APRIL 1980

THE TECHNICAL CHALLENGE (p. 3) “Your most pressing professional challenge today as a computer practitioner, scientist, or engineer is to maintain your technical competence in this world of creeping obsolescence. Conservative estimates indicate that the technical decay rate is about 20 percent per year following college graduation. To meet this need, the Computer Society has for many years supported the activities of technical committees; they are now organized into two technical interest councils under the responsibility of two vice-presidents.”

PERFORMANCE ANALYSIS (p. 9) “An unfortunate percentage of the literature on performance modeling of complex systems satisfies Rosanoff's definition of durable nonsense: 'An alloy of sense and irrelevancy, protected with a thick coating of rigor and abstraction, prepackaged in a convenient black box, produces the most durable nonsense known to man.' … Fortunately, performance modeling has now developed far beyond ‘rigorous analysis based on inapplicable assumptions.’ In the past two or three years, the use of analytical performance models instead of simulation models has become much more popular.”

QUEUEING MODELS (p. 25) “Queueing networks are important as performance models of computer systems because the performance of these systems is usually principally affected by contention for resources.”

“Nearly all queueing networks used as computer system models and having feasible exact solutions can be represented as Markov processes. …”

QUEUE LENGTH DISTRIBUTIONS (p. 33) “The startling success of queueing network models [in the 1970s] caused a good deal of puzzlement among performance analysts. The derivations of the main results of stochastic queueing theory assume that the queueing network has time-invariant parameters, is in steady state, and has exponential distributions of service time at all FIFO—first in, first out devices. These assumptions are often seriously violated in practice. Yet, the models work.”

CONFIGURATION DESIGN (p. 47) “Computer system performance depends upon interactions among the software components, the hardware components, and the user workload. Understanding the nature of these interdependences requires collecting experimental data and developing a mechanism for analyzing system behavior as a function of component variations. … Accepted analytic methods for performance analysis, given the workload parameters and the system configuration, have been established. However, systematic methods for system synthesis which enable system parameters to be derived from design requirements are not well developed. …”

TEACHING PROGRAMMING (p. 58) “… Research into programming language learning similar to the systematic research done in foreign language learning does not exist. Most of the literature consists of the authors’ opinions as to the most effective methods of instruction for given programming languages, backed up, for the most part, by teaching or industry experience, intuition, perceived trends, or a combination of these factors. Therefore, the approaches described here need to be empirically tested, in a wide range of experimental settings, to objectively determine their instructional effectiveness.”

DATA (p. 67) “Although programmers are familiar with the concepts of data structures, data types, and data abstractions, they may not immediately recognize the relationships among these concepts or be able to look at them in a unified way. The objective of this tutorial—clearly defining these concepts and clarifying their interrelationships—relates directly to software reliability, since the problems of data structures and representations are more severe than those of program structuring. …”

AIRBORNE TERMINALS (p. 79) “One solution to the air traffic control problems at major airports would be the installation of computer terminals in passenger planes. Airborne terminals would eliminate the need for all but one human traffic controller at any airport. And they would provide a new line of passenger services.”

COMPCON 80 SPRING (p. 96) “Designing the Z-8000 16-bit microprocessor, by traditional, largely manual methods, took more than 13,000 man-hours over a period of three years, Rex Rice pointed out in his tutorial, ‘VLSI from the User’s Perspective.’ … Rice noted that Gordon Moore of Intel has generalized the design complexity problem in a new law: effort required for VLSI product definition, design, and layout is doubling every two and two-thirds years.”

INTERNATIONAL RELATIONS (p. 102) “The IEEE has suspended its more than 20-year-old technical exchange program with the leading electrical engineering society of the Soviet Union. In another action, the IEEE, jointly with the Association for Computing Machinery and the American Mathematical Society, has issued a statement protesting recent actions of the Soviet government against physicist Andrei Sakharov.”
APRIL 1996

INNOVATION (p. 7) “Are there limitations to living in real time? There sure are. In fact, the most dangerous limits are the tricky limitations of innovation—the juice that keeps Siliwood going. (Siliwood isn’t a place; it’s a state of mind combining the innovation of Silicon Valley and the trendiness of Hollywood.) Innovative products like PDAs have been a big disappointment because they sputtered right past the 2-3 year time limit in which a computer product is expected to flood consumerland. …”

OPEN SOFTWARE (p. 12) “X/Open Co. and the Open Software Foundation (OSF), the two leading open systems consortiums, have joined forces to become The Open Group, they announced at UniForum 96, held February 12-16 in San Francisco.

“X/Open and the OSF will work together to develop open systems specifications and deliver specification-compliant technologies, while maintaining their individual identities for work in other areas.”

FORMAL METHODS (p. 16) “The clear advantages of a more mathematical approach to software design has certainly been well documented; the literature contains many excellent examples of applications of formal methods for large, critical, or even business transaction systems. Despite the evidence, however, a large percentage of practitioners see formal methods as irrelevant to their daily work.”

FOR FORMALITY (p. 18) “The ‘stick’ of standards and the ‘carrot’ of education, supported by industrial-strength tools, will make or break the significant industrial use of formal methods. The market sector with the greatest potential to combine these elements most effectively is probably safety-critical systems. This is an increasingly important area for computer-based systems because of the flexibility that software provides. …”

AGAINST FORMALITY (p. 19) “There is a serious problem with formal methods in the software profession. The problem goes well beyond the fact that academics engage in wishful thinking about their value, and practitioners engage in skepticism and resistance to their use. The problem is lodged in what many are calling a chasm that exists between software in academe and software in industry.”

DEPICTING PROGRAMS (p. 33) “Software visualization can help software engineers cope with … complexity while increasing programmer productivity. Software is intangible, having no physical shape or size. … Software visualization tools use graphical techniques to make software visible by displaying programs, program artifacts, and program behavior. The essential idea is that visual representations can help make understanding software easier. Pictures of the software can help slow knowledge decay by helping project members remember—and new members discover—how the code works.”

CARTOONING COMPUTERS (p. 55) “Despite Deep Blue’s loss to chess champion Garry Kasparov, many people think computers are displacing humans, and this makes them nervous. Humor is one way of coping.”

PREDICTING 2012 (p. 70) “In the year 2012, megacomputers with distributed operating systems will manage thousands of processors connected via high-speed networks. Such ‘web servers’ will be geographically dispersed, hence the designation ‘global megacomputer.’ Megacomputers avoid the performance bottlenecks forecast for single-processor systems, since they avoid the limitations proposed by Amdahl and Gustafson-Barsis.”

UNIVERSAL SERVICE (p. 74) “One observer, Eli Noam of Columbia University, has predicted that the National Information Infrastructure (NII) will bring about the demise of American universities by removing physical location as a determining factor in learning. Whether this grim prediction comes true or not, it is reasonable to assume that higher education will be vastly changed by these new technologies. The Internet will most likely become the principal vehicle for access to instruction, research collaboration, information resources, and publication.”

INTERNET COMMERCE (p. 91) “For several years, electronic commerce has been a much-touted goal for the Internet. Not long ago the Internet was used primarily for research, with access limited mainly to the university and research communities. But network service providers and commercial on-line services have opened it to a much larger user community. … The Internet is increasingly becoming a venue for conducting the commerce previously carried out by other means, and for offering new commercial on-line services.”

STANDARDS DEVELOPMENT (p. 96) “The IEEE standards development process has one major problem—it’s length. The process has been basically the same for decades, even though the way business and the economy operate has changed drastically during that time. Creating a standard typically takes three or four years, and it can take longer. This is simply inadequate in this era of high technology and rapid change.”

PDFs of the articles and departments from Computer’s April 1980 and 1996 issues are available through the IEEE Computer Society’s website: www.computer.org/computer.

Editor: Neville Holmes; holmeswn@yahoo.com.au