"Software is truly the mind, heart, and soul of the system," General Chairperson Terry Straeter (left) said in opening the Seventh International Conference on Software Engineering in Orlando, Florida. That theme was picked up by Herbert Simon in his keynote address. William E. Howden (center) served as program chair. Jean-Claude Rault (right) and Leon Stucki (not shown) served as program co-chair and Software Tools and Exhibits Committee chair, respectively.

ICSE keynote address: "Does software need to be artificially intelligent?"

Ware Myers, Contributing Editor

"I won't hold you in suspense. I wouldn't have selected that title unless I thought the answer was yes," Herbert Simon, Nobel Laureate at Carnegie-Mellon University, told the keynote audience at the Seventh International Conference on Software Engineering in Orlando, Florida, March 27. "The burden of my remarks is that software engineering and artificial intelligence are either going to have to join hands or that software engineering people are going to have to re-invent artificial intelligence."

For the past 25 years everyone has been pointing in panic at the burgeoning software tasks that face the computer community, Simon noted. We have not met total disaster—at least not often, he continued. That's because we have increased the size of the software engineering community ("so that we can fill rooms like this"). We have been structuring the task more successfully, and, without thinking of artificial intelligence,
we have been moving the boundary between man and machine. We have been putting more of the programming load on the computer through use of higher level languages, compilers, and interpreters.

"I think most of us would agree that compilers are algorithms without intelligence," Simon asserted. "All of the intelligence has been put in at the time of construction of the compiler." A compiler has a "definable, formalizable, straightforward procedure" for converting terms in a highly formal language to machine code. Between the formality of the language and the formality of the compiler, "all the intelligence has been squeezed out of the process."

Methods of artificial intelligence. What compilers don't do is convert unstructured information to a form on which a "mindless algorithm" can work. Humans beings can do this. They use what Simon called weak methods. "Weak methods are fallback procedures," he said, "that one uses to solve problems where there aren't any powerful, systematic methods around, where there is not enough information, or where the information has not yet been sufficiently structured." These are the very methods, of course, that have preoccupied the artificial intelligence community. They are the only methods that can be applied when the situation is not highly structured.

"Most of these weak methods involve something we call heuristic search in a problem space," Simon explained. By search we mean a program that undertakes some amount of trial and error and has the capability of noticing whether progress is being made toward a goal. If the search appears to be successful, it continues forward on the existing path. If it is not succeeding, it backs off and tries something else.

In this type of problem the space to be searched is enormous. Blind search, by itself, would never reach the goal. Thus, it has to be guided by some modicum of intelligence—that is, by heuristics or rules of thumb.

"Heuristic search must try only a small percentage of all the paths that are available, paths that have a much better than chance probability of yielding a path toward the goal," Simon said. "The characteristic of these paths is that there are no sure-fire procedures for searching that are guaranteed to lead to the result. If there were such methods, of course, you wouldn't be fooling around with artificial intelligence. You would simply write an algorithm."

Problem space refers to the large set of possibilities within which the solution may lie. It is the space traversed by the search.

"It is not enough to have great methods," Simon continued. "You have to have knowledge about the task domain." That means a database of knowledge. This knowledge base has to be stored in a "highly indexed form" so that it is quickly accessible. It has to be set up in a data representation form that fits the problem domain. And it has to provide a knowledge of strategies, such as sorting methods.

Application of AI to software. Software was called "the mind, heart, and soul" of a system by Terry Straeter, conference general chairperson, in his welcoming remarks. But it is not the exclusive mind, Simon noted, because there are also people in the overall system. "The question is what part of the responsibility rests in the machine and what part in the human beings," he said. "This is clearly going to be a shifting boundary."

At present this boundary lies between a high-level highly formal language written by a programmer and a compiler run by a computer. In the middle-range future the boundary will move back to the specification. "The problem is to understand enough about the problem-solving processes that take us from the formal specification of a program to a running program to be able to do it completely automatically—that is, to go from the description of what we want to the actual procedures." Simon believes this task is going to require heuristic search methods.

In the long run the research task is to go from a problem stated in natural language by a user to a running program. "One can conceive of an interactive system—and there have already been experiments"—that would carry on a conversation with the customer or the user to develop the information which the automatic part of the system knows it needs for an adequate specification to go to the next step," Simon said in response to a question. "It seems to me that the first steps in this process are obvious. The farther we go in formalizing our specifications the clearer it becomes what information we need in order to write those specifications and therefore what that (natural language) front end should look like."

*Some of this work was later reported to the conference in a session entitled "Knowledge-Based Software Engineering." See especially Gerhard Fischer and Matthias Schneider, "Knowledge-Based Communication Processes in Software Engineering," Proc. Seventh ICSE, pp. 358-368.

Conference report: International Software Update 1984

Dennis R. Allison, Stanford University

Maui was a perfect place to hold International Software Update 1984. The early January weather was warm, the beaches inviting, and the trade winds fresh. And the concept was certainly exciting. Dave Russell of Raging Bear Productions had done a good job of setting the stage for an exciting workshop-format meeting. The invited speaker roster included the chief executive officers of most of the ranking microcomputer software suppliers—gathered together to discuss the international aspects of making and selling software. There were activities throughout the week, with the heavy discussion sessions scheduled for midweek.

Very early, however, it became obvious that there were problems. First of all, the January 8-14 schedule of International Software Update conflicted with that of Comdex, forcing attendees to make a moral decision between commercial necessity (Comdex) and sybaritic conferencing in Maui. Second, the harsh realities of the software business caused some major mutations in the speaker schedule. Last, the audience was smaller than expected. The result was a conference that was an aesthetic and technical success but an economic disaster. (The back-hall conversations and the gossip were great.)

The conference theme, How To Address the International Software Market, was interpreted differently by the various speakers and panelists. Everyone agreed that the international market for personal computer software is just beginning to become important, but that it will be increasingly important in the future. It is a market that requires that software designers understand the national character of each country and customize the software to its need.

The overall meeting ranged over an ungainly set of topics. Some highlights appear here.

Language problems. Steve Rampell, president of Micropro Japan, gave a fascinating presentation on the problems caused by language and its representation. Character sets may differ from ASCII, the American Standard, and keyboard layouts may change from country to country, as do the collation sequence and the preferred notation for dates, decimal numbers, and currency. These are not trivial difficulties for two reasons: (1) they are often buried deep in the design of the program and hard to
change, and (2) they must be changed to succeed in the international marketplace.

Even such seemingly trivial matters as menu sizes and arithmetic precision can cause major problems. A menu in German can require 25 to 30 percent more space than the same menu in English—a real problem if the screen layout is tight. And as Mike Belling of Stoneware, Inc., observed, a spreadsheet program with nine digits of precision is not destined for success in Italy where million-lira transactions are commonplace.

The problems of word processing for Asian languages are even more difficult. Each language presents its own set of problems. The Japanese language, for example, uses four different alphabets, one of which, Kanji, includes over 3100 characters in a newspaper-level vocabulary. Imagine what that does to a program that was designed to represent all characters as eight-bit bytes and that assumes the high-order bit is free and available for its use!

For these languages, a typewriter keyboard is not much use. Rather than capture keystrokes that represent the characters, modern Japanese word processors capture a phonetic representation of the character and look it up in a phonetics-versus-character dictionary. Unfortunately, Japanese pronunciation is not specific, and many sounds have several different character representations. When a pronunciation question occurs, the program must interrupt and ask the operator to select alternatives from a menu. At the same time, the program keeps track of the context of the material being entered and makes an appropriate selection of phonetics based on that information. Because the text-understanding methods are still primitive, eventual human verification of the choice is required.

Just where this kind of support should be located is a matter of some contention. It is complex software needed by a large number of programs; should it be part of the operating system, or should it be part of the application? Again, this problem is only one element of the international market problem.

**Software protection.** Everyone at the conference was concerned in many different ways with the problems of software protection. Software has become a commodity product with domestic and international sales in the neighborhood of $2 billion. Software is difficult to create but easy to copy. Thus, for every copy of a piece of software that is legally sold, five to 50 or more copies exist that generate no revenue for the software developer and distributor. Needless to say, this makes the software developers very unhappy. One supplier, Larry Wilbur of Redwood Corporation, explained how an intelligence-gathering system provides him with a way to service his client base better while identifying the pirates. Even with a substantial educational effort on his company’s part, nearly 10 percent of overall company revenue is recovered from pirated copies. But then, as John Hall, president of Peachtree, said, “We are all pirates.”

It is very difficult to protect software. For several years some manufacturers have been copy-protecting the disks their software is distributed on. Protection techniques are various but are easily subverted by an appropriately tenacious hacker. Worse, a bulletin-board network broadcasts ways to break the protection throughout the US. It’s a continual cat-and-mouse game.

The traditional software marketing approach has been to sell a license to use a single copy of a program on a single CPU. When all systems are based on floppy disks, this is a reasonable strategy. When hard disks and networks become widely used, however, that strategy becomes more difficult to implement.

David Cole, president and CEO of Ashton-Tate, described a multipronged attack on the problem of pilferage, especially copying. Cole divided the maleficients into four groups: professional pirates who copy and sell, those who aid and abet the crime by making paraphernalia or renting software for evaluation, casual home and hobby hackers, and the corporate pilferer. It’s the last category—the corporate purchaser who buys one copy and distributes it to many sites—that represents the primary problem.

Cole’s approach to minimizing losses from piracy includes an education program, visible enforcement, improved technical protection, and lobbying efforts to improve the current range of legal protection. This effort is being spearheaded by MCSA, a division of Adapco. In addition, an ad hoc organization, the Software Protection Fund, is working in cooperation with MCSA to devise realistic actions to frustrate illegal copying.

**Cautionary tales and the future.** There were cautionary tales also.

It is very expensive to buy back the right to use your product’s trademarked US name if you have failed to register it in all countries in which you plan to do business—or so it was found by David Gordon, president of Datamost. It was also found best to thoroughly check anyone who proposes to represent you in a foreign country; a fancy business plan and the company it describes may all be a facade. Contracts and agreements in the international arena can be complex and difficult. President Ed Currie of Lifeboat provided some guidelines, while many other speakers spoke of the problem with the direct knowledge of experience.

Mitch Kapor, president of Lotus Development Corporation, saw machines as still rapidly evolving; to be successful, a company must be able to adapt quickly to changing technology. While conventional wisdom holds that personal computer software is market-driven right now, Kapor argues against the belief that a second-rate product with first-rate marketing is always successful. There will always be a market for quality and innovation. But there isn’t much innovation, he complains; the mouse-window scheme just now becoming popular is over 20 years old. How good are our innovative abilities if we cannot devise at least one other metaphor? Perhaps, he speculated, artificial intelligence programs will be the next fad. And, maybe he is right. Company presidents are paid to predict the future, or so he says.

**NCC 84 educational session explores PC impact, trends**

“Enhancing Creativity with Computers in Education,” an educational session planned for the 1984 National Computer Conference in Las Vegas, Nevada, will acknowledge the impact of the electronic educational market on the personal computer industry. Scheduled for 1:30 p.m. Tuesday, July 10, this session will present representatives from IBM, AT&T, CBS Software, and DEC, who will attempt to forecast the future of computers in education.

Bob Wallace, educational marketing director for IBM, will discuss his company’s role in educational computing. AT&T’s Harry Shoemaker plans a presentation entitled "The Interactive Videodisc and Individuality." Henry A. Kaplan of CBS Software will discuss developing and marketing self-improvement software for home use, and Bob Trocchi of DEC will speak on educational mass communication.

Individuals interested in further information on this session can write the organizer, LaRuth H. Morrow, president of Stellar Solutions, at 1317 Navaho Trail, Richardson, TX 75080, or call (214) 231-6454.
Software emphasized in new CSE model program

The Computer Society's new Model Program in Computer Science and Engineering places a "greater emphasis on software," according to Taylor L. Booth, who served as vice president for educational activities during the guideline's development. The December 1983 publication updates and expands the society's 1977 Model Curricula in Computer Science and Engineering, reflecting the change in technology since that time.

Computer science and engineering is a separate discipline from electrical engineering, Booth noted, and the guideline "creates an integrated program that balances the software and hardware needs of the profession."

The recommended program satisfies the new Accreditation Board for Engineering and Technology criteria for computer science and engineering programs, but the report goes beyond curriculum recommendations to address resource planning and problems of implementation. The report will be useful to departments preparing new programs or expanding and updating existing ones.

The new guideline is the result of over two years of effort by the Educational Activities Board's Model Program Committee. The project was initiated in 1981 when it became apparent that the dynamic nature of the computer field necessitated expanding project scope to address all aspects of a computer science and engineering program. The report was developed during 1982-83 under the leadership of J. Thomas Cain, Glen G. Langdon, Jr., and Murali R. Varanasi.

Copies of the CSE report have been sent to every Electrical Engineering Department in the US, Booth said, and the committee will provide informal advisors to institutions desiring such assistance. Currently, there are 28 accredited CSE departments. Booth expects this number to increase to 50 in the "not too distant future."

Additional information on the guidelines can be obtained from the current vice president for educational activities: J. T. Cain, Dept. of Electrical Engineering, University of Pittsburgh, Pittsburgh, PA 15261.

The 154-page report can be ordered from the Computer Society's West Coast Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; (714) 821-8380. The nonmember price is $20; the price to IEEE and/or Computer Society members is $10. Specify order number 932, include member number if applicable, and add $3 to cover postage and handling.

Softcon keynote: Personal computers must become "desk appliances"

"High tech" and "high touch" must be combined in the same product if personal computers are to exceed their current five-percent penetration of the market represented by 25 million North American "knowledge workers," according to Steven Jobs, chairman of Apple Computer. The concept of a personal computer on every desk to improve worker productivity won't work, Jobs asserted, as long as people have to spend 40 to 50 hours learning 400-page manuals.

There has been only one "desk appliance"—the telephone—in the past 100 years, Jobs reminded the audience during his Softcon keynote address February 21 in New Orleans. "Personal computers," he said, "must become the second." But to do this they must be not only useful and cheap, they must also be easy to use, and Jobs predicted an 18-month logjam in the industry until that becomes more of a reality.

The new generation of products now being introduced—among them Apple's Macintosh—parallel the telephone in their "radical ease of use," according to Jobs. "People are learning to use Macintosh in under 30 minutes," he said. "Macintosh and other products like it have the chance to become our industry's first telephone—the second desk appliance, the first one in over 100 years."

With Macintosh, Job's company plans to "recreate the Apple II phenomenon" in which software was produced by over 200 vendors. "Software is the most important thing," he said, "and Apple will never do more than one percent of its own software."

Softcon, the first nationwide trade show devoted exclusively to software, was held February 21-23 in New Orleans. With over 20,000 people trampling through the 13-acre Louisiana Superdome to look at 12,000 software packages displayed by more than 60 manufacturers, exhibitors resorted to attention-getting devices that ranged from balloons to mimes and from roulette wheels to robots. The hands-down winner in the crowd-gathering sweepstakes was the adept young woman at right, who juggled, twirled, and balanced to emphasize key points in her product spiel.
**Database computer enhances programmer productivity**

*Ware Myers, Contributing Editor*

If the relational database model contributes to programmer productivity, then a database computer embodying the relational model should also do so.

For the first clause we have the authority of E. F. Codd in his 1981 ACM Turing Lecture.* "It is well known," he asserted, "that growth in demands from end users for new applications is outstripping the capability of data processing departments to implement the corresponding application programs. There are two complementary approaches to attacking this problem (and both approaches are needed). One is to put end users into direct touch with the information stored in computers; the other is to increase the productivity of data processing professionals in the development of application programs. It is less well known that a single technology, relational database management, provides a practical foundation for both approaches." Codd then devoted the body of his lecture to setting forth the reasons for this conclusion.

Unfortunately for the application of the relational model—particularly for large-scale production systems—when it is implemented in software on a mainframe, it has drawbacks. Complex, multilevel requests execute slowly; performance is poor; and resource costs are high.

For the second clause we have the new database computer announced in January by Teradata Corporation. Founded in 1979 by Jack E. Shemer (a former technical editor of Computer) and a number of associates, Teradata set out to address these problems. First, it enlisted the advisory services of 10 "partner" organizations of the caliber of Proctor & Gamble, Merrill Lynch, and Standard Oil of California. Initial interviews with the partners confirmed that "the relational database was on the path of the future," as Teradata's Dave Clements put it in an executive briefing January 31.

"These firms were eager to have the ease of use and productivity gains that are inherent in the relational model," Clements continued. The partners helped Teradata establish the user requirements that came to fruition in the announcement of the DBC/1012 database computer.

**Database computer.** The new product employs from six to 1024 8086 micro-processors with 8087 floating point co-processors, operating in parallel. Some of the processors are specialized and programmed to interface with a host mainframe. Most of them, however, are connected to Winchester disk storage units in which the relational database resides.

The processors are interconnected with an active-logic tree-structured network called the Y-net. This patented net is the key technical concept on which the DBC/1012 rests. It performs two principal functions: (1) It broadcasts work steps developed by the interface processors to the disk processors, and (2) it merges results (data) from the disks in the order specified by the user. It performs these duties at the rate of six megabytes per second. Since for reliability reasons two Y-nets are used, the total transmission rate is 12M bps.

The processors, running at 8 MHz, each perform a little over 0.4 million instructions per second. The disks store 474M bytes and handle 30 random accesses per second. Thus, the minimum six-processor configuration—two interface processors and four disk processors—performs 2.4 MIPS and 120 accesses per second and stores 1.9 gigabytes. Teradata has proposed a 200-processor configuration to one of its partners with the following capabilities: 80 MIPS, 6000 accesses per second, and 94.8 G-byte storage capacity. "At 500 processors," Clements noted, "we approach the workload requirements that both we and our partners characterize as future requirements." The partners estimate that their databases are growing at an average rate of 45 percent per year, thus anticipating a trillion-byte database by the end of the decade.

At $480,000 for the minimum six-processor configuration, the DBC/1012 is relatively economical. "In studies that we have done with our partners," Clements reported, "we have found that a DBC/1012 system can do the job for typically one-third the cost of a mainframe, all-software relational approach."

**Software.** Though the DBC/1012 is based on a hardware approach to the relational model, it embodies over 750,000 lines of code. Part of that software resides in the host:

* Tequel (Teradata Query Language) for data definition, manipulation, query, and control; similar to IBM's Sequel;
* Iteq (Interface Tequel) for direct and user access to the database via  

---

**News from ACM**

**216 members.** Dineh M. Davis, an assistant professor in the Computer Information Systems Department of Bentley College, Waltham, Mass., has been recognized as the 65,536th member of the Association for Computing Machinery. A framed ACM Membership Certificate was presented to Davis at a luncheon ceremony at the ACM Computer Science Conference in Philadelphia, Penn., on February 16, 1984.

"The recognition of the association's 216th member has no relation to certain 16-bit machines," ACM President David Brandin assured the audience while making the presentation. "It just seems like a good idea to acknowledge a binary milestone. We don't get this chance very often."

Davis claimed to have planned her selection and free trip to Philadelphia by means of an "as yet unperfected" computer program she had written.

**Dissertation award.** Thomas Reps received the 1984 ACM Doctoral Dissertation Award for his thesis, "Generating Language-Based Environments," during the ACM Computer Science Conference. Reps received his PhD in 1983 at Cornell University where he is a post-doctoral associate in the Department of Computer Science. The thesis, which has been published by the MIT Press, applies formal techniques and rigorous analysis to a practical problem—the incorporation of language-specific knowledge in interactive programming environments.

**Codosyl develops menu-driven interface.**

The DoD's Codosyl-COSCL Committee has developed a menu-driven operating system interface called Video-COSCL designed for use with networks, hosts, and personal computers. The Common Operating System Control Language Committee plans to consider additional user-friendly methods for the interface, such as graphics and pointer techniques.

Company-sponsored individual contributors with backgrounds in operating systems' design, human factors, or screen-generating processors are invited to join in the committee's work. Interested individuals should contact the vice chairperson: Eric H. Clamons, Honeywell Information Systems, Inc., PO Box 8000, MS Z20, Phoenix, AZ 85066. COSCL will meet in June in Wickenburg, Arizona.
a terminal connected to the host computer;
• Cobol preprocessor for embedding Tequel statements in Cobol source code.

The interface processors contain transaction-translation programs: the parser and a work-step dispatcher. The database management software resides in each disk processor.

The database computer enhances software productivity in at least five ways:
(1) It provides the application programmer with a ready-made relational database, relieving him of the time-consuming task of planning the supposedly permanent arrangement of the data required by other approaches.
(2) It removes the responsibility from the programmer of unloading and reloading the database when the system is expanded. (The DBC/1012 automatically redistributes the existing data equally over all available disks. This equal distribution is what enables each one of perhaps hundreds of disk processors to share the database management workload.)
(3) It provides a macro capability in its query language, permitting lengthy procedures to be worked out once and thereafter called by name.
(4) It permits Tequel, a fourth-generation, nonprocedural query language that is more efficient than Cobol, to be embedded in Cobol application programs.
(5) A fault-tolerant system with optional fallback capability, it permits the programmer to ignore this area in his application code.

Legislation to enforce "shrink-wrapped" license agreements introduced in California, Louisiana

A multimillion dollar drain on California's economy may have come a step closer to being plugged as Assemblyman Gray Davis (Democrat-Los Angeles) announced action to halt the "piracy and pilferage of computer software.

At an April 5th news conference in Los Angeles, Davis announced plans to introduce legislation to halt the illegal copying of software, projected to cost software publishers $700 million nationally in 1984. California's share of the loss was estimated at $140 million. Total US software revenues are projected at $3.5 billion for 1984.

The news conference, held at the Los Angeles Computer Center during Comdex, a nationwide computer manufacturers' exposition, was sponsored by Vault Corporation of Westlake, California. Vault, which develops and markets copy protection systems, previously worked with legislators and Secretary of State James H. Brown of Louisiana to have a similar bill, called the "Software License Enforcement Act," introduced in that state's current legislative session. Assemblyman Davis, who represents Los Angeles' west side and parts of the San Fernando Valley, said the legislation is linked to a license agreement printed on the wrapper or inserted in a software program package. In opening the package, the software purchaser becomes a licensee of the company that produced the software. Such license agreements typically prohibit any copying, modification, transfer, sale, or rental of the software.

Davis noted, however, that serious questions exist as to whether such license agreements are fully binding and enforceable. At least one alleged software pirate in California claims "they're not worth the paper they are written on," he said.

"This bill is intended to strengthen significantly the ability of software publishers and distributors to enforce their rights," Davis said. "It does make it clear that these so-called 'shrink-wrapped' license agreements would be enforceable under California law."

W. Krag Brophy, Vault chairman, noted that although computer software is protectable under US copyright law, the existence of an enforceable license agreement, backed by state law, "is critical in preserving some of the copyright owner's rights under copyright law."

Attorney Allan Grogan of Blanc, Gilburne, Peters, Williams & Johnston, who prepared the California draft legislation, said, "The bill strikes a balance between the legitimate interests of the software industry in preventing piracy and the legitimate interests of the customers who acquire copies of software pursuant to license agreements. Only those provisions which may significantly enhance the ability of software publishers and distributors to enforce their rights under trade secret law and copyright law are affected."

Essentially, both the California and Louisiana bills state that if a party acquiring the software uses it, or opens the package, he will be deemed to have accepted certain specific terms, if those terms are contained in the accompanying license agreement, making the person a licensee of the software publisher.

GROWTH POSITIONS: $25,000-$50,000 +. 30 + years National Coverage. Software/systems design, application, project, Q.C./Test/Reliability, graphics, CAD/CAM, CNC, Aerospace/commercial. Fees Paid. Jim Longberry, Longberry Employment, 940 Main Place, P.O. Box 471, Niles, Ohio 44446. (216) 652-5871.

WANG INSTITUTE OF GRADUATE STUDIES
School of Information Technology
Software Engineering Professionals

Faculty applications are being accepted from software engineering professionals having extensive academic and industrial software design, implementation, and management experience. One-year visitors are also sought. Expected duties include teaching, closely interacting with students, supervising production-quality software development projects, and participating in curriculum development for the School's Master of Software Engineering degree program.

The Wang Institute is located near the heart of Boston's high tech corridor. It offers a unique environment for academic and professional growth. Instructional computing facilities include a VAX 11/780, a Prime 750, a Wang VS 100, Apollo DOMAINS, color graphics, and local network.

Qualified applicants should submit the names of three references and resume stating professional speciality areas and past software development or management efforts to: Faculty Search Committee, Wang Institute of Graduate Studies, Tyng Road, Tyngsboro, Massachusetts 01879.

April 1984