National Academy Looks at Computing's Future

The United States is doing well, but the Japanese are gaining; perhaps we should accept some Grand Challenges

For its first official report since being formed in mid-1986, the National Academy of Sciences' Computer Science and Technology Board has just released a broad survey that assesses the health of the U.S. computer enterprise, and identifies the most promising areas for further research.*

The 18-member board, chaired by Columbia University's Joseph F. Traub, found much to like: the American "Innovation Engine"—a combination of university basic research, high-technology venture capitalists, and mature corporations—is generally working well. In computers, at least, the United States still leads the world.

The board also found much to be concerned about: the Japanese are catching up.

And finally, the board made two broad recommendations. First, it called for the expansion of the current hodge-podge of ad hoc, balky computer networks into a truly national networking system, which would allow any computer to communicate with any other computer "easily, reliably, and over a broad range of speeds commensurate with individual application needs."

Such a network would facilitate commerce, manufacturing, and research in much the same way that the interstate highway system has facilitated transportation. Indeed, momentum seems to be building: the general idea has already been endorsed at the White House by the Office of Science and Technology Policy, and in Congress by Senator Albert Gore (D–TN), who has announced that he will soon submit a bill to establish the network.

However, the board hastened to point out that a national network need not be federally owned or even federally funded. Indeed, some of the essential pieces, such as high-data rate fiber optic lines, are already being put into place by private carriers such as AT&T and MCI. To be effective, however, the network will require some form of overall coordination, particularly in insuring that computers on the network can all talk to each other using common data protocols.

* *The National Challenge in Computer Science and Technology* (National Academy Press, Washington, DC, 1988).

The board's second broad recommendation is that the federal funding agencies—primarily the National Science Foundation and the Defense Research Advanced Projects Agency in this case—should increase their support for basic computer science and technology research. Indeed, the board pointed to some particularly promising areas in multiprocessor hardware design, software development, artificial intelligence, and theoretical computer science.

One way of organizing the research effort might be to fund a number of high-profile "Grand Challenges."

In addition, however, the board suggested that one way of organizing this enhanced basic research effort might be to fund a number of high-profile "Grand Challenges," which would function as technology drivers in multiple areas simultaneously. To illustrate what it has in mind, the board gave six examples:

■ **Technology for large, correct software systems.** The development and maintenance of software is now *the* major cost of computation. Programmer productivity has improved only incrementally over the years. Development schedules and budgets are often little better than guesses. And it is exceedingly difficult to know whether a given piece of software is correct when it is released—even though (or because) dozens of programmers have worked on it. This challenge calls for tools and methods that would allow for the development of large software systems that are knowably correct, that have a predictable cost, and that can be easily modified and upgraded.

■ **An ultrareliable computer system.** The challenge here is to devise a computer system that would run for 20 years or more without failing. Such a device would be particularly useful in space and in other hazardous environments. Since no piece of hardware is ever perfect, of course, ultrareliability means that redundancy would have to be built in at the microchip level. Moreover, the computer's operating system would need to use advanced artificial intelligence to detect faults as they occur, and to reconfigure the machine's memories and data paths to work around the faults.

■ **A trillion-operations-per-second ultracomputer.** Such a system would be at least a thousand times faster than any existing supercomputer. It would require major advances in microelectronics—with, say, a billion bits of memory on a chip—together with equally major advances in computer architectures that harness hundreds or thousands of processors in parallel.

■ **A translating telephone.** This challenge calls for development of a telephone that would allow two people, speaking different languages, to converse directly. Such a system would require a massive effort in artificial intelligence, since it would involve a speech recognition system capable of speaker-independent recognition of large-vocabulary, continuous speech; a natural-sounding speech synthesis system that preserves speaker characteristics; and a machine translation system capable of dealing with all the vagaries of human language, including ambiguity, nongrammatical phrases, and incomplete phrases. The Japanese have already accepted the challenge: they have recently initiated a 7-year, $120-million project to build a telephone that will translate from Japanese to English.

■ **Computers and robots that learn.** Much of the burden of programming could be eliminated if machines could learn the way humans do: from examples, from observation, from books, and from practice. Two prototype challenges might be to build a computer that could read, say, a chapter from a college physics text and then answer the questions at the end; and to build a robot that could assemble, say, a food processor after watching someone else do it. These are both extremely hard problems that would require major advances in robotics, computer vision, natural language understanding, and machine learning.

■ **Self-replicating systems.** This is a problem that might be of practical interest if we ever wanted to exploit raw materials on the moon or in the asteroid belt: instead of launching a whole factory, launch a handful of robotic machine tools equipped with a "genetic code" that would allow them to create the rest of the factory on the spot, out of local materials. In any case, this challenge clearly calls for major advances in robotics, artificial intelligence, and what is often called "design for manufacturability."

■ M. MITCHELL WALDROP