACCOMPLISH A PURPOSE: PDA, "PURPOSE DESIGN APPROACH"

Lynn Simarski
Univ. of Wisconsin – Madison
610 Walnut St., Rm. 1215
Madison, WI 53706

The gypsy moth's devastation of forests in the northeastern U.S. prompted the U.S. government some years ago to fight back. But more than 3 years of exhaustive biological research and modelling of systems failed to produce a remedy for the ravages of the gypsy moth.

"Researchers had not identified the purpose of the gypsy moth program", says Gerald Nadler, industrial engineer, Univ. of Wisconsin-Madison. "Everyone involved assumed the program's purpose to be obvious, and that a solution would emerge after enough data were collected."

Nadler begins by asking the federal agencies to specify the purposes of the moth program. The agencies finally defined the program's aim: to cope with, rather than eradicate, the gypsy moth.

Nadler says "The interchange between the experts and those who have the problem achieves the results. It is not the elegance of the model which will produce change in the real world."

How will the research results be used in the future? Who will implement the solution produced or suggested by the experts? How will they do it? Is the right problem being worked on? What are the alternate solutions? What about the political setting? What about the personalities of officials? What are the goals? How do we stay in touch with the real world where the change must take place?

This is the class of questions which the PDA (Purpose Design Approach) asks again and again.

WRITER AND PUBLISHER USES PERSONAL COMPUTER TO PREPARE STORIES, CHANGE THEM, KEEP TRACK OF SUBSCRIBERS, MAKE LISTS, AND DOUBLE PRODUCTION

Phil Roybal
Apple Computer Inc.
10260 Bandley Drive
Cupertino, CA 95014

John Tiner, writer, editor, and publisher, has almost doubled his productivity in the past year since he added a personal computer to his traditional editorial tools.

He produces a quarterly newsletter which is mailed to subscribers throughout the nation. It carries recurring features, such as calendars of events, listing of seminars and lectures, etc. "I used the computer memory to store these lists so that updates could be made in minutes instead of retyping all the material. In addition I maintain the list of subscribers and my own correspondence."

He has 1500 names in the computer memory, categorized by special interest in the newsletter: advertisers, book editors, institutions, etc.

He also uses an electric typewriter and can edit manuscript using the computer, making additions, deletions, and changes without retyping the material. He also writes material for books and articles, and he has increased his monthly output from 120 pages to 250 pages. "Often I can sell the same article 20 times." Then with a planned series of articles, and some minor editing, he can make a book and market the book, in several versions, for several markets, such as children, teens and adults.

Energy, imagination, and personal computer make a formidable combination.

The total investment in the system is about $2500 for the personal computer, printer, and software, plus the value of the television set, the value of the electric typewriter, and the value of the work connecting them together.

Ω

Games and Puzzles for Nimble Minds – and Computers

Neil Macdonald
Assistant Editor

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

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

NAYMANDIJ

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

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

NAYMANDIJ 8109

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

MAXIMDIJ

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

MAXIMDIJ 8109

NUMBLES

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

NUMBLE 8109

```
            D E A T H
        *   M A K E S
        -------------
          H B E D E S
        V K T M S R
      H E B H M M
    H A K B D S
  V M D E S A
  -------------------
= B K M T T I R D Y S
```

B=V
E=I=Y

10515 95

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

SOLUTIONS

NAYMANDIJ 8105: Column 9: under 3.
MAXIMDIJ 8105: He is a slave who cannot live on little.
NUMBLE 8105: God curses haste.

Our thanks to the following people for sending us solutions: T.P. Finn, Indianapolis, IN – Numble 8105, Maximdij 8105; Steven Shulman, Edison, NJ – Maximdij 8105, Naymandij 8105; Steve Werdenschlag, Livingston, NJ – Maximdij 8105, Numble 8105.

The Frustrating World of Computers

by Harry Nelson
1135 Jonesport Court
San Jose, CA 95131

SIR, IF I TRY TO CORRECT OUR COMPUTER, IT WILL FEEL I'M ACCUSING IT OF MAKING AN ERROR——

OUR COMPUTER IS INFALLIBLE AND WE HAVE THREE OF THE TOP PROGRAMERS IN THE COUNTRY——

WELL, BEFORE WE GOT THIS COMPUTER WE MADE OUR OWN MISTAKES——

I DO TOO KNOW A MISTAKE WHEN I SEE ONE——