JANUARY 1967

Chairman’s Letter (inside cover) “Technical Committees are organizational units of the Group established to provide leadership in developing those technical areas of interest to the Group’s membership.”

First Annual IEEE Computer Conference, 6–8 Sept. 1967 (p. 10) “Papers describing original research in the following areas (or related topics) are invited: design automation, new computer elements and new computer systems organizations, reliability, pattern recognition, and online computer systems and their applications.”

Trends in Pattern Recognition (pp. 16, 19) “A Report on the 1966 IEEE Pattern Recognition Workshop ... Self-Organizing, Bionic, Heuristically Programmed, Pattern Recognizing, Learning, Neuronal, Cybernetic, Goal-Seeking, Problem-Solving, Microprogrammed, Multiprogrammed, Multi-input, Redundant, Adaptive, Self-Repairing, Self-Teaching, Time-Sharing, Self-Reproducing, Cluster-Seeking, Online, Trainable, Stochastic, Kilomegacycle, Optimal, Artificially Intelligent, Symbiotic Computing Machines—was one speaker’s list of the key words necessary to describe the range of topics discussed at the recent ‘happening’ (the chairman’s characterization) instigated in Puerto Rico by the Pattern Recognition Subcommittee of the IEEE Computer Group. ... This diversity of interests is particularly apparent to anyone attempting to organize a university course, graduate or undergraduate, covering the general area of pattern recognition. A quick survey of existing courses reveals little agreement as to the basic topics to be included in such a course. Thus, perhaps fortunately, the next generation of pattern recognizers is likely to be as heterogeneous a group as the present set.”

JANUARY 1992

www.computer.org/csdl/mags/co/1992/01/index.html

Modeling the System (p. 11) “To model systems, we need an underlying set of fundamental concepts and notions—some call them ‘abstractions’—that, in [Frederick] Brooks’ terminology, capture the ‘conceptual construct’ of complex systems.”

Model Execution (p. 15) “One of the most interesting notions to come out of recent work in systems engineering is that of executable specifications or, to fit in better with the terminology used here, executable models.”

3D Processor (p. 25) “The 3DP, for 3-Dimensional Processor, is a parallel-computing architecture that targets these problems. It includes a hardware and software design that gives users an intuitive 3D object-oriented programming environment. It uses a C++ optimizing compiler to create assembly instructions that exploit underlying hardware capabilities for parallel processing.”

Aquarius Prolog Compiler (p. 54) “Commercial Prolog systems are robust and useful for solving many problems, but they are often an order of magnitude slower than systems implemented with popular imperative languages such as C. The goal of our work is to reduce this performance gap. Our hypothesis is that Prolog can be implemented as efficiently as an imperative language by compiling the more powerful features of logic programming only where they are needed, and then only in the simplest form.”

Automatic Test Pattern Generation (p. 71) “The efficiency of algorithms for automatic test pattern generation [ATPG] has not kept pace with increasing circuit size. Mapping the problem to parallel-processing machines might improve performance. ... An efficient combined method for solving the ATPG problem uses statistical methods to find tests for the easy-to-detect faults on the fault list and switches to an algorithmic method to find tests for the remaining hard-to-detect faults.”

High-Performance Computing (p. 87) “High-performance computing will remain the centerpiece of the administration’s efforts in science and technology,” said D. Allan Bromley, assistant to President George Bush for science and
technology and director of the US Office of Science and Technology Policy. Bromley delivered the keynote address to an audience of several thousand on the opening day of Supercomputing 91.”

**Windows Programming Environments** (p. 97) “The past year has seen an explosion of Microsoft Windows 3.0 programming environments. In fact, we have six of them for review. The six fall into four distinct groups. The programming languages for ObjectScript and Realizer are similar to Basic. Actor and Smalltalk/V are object-oriented environments. Borland C++ is also object-oriented but looks and feels like C. KnowledgePro comes from an AI environment and reminds me a little of Lisp. While all these environments let me write Windows applications, you will see that some made it easier than others.”

**Massive Show of Supercomputing** (p. 105) “New Products: Cray Y-MP C90, Paragon XP/S, Meteor Multibus II, CAMPUS/800, Entry Level C3400-ES, Zephyr ‘deskside.’”

**Book Review of Computers under Attack: Intruders, Worms, and Viruses, Peter J. Denning, ed.** (p. 134) “The siege is on! Computer systems today are increasingly subject to attack from a variety of sources. Networks and PCs are among the targets. Industry, government, academe, and private citizens are the victims. What are the threats? Who would perpetrate a crime against a machine? What can be done to minimize or prevent damage to valuable data and equipment? ... Readers less familiar with security issues will gain valuable insight into the workings of expansive computer networks. They will be amazed at how much territory these networks cover and dismayed at how many ways there are to break through network security features or disrupt computer system operations. They will discover the world of computer espionage, cyberpunks, worms, and viruses.”

**COMPUTER 50 YEARS AGO**

What a Non-programmer Should Know about Programming Languages (vol. 1, no. 4, 1967, pp. 1, 7, 9) “There are literally thousands of different languages now in use for communicating with digital computers. Each computer has its own set of acceptable languages—some are unique to a particular computer, others are common to many. But the vast majority of these computer languages are data languages, notations for expressing the data the computer processes. Still, there are hundreds of different programming languages—more, if you count all the dialects. ... Programming languages are complex phenomena; they have many vital and fascinating aspects and attributes. Four of the most important and commonly used yet controversial properties are machine-independence, procedural, generality, and usability. In a sense, these are four separate though related scales that we can use to measure a programming language, except we really know none of the units. ... 'What language shall we write our programs in?' To answer this question, you must consider both the language and its processor, and look at them in relationship to each of the other factors. So you want to ask: Is the language capable enough and general enough to handle your problems? Will your people find it easy to learn and use? How well has it been implemented for your computer, and will you be able to use it on your next computer? And finally, how well does it fit into your OS, and into your whole scheme of operations?”

**Figure A.** Images from Computer’s first year in publication, when it was called Computer Group News.