

50 & 25 YEARS AGO



EDITOR ERICH NEUHOLD
University of Vienna
erich.neuhold@univie.ac.at



FEBRUARY 1972

In the early years, *Computer* was only published bimonthly. Therefore, we will have to skip our interesting and/or informative extractions for February. The next one will appear in the March 2022 issue of *Computer*, and we hope you will eagerly wait for our next publication of this column.

FEBRUARY 1997

<https://www.computer.org/csdl/magazine/co/1997/02>

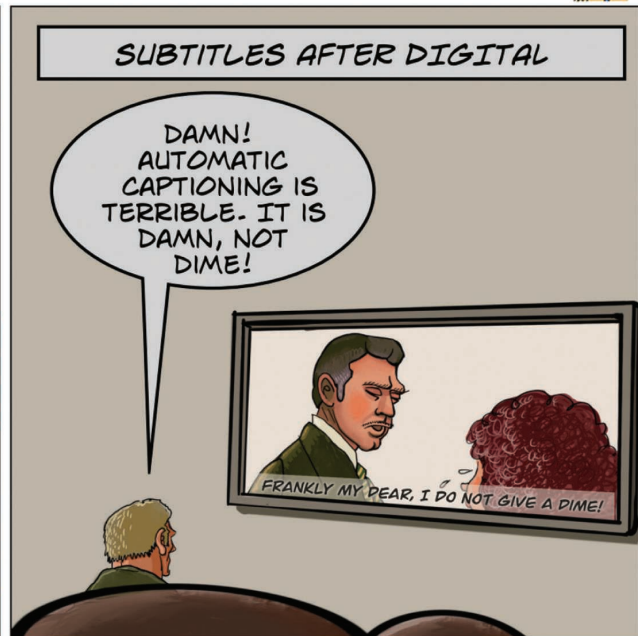
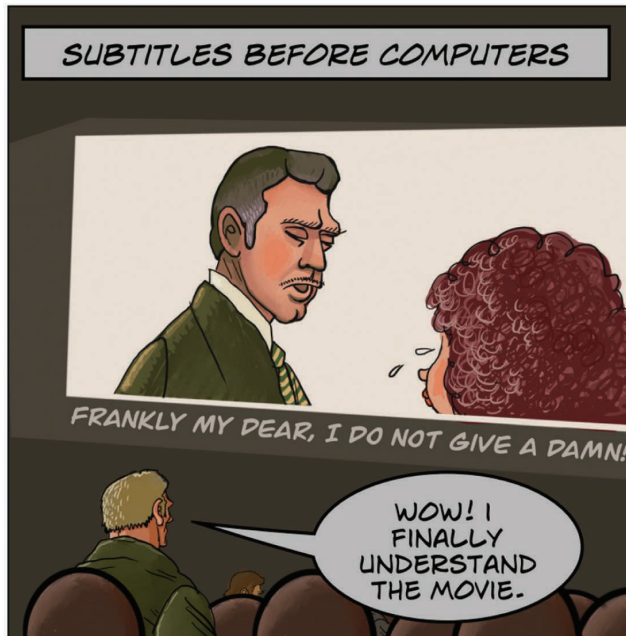
Breaking the Teraflops Barrier; David Clark (p. 12) “An important milestone in high-performance computing was achieved in December when a computer that Intel is designing for Sandia National Laboratory tested at more than 1 trillion floating-point operations per second. ... The US Department of Energy’s 10-year Accelerated Strategic Computing Initiative is designed to triple computing performance every 18 months. The triangles on each bar, which are separated by 18-month intervals, show delivery milestones for significantly more powerful supercomputers.” (p. 13) “The burst of supercomputing research could also benefit organizations and individuals that don’t use high-performance computers. Because most of the work is focused on increasing performance in systems that use multiple off-the-shelf microprocessors, the research could identify ways to generate more power in other multiprocessor systems, such as mainframes and eventually even PCs.” [Editor’s note: *The prediction/plan of the U.S. government has held pretty well, and the continued increase in speed has led us now to 415.530 peta-floating-point operations per second in Kobe, Japan—that is, a 415-billion-times increase since 1997.*]

Investigating the Influence of Formal Methods; Shari Lawrence Pfleeger et al. (p. 33) “Formal methods promise much, but can they deliver? In this project, results are inconclusive, but careful data gathering and analysis helped

establish influences on product quality. ... In this study, we investigated the effects of using formal methods to develop an air-traffic-control information system. Because we are studying one project in isolation, we cannot draw conclusions about the suitability of formal methods for all projects.” (p. 41) “On the one hand, we found no compelling quantitative evidence that formal design techniques alone produced code of higher quality than informal design techniques. The predelivery fault profile showed no difference between formally designed and informally designed code. On the other hand, the unit testing data showed fewer errors in formally designed code, and post-delivery failures were significantly less for formally designed code. Thus we can conclude that formal design, combined with other techniques, yielded highly reliable code.” [Editor’s note: *This very detailed analysis of using five different techniques (finite-state machine, Vienna development method [VDM], VDM/calculus communicating systems [CCS], total formal with proofs, and informal pseudocode) in the end did not show any significant advantages of using formal methods against using pseudocode for the design. Unfortunately, the effort needed and the expertise of the developers were not recorded as they may have had a significant influence on the quality of the results. An indication of this is the fact that the best method was VDM/CCS, but that technique was done by a single person, whereas all of the others had a team involved.*]

Trends in Mobile Satellite Technology; Gary Comparetto et al. (p. 44) “Demand for sophisticated personal communication services has changed communications satellite design. Satellites have moved closer to the Earth to improve communication speed and enable personal communication services. But in so doing, they require more computing resources and more sophisticated protocols to handle intersatellite communications.” (p. 50) “There are two predominant protocol reference models for data communications: the OSI (Open Systems Interconnection) and Internet (which used to be called the DoD Reference Model). And there are two corresponding protocol suites: the OSI and Internet protocol suites. The standards in both of these suites are based on a set of assumptions

COMPUTING THROUGH TIME



SUBTITLES FIRST APPEARED IN MOVIES IN 1903, IN EDWIN S. PORTER'S *UNCLE TOM'S CABIN*. THEY WERE WRITTEN SIGNS THAT WERE FILMED AND PLACED BETWEEN SEQUENCES IN THE MOVIES. IN 1909, M.N. TOPP RECEIVED A PATENT TO RAPIDLY SHOW TITLES FOR MOVIES. THESE ORIGINAL SUBTITLES ARE ALSO KNOWN AS OPEN CAPTIONS. THE TERM "CLOSED CAPTIONS" MEANS SUBTITLES ARE NOT VISIBLE UNLESS ACTIVATED BY THE VIEWER, AND IT IS VERY HELPFUL FOR VIEWERS WHO HAVE HEARING PROBLEMS.

Digital Object Identifier 10.1109/MC.2021.3132709
Date of current version: 14 February 2022

about available resources and environment conditions that do not apply to space communications. ... The SCPS (ed. Space Communication Protocol Specifications) project evaluated the functionality in each existing standard against the requirements" [Editor's note: This very detailed article concentrates on low-Earth orbit satellite clusters and tries to handle the ever-changing configurations of such nets. Interestingly enough, it does not even mention similar requirements that arise with mobile devices. But they just started to appear 25 years ago.]

Global Teleporting With Java: Toward Ubiquitous Personalized Computing; Kenneth R. Wood et al. (p. 53) "The concept of ubiquitous personalization extends ubiquitous computing by tailoring the computing and/or control interface to the user's specifications, wherever the user might be and whatever device he or she might be using. ... However, at the Olivetti and Oracle Research Laboratory, we have introduced another form of mobility in which it is the user's applications that are mobile. Users do not carry any computing platform but instead bring up their applications on any nearby machine exactly as they appeared when last invoked. We call this form of mobility teleporting, and it has been used continuously and fruitfully by many members of our laboratory for the past three years." [Editor's note: The article then continues to describe the system the authors built. It relies heavily on the wide availability of Java and, of course, the Internet. However, as it does

not even mention security and privacy concerns, it is no wonder it did not catch on. Only 20 years later, the cloud technology moves in this envisioned ubiquitous computing scheme, but even today the solutions are plagued by privacy and security concerns.]

Trusting Mobile User Devices and Security Modules; Andreas Pfützmann et al. (p. 61) "The market for devices like mobile phones, multifunctional watches, and personal digital assistants is growing rapidly. Most of these mobile user devices need security for their prospective electronic commerce applications." (p. 67) "No mobile user devices currently available are tamper resistant in all their parts, so it is reasonable to supplement less secure devices available from many manufacturers with security modules. Providers can concentrate on developing protocols for security applications that can run in any device, from mobile phones to personal digital assistants. ... If an undercover agent and a personal agent are separate parts (possibly removable) of a mobile user device, then the manufacturers must provide them with protocols for interaction. Actions critical for both the user and the mandator, such as spending electronic cash, are valid only when authorized by both agents." [Editor's note: An interesting article that analyzes well the different security threats for mobile devices. However, this was 1997, and the focus was an attached and removable security gadget, which, as we now know, existed only for a short time. Security is now handled inside the devices,

but, as we also know, only more or less successfully. With the Internet of Things, problems are getting even worse.]

Customizing System Software Using OO Frameworks; Nayeem Islam (p. 69) “Today’s applications have exploded in their diversity, but most operating systems are still general-purpose and inefficient. One of the benefits of using an OO approach is the ability to modify very small details of an operating system, which makes it easy to tailor the system to the application.” (p. 71) “I illustrate my approach with parallel applications that are structured as concurrent processes, communicating with message passing. The performance for this kind of application depends on how the application uses the message-passing interface. ... A message-passing system should support different message-passing primitives, such as synchronous and asynchronous messages, multicasts, and broadcasts. ... One of the distinguishing features of an OO design is the granularity at which customizations can be made. ... My approach allows the incremental addition of a new type of communication port.” [Editor’s note: Interesting as it sounds, the article does not really explain how much detailed problem knowledge is required to select the optimal object-oriented modules for a specific task; for example, here it is communication between concurrent processes.]

Neural and Fuzzy Methods in Handwriting Recognition; Paul D. Gader et al. (p. 79) “Handwriting recognition requires tools and techniques that recognize complex character patterns and represent imprecise, commonsense knowledge about the general appearance of characters, words, and phrases.” (p. 80) “We experimented with a variety of novel techniques for applying neural and fuzzy methods to the hand-writing recognition problem. They are as follows: • We assigned character confidence values to individual segments using fuzzy sets and multilayer feed-forward neural networks (MLFNs). • We gave the segments contextual information by assigning a confidence that character pairs were spatially compatible, again using MLFNs. • We aggregated character confidence values from a segmentation, by means of fuzzy integrals.” (p. 85) “We have not yet considered the application of fuzzy and neural methods at levels higher than the word or numeric field levels. Commercial handwriting recognition systems are required to process entire documents, not just individual words.” [Editor’s note: The article is interesting as it combines MLFN and fuzzy set technology to decide among close alternatives. However, restricting itself to characters and single words, it cannot include the “meaning” of a word within its context. Only those new approaches, relying on much more than dictionaries, have brought the progress that we see today.]

Linking Computers and Consumer Electronics; Charles Severance (p. 119) “Enter the IEEE 1394 standard (ed. 1995) known as Firewire. This standard proposes to provide a single port on the back of our computers that can handle nearly all of

the communication for which we now need eight to 10 ports.” (p. 120) “With 1394 technology a user can develop an ‘entertainment center area network’ that integrates all residential video, audio, computer, and data communications equipment. ... The closest technology is called Universal Serial Bus from Intel. Unlike 1394, USB is not designed for high speed and is better suited for connecting keyboards and mice.” [Editor’s note: Of course, the USB standard of 1996 mentioned and discounted in this article took over the world and pushed Firewire into niche application.]

Toward Synergistic Engineering of Computer Systems; Jerzy W. Rozenblit et al. (p. 126) “Today it is increasingly important that a system’s hardware, software, interfaces, and related functions be developed synergistically—with an understanding that every system element will influence other elements.” (p. 127) “Ideally, an integrated codesign environment supports hardware/software evaluation throughout the development process using models with various levels of detail. Appropriate abstractions are crucial. Abstractions let us focus on aspects of interest and thus manage complexity.” [Editor’s note: This interesting article suggests the use of multifaceted models in complex system designs. The Unified Modeling Language and others offer similar approaches. It ignores in this synergistic approach, however, the very important aspect of human interface and usability design. Unfortunately, “modern” techniques like Power Programming have moved away from careful and synergistic engineering, relying, in my opinion, more on the trial-and-error development cycle.]

Conducting a Job Search on the Internet; Dwaine Maltais (p. 131) “Now that online career services are more prevalent on the Internet, the armchair job search is a viable alternative. In fact, job hunting is probably one of the best Internet applications to date, since it promises to bring candidates and companies together worldwide. ... There are many online services, each catering to its client industries. The sidebar “Selected Job-Search Web Sites” lists some of the major services. Before submitting your resume, it is a good idea to visit each online service and review its client list. This helps you target companies specific to the industry that best meets your career goals.” (p. 133) “At first glance the Internet seems to be the perfect medium for centralizing the global job market, and online career services look like the most efficient way to search for a job. However, you should not have to compromise your right to privacy or fair hiring practices to gain increased visibility. Through careful planning, candidates can ensure maximum security and privacy. We can only hope that future technological advances do not make it more difficult for the individual to retain these rights.” [Editor’s note: The article analyzes extensively the various issues of online job searches, whether they involve resumes, email, videos, or online video meetings. Unfortunately, the warning expressed in the last paragraph became true, and the Internet offers many opportunities to the recruiter to invade the privacy of the job seeker.] ■