AUGUST 1968
In its early years, *Computer* was published bimonthly. Stay tuned for more interesting historical highlights in the upcoming September 2018 issue.

AUGUST 1993
www.computer.org/csdl/mags/co/1993/08/index.html

Computer Society Treasurer’s Message—1992 Financial Results (p. 8) “Recognizing the need to build up the Computer Society’s financial reserves, the Board of Governors in 1992 established demanding surplus targets for 1993 and 1994. ... The result was a welcome improvement in our financial picture: The society ended 1992 with a $515,300 surplus compared to a $490,800 loss the year before.” [Editor’s note: This detailed financial statement explains the type of actions that led to financial improvements. It’s clearly worth reading to get ideas on what to do regarding today’s similar financial problems.]

Guest Editors’ Introduction: Extending Telecommunications Systems—The Feature-Interaction Problem (p. 14) “One day not far in the future, you will be struggling to compose a technical report on your home computer. You set up a conference with several colleagues, who are your coauthors. During the conference, you converse with your colleagues while seeing them on screen. They also see an image of your report on their workstations. You have begun to make some progress on the report when your working session is suddenly suspended and replaced by an emergency call from the school nurse; your daughter has been struck in the abdomen by a hockey stick. While you talk with the nurse, a call from a telemarketing firm is routed to your voicemail system. When you resume the session with your colleagues, you explain the situation and quickly arrange for the remainder of their discussion to be recorded for your later review. You then call your spouse’s personal communications number and are routed through in the midst of a sales meeting at a client’s business. You relay what has happened and you both agree to meet at the hospital emergency room. This scenario is feasible because each of these new telephone features exists today. However, their appearance in the telephone system as a group of compatible features might take much longer than expected. The problem? Every new feature can change the behavior of pre-existing features or even ‘break’ them, crashing the system. In the telecommunications industry, we call this the feature-interaction problem. It is a key obstacle to rapidly adding new features to the public network and evolving it to meet user needs. ... ” [Editor’s note: Overall, this editorial and the accompanying articles are very focused on telecommunications and ignore events in other complex systems. Also interesting to note is that although the Internet existed in 1993, it wasn’t mentioned in any of the examples cited.]

Feature Interactions and Formal Specifications in Telecommunications (p. 20) “This tutorial is aimed at computing scientists and engineers who are interested in the telecommunications domain. It explains the feature-interaction problem and explores the relationship between feature interactions and formal specifications. The article assumes some acquaintance with commonly known techniques for formal specification. ... I also show how many feature interactions can be eliminated by applying known techniques of formal specification (along with a suitable amount of foresight). But I also tell the other side of the story: feature interactions that current specification techniques cannot handle satisfactorily. I hope that these examples inspire new ways of looking at systems and new techniques for formal specification. ... What is a call? ‘Call’ might be the most commonly used term in telecommunications, but these days we must be careful not to make unwarranted assumptions about it. For example, no individual party is indispensable to a call. Features such as conference, transfer, and drop let users add and delete parties to an ongoing call, so that the conversation can continue while its membership changes completely. The ultimate in dynamic calls is the chat line, an ongoing public call that...
Improving Public Switched Network Security in an Open Environment (p. 32)

“Recognizing that the public switched network is increasingly vulnerable to hostile users, the federal government has stepped up efforts to maintain the integrity of telecommunications services. … A second NSTAC [National Security Telecommunications Advisory Committee] network security report, released in 1992, contained an updated risk assessment confirming that malicious hackers continue to penetrate PSN (public switched network) systems and intrude upon sensitive government interests. Outages experienced by telecommunications providers have also helped to focus the federal government’s attention on the need to ensure dependable communications. … Working with the National Communications System (NCS), the National Institute of Standards and Technology (NIST) has identified some additional concerns: Feature interactions could disrupt a needed service or be targeted for intentional abuse by malicious users. The potential for undesirable feature interactions will increase as more services are added to the network.” [Editor’s note: A large amount of work has been done since 1993; however, as the reoccurrence of intrusions and malicious malfunctions has shown, more remains to be done to satisfy reliability, security, and privacy concerns in such systems.]

Distributed Artificial Intelligence [DAI] for Runtime Feature-Interaction Resolution (p. 48) “DAI extends distributed processing so that the computers reason about when to interact with other systems and what information to exchange. Interactions are then more flexible to reflect particular, possibly changing, requirements for a specific system and changes in a system’s environment, such as the emergence of other systems. … The DAI literature includes a number of approaches that address the feature-interaction problem. Here I describe four. LODES [Large-Internet-work Observation and Diagnostic Expert System], TEAM-CPS, and Multistage Negotiation primarily address interactions of different administrative domains. An agent encapsulates the network management system in one administrative domain. TEAM-CPS, Multistage Negotiation, and Negotiating Agents address resource contention in cases of limited network support. The Negotiating Agents mechanism also deals with personalized instantiations. … Constraints of the application domain. The agents’ tasks are complicated by additional constraints on the problem-solving process: Telecommunications systems are dynamic; they do not stop during problem solving and they have strict availability and response requirements. Cooperative problem solving in such an environment must therefore be fast and reliable, without consuming too much processing power and communication bandwidth. … Applying DAI approaches to feature interaction also raises interesting research problems. Systems need evaluation functions to rate generated proposals, possible but not-yet-generated proposals, and constraints. Moreover, different problem-solving processes might interact and interfere with each other.”

Formal Models of Communication Services: A Case Study (p. 37) “Formal methods can play an important role in exploring these new services. The telecommunications and data communications communities have long accepted the need for formally describing protocols, but only recently have they considered formally describing a service by abstracting specifications from a particular protocol that provides that service. Specifying a service at an abstract level meets two important needs: standardization and customization. … To illustrate the uses of the formal model, I also describe a simple protocol for providing an atomic multicast service. The example is not realistic in that it ignores all sorts of failures, but it is short and does demonstrate the model’s features. The protocol provides one service by making use of a lower level, simpler service. In this example, I use a service that reliably passes data from any node to any one neighbor, preserving order between each pair of nodes.”
InterBase: An Execution Environment for Heterogeneous Software Systems (p. 57) “InterBase integrates pre-existing systems over a distributed, autonomous, and heterogeneous environment via a tool-based interface. It supports heterogeneous applications without violating the local autonomy of component systems. ... As the InterBase focal point, the distributed Flex transaction manager (DFTM) interprets and coordinates the reliable execution of global transactions over the entire system. It provides a unified and flexible interface, the IPL language, with which system programmers can specify the data and control flow of a global transaction. The Remote system interfaces (RSIs) are specially designed InterBase agents that are superimposed on individual local systems. RSIs provide a uniform interface that buffers the heterogeneity of local systems, thus relieving the DFTM from dealing with each local system directly.” [Editor’s note: This architecture reflects the five-level architecture that has been widely used to deal with the interoperation of heterogeneous distributed database management systems. This article is especially interesting because it covers installed systems and, therefore, discusses all the issues that arise when moving outside of a research/prototype environment.]

Finalists Shatter Records in 1993 Gordon Bell Prize Competition (p. 73) “The performance reported by the top entry was over eight times that of the previous best, and two entries exceeded 60 billion floating-point operations per second. More significantly, 12 entries achieved sustained rates exceeding 25 percent of the theoretical peak performance of the machines they ran on. Price/performance improved by almost 600 percent to 7.5 Gflops per million dollars.” [Editor’s note: Compare this to the performance of the current world leader Summit: 200 Petaflops and about 3 million times as fast at a cost of $200 million, or $1 per Gflop.]

From the analytical engine to the supercomputer, from Pascal to von Neumann, from punched cards to CD-ROMs—IIEEE Annals of the History of Computing covers the breadth of computer history. The quarterly publication is an active center for the collection and dissemination of information on historical projects and organizations, oral history activities, and international conferences.

www.computer.org/annals