2008

Remembering Jim

Ross, Michael

news

DOI:10.1145/1378704.1380421 Michael Ross

Remembering Jim

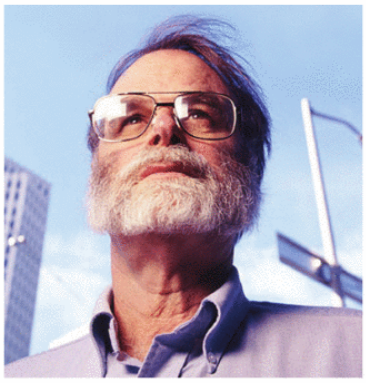
Both melancholy and reverential, the Jim Gray Tribute at the University of California at Berkeley honored one of computer science's leading pioneers and visionaries.

ON MAY 31, almost 750 colleagues, friends, and family members gathered at the University of California at Berkeley campus for a day-long tribute to Jim Gray, who disappeared at sea while sailing his 40-foot sailboat off the coast of San Francisco on Jan. 28, 2007.

For nearly four decades, Gray worked for some of the computer industry's largest companies including Digital Equipment, IBM, Microsoft, and Tandem. His intellect and technical achievements in database and transaction processing are legendary. He won the ACM A.M. Turing Award in 1998 for his description of the basic requirements for transactions as well as his research in locking, concurrency, and fault tolerance. He initiated the use of performance benchmarks for a wide variety of transaction environments. And in recent years he stimulated the creation of massive distributed scientific databases that are reshaping the fields of astronomy and oceanography, and are laying the groundwork for eScience, a new method of research.

Even more impressive than his professional achievements is the fact that hundreds of Gray's colleagues considered themselves to his best friend, so genuine and deep was his interest in each of them. Accordingly, the tribute's audience included graduate students and new employees, astronomers and geologists, department chairs and CEOs, and several generations of computer scientists and database experts from academia and industry. A handful of attendees traveled from as far away as Europe and Asia solely for the event.

Gray entered the University of California at Berkeley in 1961, initially majoring in physics. He briefly considered philosophy, then switched to mathematics before settling on the emerging field of computer science. His thesis advisor, Michael Harrison, urged his



students to write down all that they learned. Gray took this advice to heart, developing the habit of writing up—and distributing—reports of meetings, trips, and conferences. He soon established two rules for authoring technical papers: "He who types the paper is first author" and "It's easier to add a co-author than deal with someone's hurt feelings." At times, colleagues were unaware of their participation. "I co-authored [one paper] while I wasn't looking," quipped Microsoft's Pat Helland, who joined Tandem in the early 1980s to work with Gray.

"Jim was not dangerous to his colleagues, as some scientists can be," recalled Mike Blasgen, who was one of Gray's managers at IBM in the 1970s. "He would not take credit for your ideas. Also, he always had an interesting or provocative insight, so people wanted to be his friend."

In time, Gray developed a large professional network with which he shared information, both inside and outside of the company he was working for. At Tandem, for example, "he was a great pollinator," productively sharing ideas between departments, said Wendy Bartlett, now with Hewlett-Packard.

When visiting companies and universities or attending conferences, Gray often sought out students, interns, and young professionals, listening intently to their research and subtly offering suggestions. "He would not say 'This is what you must do,' " said Alex Szalay of Johns Hopkins University, who had worked with Gray on the Sloan Digital Sky Survey. "He would gently light the way, so that people would find the path themselves."

Although Gray's closest colleagues knew firsthand the many hours he typically spent working, many attendees at the tribute were incredulous that any one person could accomplish all that Gray had.

His wife, Donna Lee Carnes, offered some clues. "I don't think I ever saw Jim procrastinate," she said. "Writing also came very easily to him. You can get a lot done when you're focused and fast. He could also read and absorb knowledge very quickly. "

Carnes said her husband had an astounding amount of energy despite the fact that he often slept only five hours a night during the week. "If he was looking for a bug in his program or a product was nearly ready to ship, nothing came between Jim and his work," Carnes added. "He wouldn't even stop to eat. Just coffee, sometimes three pots."

However, after the bug was found and fixed or the product shipped, Gray would enjoy good food, and often sailing, with his wife and friends. When John Nauman, who hired Gray at Tandem, asked the tribute audience how many had gone sailing with Gray, about 100 hands shot into the air.

In announcing his departure from IBM Research in 1980, Gray wrote that he aspired to be "a scholar of computer science," noting that all fields of scholarship emphasized research, teaching, and service. The Jim Gray Tribute demonstrated that he had clearly achieved that—and much, much more. C

Michael Ross writes about science and technology from San Jose, CA.

PHOTOGRAPH BY RICHARD MORGENSTEIN

viewpoints

DOI:10.1145/1378704.1378711

Peter J. Denning

The Profession of IT
Voices of Computing

The choir of engineers, mathematicians, and scientists who make up the bulk of our field better represents computing than the solo voice of the programmer.

Although enrollments in computing degree programs appear to have bottomed out at approximately half of their 2001 level, there is no reassuring upward trend. The industry need for computing professionals will continue to exceed the pipeline by at least one-third for some time to come. Why do low enrollment rates persist in such a good market?

Several key factors influencing low enrollment rates are connected to the myth "CS=programming"—tales of dwindling employment opportunities, negative images of computing work, and inflexible curricula.[2,3] Reversing this myth can result in considerable progress.

Thirty-five years ago, Edsger Dijkstra reacted to his generation's version of this myth by declaring himself proud to be a programmer.[5] Many followed his lead. ACM has been proud: half the A.M. Turing Award winners are in programming, algorithms, and complexity.[3] But our internal self-confidence did not dispel the external myth.

Twenty years ago, the ACM and IEEE warned that the myth could become damaging.[4] Today, the word "programming" itself generates misunderstandings. Internally, it is broad: design, development, testing, debugging, documentation, maintenance of software, analysis, and complexity of algorithms. Externally, it is narrow: the U.S. Bureau of Labor Statistics defines "programmer" to mean "coder." Often without realizing it, insiders and outsiders interpret the same words with entirely different meanings. When insiders broadened to object-oriented programming, outsiders thought we narrowed to the Java language.

Ten years ago, we tried another tack. We broadened our view of computing to include information technology (IT),[1] and we defined what it means to be fluent in IT.[7,8] These works were embraced in high schools and helped generate enrollments in IT but not CS. They have not dispelled the myth.

Today, Clay Shirky notes a trend that may help explain the durability of the myth.[6] The general public is now confronted with an amazing array of powerful tools for the common computational tasks. Many believe they can accomplish what they need as amateurs. Only a few professionals are needed to program all these tools for the many. There is no special attraction to being a professional.

How can we communicate the richness of our field and dispel the myth? What if we learn to speak in the voices of the many kinds of computing professionals? The programmer is a solo voice. The whole, loud choir could dispel the perception that the bulk of computing is about programming. The choir might also help make professionalism more attractive by showing our many critical specialties that cannot be done by amateurs.

To speak in a professional's voice, I immersed myself in the professional's practice. I spoke of war stories, experiences, ambitions, fears, and everyday things. I sang the joys (and sorrows) of being a professional.

Education philosophers such as John Dewey maintained there are two ways of learning, which can be called "learning-about" and "learning-to-be." Learning-about means to acquire a description; learning-to-be means to acquire the practice. Learning about carpentry, music, or programming is not the same as being a carpenter, a musician, or a programmer. Programming seems to blur this distinction because programmers build descriptions of al-