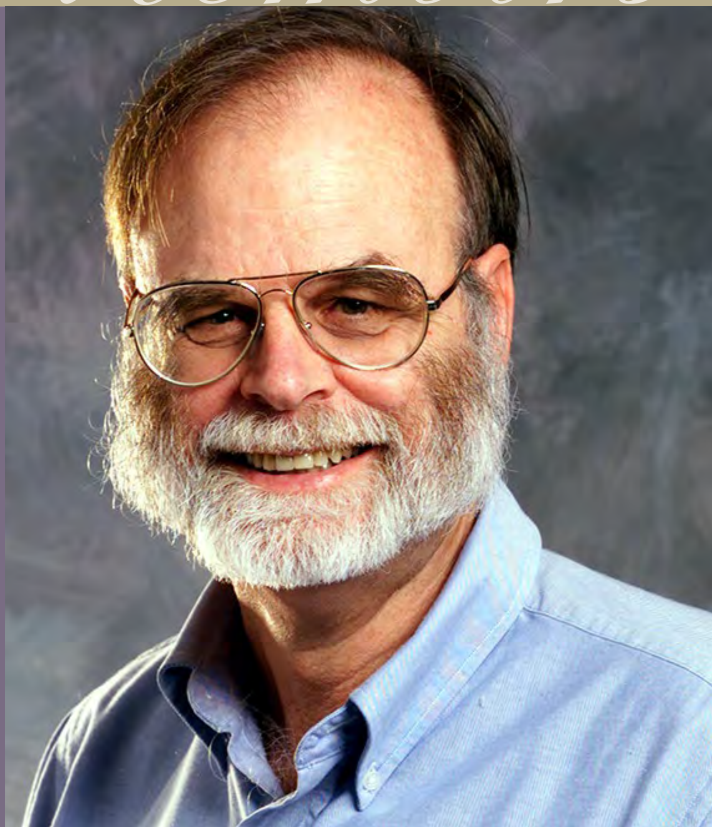
James N. Gray

1944-2012

BIOGRAPHICAL *Memoirs*



A Biographical Memoir by Gordon Bell, Leslie Lamport, and Butler W. Lampson

NATIONAL ACADEMY OF SCIENCES

JAMES N. **GRAY**

January 12, 1944–January 28, 2012 (Lost at sea 1/28/2007)

Elected to the NAS, 2001

Photograph courtesy Microsoft.

By Gordon Bell, Leslie Lamport, and Butler W. Lampson

James Nicholas ("Jim") Gray was born January 12, 1944, in San Francisco, California, and lost at sea January 28, 2007, during a trip to cast his mother's ashes at the Farallon Islands near San Francisco. The enormity of the loss to his personal friends and to computer science—especially the database community that he helped create and lead—was quickly apparent. The U.S. Coast Guard immediately began an extensive sea search, and hundreds of friends searched both directly and by examining a vast array of satellite image data. Even in his loss, Jim was helping to innovate—in this final case, advancing lost-at-sea search methodology.

Jim was a graduate of the University of California, Berkeley, receiving a B.S. in 1966 and his Ph.D.—the first one awarded by the institution's Computer Science Department—in 1969. Among his many honors, Jim was a fellow of the Association for Computing Machinery (ACM), a fellow of the American Academy of Arts and Sciences, and a fellow of the Institute of Electrical and Electronics Engineers. He was elected to the National Academy of Engineering in 1995, the National Academy of Sciences in 2001, and the European Academy of Sciences in 2003. He received the ACM's 1998 A.M. Turing Award, the highest honor for a computer scientist or engineer.

A one-day memorial to celebrate Jim's life and contributions was held at Berkeley on Saturday, May 31, 2008, with 700 of his colleagues, family, and friends—all of us who would claim: "Jim Gray is one of my closest personal friends"…and we really thought we all were! Years after the tribute, we all retain that claim, and we continue to mourn his loss. In so many ways, Jim just made our lives more interesting.

This memoir draws on the 20 talks from that Berkeley memorial tribute.[1] Nominally, they address Jim's contributions in professional environments, or his many "professional lives," over almost a half-century. But these stories also transcend "what he did." A much larger part of Jim was "how he worked"—which helps explain why we all loved and respected him so much. All of the stories include some (often unique) facet of Jim's personality that goes beyond the professional contribution. For example, in his own tribute, Ed Lazowska, a longtime colleague and friend, nicely captures Jim's extraordinary ability to mentor and what it was like to be "a friend of Jim."[2]

Jim Gray's professional lives: what Jim did

Jim's short, five-decade professional life can be categorized into seven periods: (1) fundamentals-building at the University of California, Berkeley (1961-1969), often summarized to his wife Donna as, "Berkeley made me, and I owe them a lot;" (2) creating a theory for transactions, participating in the building of the first relational database system, and establishing design principles of database operating systems at IBM Research (1969-1980); (3) engineering fault-tolerant transaction processing systems and establishing measures for them at Tandem (1980-1990); (4) consolidating lasting gains through principles and measurement with his *Transaction Processing: Concepts and Techniques* book[3] and transaction-processing handbook[4] (1990-1994); (5) extending the applications of large data systems to scientists and consumers with the TerraServer (an online repository of public-domain aerial imagery and topographic maps) and the Worldwide Telescope at Microsoft Research (1995-2007); (6) developing "lifelogging," or "digital immortality;"and (7) demonstrating eScience. His last and lasting talk on the latter issue, given to the National Research Council's Computer Science and Telecommunications Board (CSTB), posited the Fourth Paradigm of Science based on data exploration;[5] that talk is also included in *The Fourth paradigm: Data-intensive scientific discovery*.[6] These seven periods are discussed in turn in the sections below.

> "One of the things that my research advisor Mike Harrison taught me to do is to write things down. So whenever I would go on a trip, I would write a trip report; and whenever I'd talk to people and we had an idea, I would write a memo about our discussion to document it."

This progression of a many-faceted career as a computer systems architect, engineer, and computer scientist—from foundation-building to the creation and measurement of tools to their application—is one that we view as Jim Gray's need and ability to "be where the action is" and to be forever relevant. Indeed, perhaps the greatest lifetime honor that one can bestow on a scientist or engineer, or any human being, is to remain relevant.

Jim Gray at Berkeley

In his tribute at the Berkeley memorial, Michael Harrison noted that "Jim Gray spent a decade as a student and researcher at Berkeley. In action, he is remembered for his breadth, his depth, and his generosity." [7] During this period, Jim wrote nine papers with colleagues, beginning with Harrison while Jim was an undergraduate. His last work at Berkeley was an exploration of Jay Forrester's System Dynamics, including a critique of its Urban Model.

In an ACM Interview, Jim offered:

> *One of the things that my research advisor Mike Harrison taught me to do is to write things down. So whenever I would go on a trip, I would write a trip report; and whenever I'd talk to people and we had an idea, I would write a memo about our discussion to document it. One consequence of this is that I wrote lots of papers and went to lots of conferences. [Another] consequence…is that I got to be very famous for the work of a lot of other people, which is not fair—but that's life. I've continued to tell people, [for example, that] that was Franco Putzolu's idea or…that Irv Traiger and I thought of that, and so on. But, since I wrote it down or gave the talk, I got credit for it in the world outside our group.*[8]

Jim and his wife Donna have been generous supporters of UC Berkeley and other universities through research fellowships and chairs. In that spirit a few of his friends matched the Hewlett Foundation's support in creating the Jim Gray Chair at Berkeley.

IBM Research: the relational database and the beginning of transaction processing

In 1971, Jim started his career at IBM Research in Yorktown, NY. Having grown up a San Franciscan, he could only tolerate this locale for about two years. Denied permission to be transferred at that time to IBM Research in San Jose, he quit, but he was rehired in October 1972 to work in the System R group (in San Jose), which was building

Receiving the Turing Award, 1998. (Photo courtesy Microsoft.)

the first relational database. He remained there until September 1980. In the ACM interview Jim described his work on System R:

The net result is that I was much more the researcher and much less the developer, and yet we had this project called System R. Franco wrote 20,000 lines of code a year and it worked, I wrote 10,000 lines a year that sort of worked. The project had all sorts of deliverables, and the managers wanted to show prototypes, and so there was a certain amount of pressure to deliver. At one point my boss came in and put a sign on my door that said, "Code faster." He was somewhat dismayed by the amount of time I spent writing papers, traveling around, and goofing off. I actually coded too fast. I created a lot of bugs. [Laughs] But yeah, I wrote, I don't know, 50,000 to 70,000 lines of code in System R. Concurrency control, recovery, system start-up, security, and administration are the parts that I worked on, mostly in the lower half, the so-called RSS layer, of the system.[9]

Michael Stonebraker wrote of this work in his article, "Why did Jim Gray win the Turing Award?:"

Jim wrote a pioneering paper in 1976 and followed this up with a book in the mid-1980s on this topic. He is largely responsible for the following (very simple in retrospect—but revolutionary at the time) ideas.

One should divide DBMS [database management system] activity into units of work, called transactions. A transaction consists of one or more statements in SQL (or whatever interaction language is supported) interspersed with code in a general purpose programming language. For example, a transaction might consist of moving $100 from account A to account B. In SQL (and most other interaction languages), this requires two statements, one to decrement account A and one to increment account B.

Each transaction must have the following properties:

Atomic: *Either the entire transaction happens or none of it happens. i.e., it is illegal to have the decrement happen unless the paired increment also happens. Hence transactions move the database from one consistent state to another.*

Consistent: *The database is free to define a collection of integrity constraints, [which] define legal data base states. One such requirement might be that account balances are nonnegative. Any transaction that makes an update [that] violates an integrity constraint must be aborted. Hence it is illegal to execute a transaction that produces an inconsistent DBMS state.*

Isolation: *This requirement means that parallel transactions cannot see the intermediate states of other transactions. In other words, the outcome of this collection must be the same as the collection run in some serial order, one after the other. Any other outcome is an inconsistent state. There is no requirement to obey any specific serial order, just a requirement to obey some serial ordering. This requirement defines legal database states when a collection of parallel transactions [is] run.*

Durable: *In the event of a failure, there are only two possible outcomes. Either a transaction "happened," i.e., it is committed; or it did not happen, i.e., it is aborted. If the user was notified that the transaction committed, then the DBMS agrees that it cannot develop a case of amnesia. Hence the effects of the transaction can never be lost, regardless of what failures might occur.*

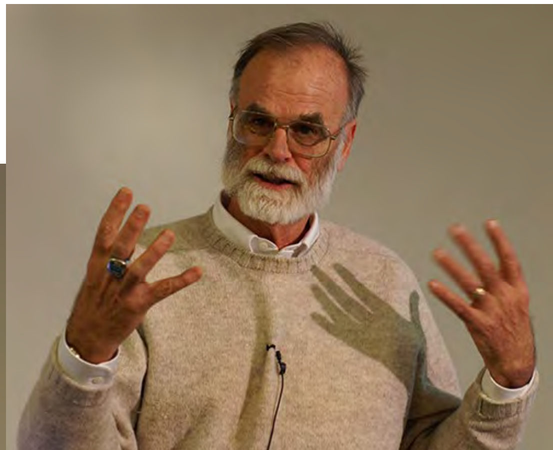
Together, these are called the "ACID properties." Supporting these properties efficiently is a deep intellectual topic, about which much has been written over the last quarter of a century. For example, one simple scheme is to "lock" all objects a transaction touches and hold all locks until the transaction ends. Every time a transaction makes an update, a "log record" is written holding both the "before image" of the object as well as the "after image." If the transaction must be undone, then the before image is used to "rewind" the database. If the effect of a committed transaction is lost because of a storage failure, then the after image is used to restore the effects of the committed transaction.

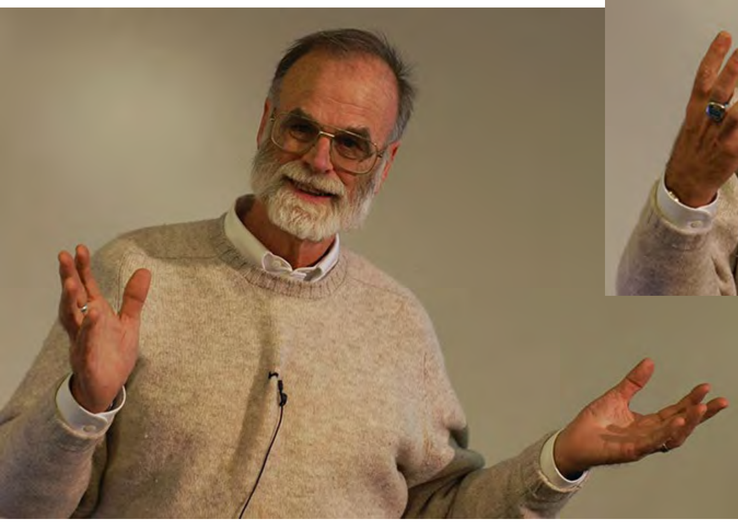
> *Working out the properties of transactions and then constructing efficient implementation schemes was the major contribution of Jim Gray in the 1970s and early 1980s. For this pioneering effort, he received the Turing Award in 1998.*[10]

Stonebraker concluded this tribute as follows:

> *Jim had three characteristics that I truly admire. First, he was an intellectual sponge…and seemed to know "everything about everything." Second, he was always willing to spend time discussing new ideas, and would freely give his perspective on other researchers' thoughts. …Third, he is one of the smartest people I have ever known.*

Jim's courses, and especially his "Notes on Database Operating Systems,"[11] are especially noteworthy and were important to advancing the field. The notes article also exhibits a recurring theme: his lifelong ability as a teacher and researcher committed to the evolution of computing.





(Photos courtesy Microsoft.)

Tandem: the beginning of a fault-tolerant transaction processing industry

Jim arrived at Tandem Computers, Inc., in 1980, just after it had introduced the first fault-tolerant computer, called NonStop I, which consisted of independent communicating computers—a primary process running in one computer reported its status to a backup process running in another computer. At the Berkeley memorial, Nauman and Bartlett described Jim's contribution at Tandem:

> *NonStop SQL was developed by a relatively small team, many of whom Jim recruited from outside Tandem. He served as everything from architect to developer to cheerleader within the team while at the same time continuing to explain the benefits to Tandem's upper management and fostering customer interest and support.*[12]

It was at Tandem that Jim was instrumental in creating the whole transaction processing industry, in no small part through his leadership in the establishment of a standard measure of performance. The launching of this benchmark was somewhat unconventional—it was published in *Datamation*, the dominant computer-industry trade magazine, on April 1, 1985. It listed Anon et al. as the authors so as to assign no attributions to them. The article, "A measure of transaction processing power," began with:

> *A measure of transaction processing power is needed—a standard that can measure and compare the throughput and price/performance of various transaction processing systems. Vendors of transaction processing systems quote Transaction Per Second (TPS) rates for their systems. But there isn't a standard transaction, so it is difficult to verify or compare these TPS claims…This paper is an attempt by two dozen people active in transaction processing to write down the folklore we use to measure system performance. The authors include academics, vendors, and users.*

In their tribute, Nauman and Bartlett conclude:

> *Throughout his career, and particularly while he was at Tandem, Jim recognized areas where the existing ideas and practices would not be sufficient in the future. He became involved in those areas from both a theoretical and practical perspective and moved them forward with insight, research, papers, presentations, and products.*

How to build and measure transaction processing systems

"The process of writing had many aspects of a typical software project: In the end, the book was more than twice as thick as we had planned, it covered only three-fourths of the material that we wanted to cover, and completing it took much longer than we had anticipated."

After leaving Tandem, Jim went to the Digital Equipment Corp. (DEC) as a corporate consulting engineer. He was given free rein to work on a wide range of problems, and also served as editor of a database book series and as a member of the National Research Council's Computer Science and Telecommunications Board. While at Digital, he created and edited two editions (1991, 1993) of *The Benchmark Handbook for Database and Transaction Processing Systems*[13] and wrote the *Transaction Processing: Concepts and Techniques* textbook with Andreas Reuter[14] as a result of teaching various courses at Stanford University's summer school. In Reuter's tribute to Jim,[15] he summarized the experience of writing this monumental 1,000- page text:

> *The process of writing had many aspects of a typical software project: In the end, the book was more than twice as thick as we had planned, it covered only three-fourths of the material that we wanted to cover, and completing it took much longer than we had anticipated.*

One aspect of Jim revealed by this book was his incredible discipline and stamina. Reuter tells a story that many of us who have written a book can relate to:

> *In 1986 Jim had signed a contract with a seminar organizer for teaching a one-week course on transaction processing in spring 1987. The course was to take place in Berlin, and because he did not want to teach the full five-day load all by himself, he invited me to share some of it, assuming that university professors have most—if not all—of the course material ready for delivery on short notice. For the following eight months we worked on preparing the slides. The plan guiding the process was a list of chapter headings that each of us would work on, with each chapter representing a 90-minute lecture.*

When we reviewed the course [after teaching it], Jim observed that the students' questions had produced something that our initial organization of the material had not suggested: a systems-oriented perspective on transaction processing. So instead of describing TP technology in isolation, and then describing databases, networking, programming issues, etc., we presented transactional concepts as some kind of unifying framework for all layers of a system, from the operating system all the way up to the applications and the user interfaces. Jim went on saying that there were no textbooks taking that integrative approach, and that it should not be too hard to turn our 980 foils into text. We estimated that on average two slides would transform into one page of prose (including tables and figures), so that we would have to write ca. 500 pages—250 pages per person. Doing this within a year seemed quite reasonable at the time, given that we had most of the material already. This is how the whole thing started.

Before we actually tried to implement it, the initial plan seemed to make perfect sense: the major portion of the work had already been completed by putting all the technical material on the slides, or so we thought. We [expected that we] would only have to convert bullets into sentences, redo some of the figures, add a chapter drawing all the details into the grand picture of transaction-oriented systems, compile a list of references—and be done! It was the kind of plan that everybody will enthusiastically agree to at the end of a meeting, so they can get on with their real work. In our case it was the review meeting after the course, and neither Jim nor I had a clear idea of how to implement it after we got back from Berlin. However, with the best of intentions we agreed on producing text from the slides sometime soon.

With no deadline at all and many other things to do, we made very little progress in turning the foils into prose—in fact, we did not make any progress at all. I used the material for a variety of courses I taught at the university, extending and changing it as new algorithms, new systems, etc., became available. Jim did the same, teaching transaction processing at Stanford, but we still were just using and updating the slides; no prose was being produced as a result of the teaching activities. The only new type of content that proved useful when, much later, we actually wrote the book was a rapidly *growing number of problems and exercises related to the various topics that were covered in the foils. Those problems were specifically created for the university courses; they had not been part of the Berlin seminar.*

In his San Francisco office.
(Photo courtesy Gordon Bell.)

That was the situation in 1987, and it did not change in 1988, or in 1989. In the fall of 1989 we discussed the project and found that the original plan had been a failure. It was obvious that if we wanted to get anything written, we would have to hide in some remote, quiet, and pleasant place, equipped with PCs, printer, [and] toner, with easy access to good food, and spend all our time typing—well, most of it. We figured that three months should be enough to produce a complete first draft of the book, the polishing of which could be done later, when we were back in our normal habitats. After some lengthy and careful deliberation it was decided to rent a house in a small village in Tuscany named Ripa (near Carrara) and spend February through April of 1990 there.

This time we got it partially right; at the end of April we had about 600 pages of text, thanks to Jim's strict regime that required each of us to produce 2,000 words per day, no matter which day. Six hundred pages were very close to our estimate—but they only covered less than half the topics we wanted to discuss. So in order to preserve the investment, we had to plan for a second hideaway, which took place one year later in Bolinas (north of San Francisco), again from February to April. At the end of this period, we had about 1,000 pages of text, plus a number of lessons learned:

- *We would not be able to cover all the material that was contained in the foils of the course.*

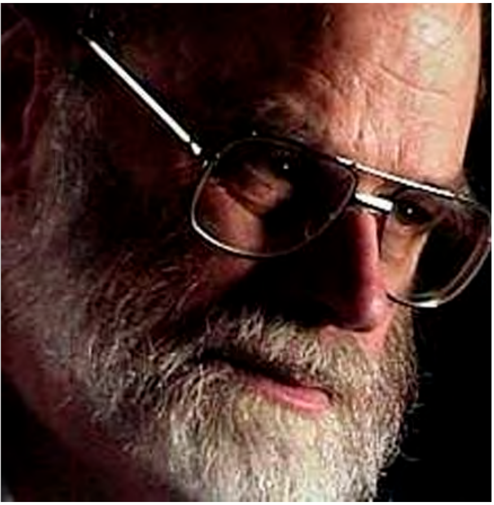
- We would still have to do a lot of work in order to get the book to the printer (glossary, index, and proofreading).

- Writing a book is hard work; we would never do it again.

So Robert Burns was right indeed: The best laid plans…

Microsoft Research: It's about data for science

I (the first author, Gordon Bell) don't recall how we decided to get together in 1994, but I felt flattered and honored for what became a wonderful friendship. Our first meeting at my Los Altos, CA, home was wide-ranging: scalable architectures, the importance of industry standards (hence Microsoft), and research driven by creative applications. Jim's professional trajectory at Microsoft is nicely characterized in Stokes' book *Pasteur's Quadrant*,[16] which describes the motivations and behaviors of scientists and engineers in research and development. Jim is rare, "up and to the right," in the quadrant where research is inspired both by a quest for fundamental understanding and considerations for use.



(Photo courtesy Donna Carnes.)

I also would like to think I precipitated Jim to join Microsoft Research, but in reality Jim had many friends at the company, and in the end just urged Rick Rashid and Nathan Myhrvold to hurry up and make him the offer to establish a San Francisco lab. The Bay Area Research Center (BARC) opened in the summer of 1995. Tom Barclay came back from Redmond to join Jim; I was honored and delighted to join in August to work on telepresence. By October 1995, Jim had specified an "NT clusters" research agenda that included a BARC project that could be built to test the concept. Working with the Redmond NT clusters group, it would especially answer questions about computer cluster scalability.

In 1997, a cluster with 120 disks holding a terabyte of earth-image data became live. On the May 1997 Scalability Day, Jim gave an on-stage demo with Bill Gates using 120 computers in a room of 20-plus racks to achieve one billion transactions per day. I also recall the glee when Jim had just finished measuring transactions on April Fool's Day 2005 using his relatively old laptop and observing "Thousands of debit-credit transactions per second: Easy and inexpensive." This was 20 years after the article describing the transaction benchmark. In an article abstract in 2005, he wrote:

> *A $2K computer can execute about 8K transactions per second. This is 80x more than one of the largest U.S. bank's 1970s [daily] traffic—it approximates the total U.S. 1970s [daily] financial transaction volume. Very modest modern computers can easily solve yesterday's problems.*

This rate amounted to a half-billion transactions per day on a laptop, compared to the roomful of servers at Scalability Day 1997.

Through various paths, Jim had infected me (and many others) with the importance of data—it's "all about the data." In one of our more playful times, while discussing how to get the concern for data into the national computing resource-allocation agenda, Jim and I bumped into John Markoff, a friend and columnist at the *New York Times*. We proceeded to posit a view that science is missing the point by just thinking about computation speed, aka FLOPS (floating point operations per second). John took our picture in the lab on a Friday and the article appeared [two days later] in the Sunday [June 1, 2003] *New York Times*.[17] Our friends in Washington were not especially happy, but they acknowledged we were right and in 2012, after almost a decade, things are slowly changing—"big data" is a current rallying cry for computing funding, including that of the high-performance community, whose programs generate so much data.

Jim's collaborative and supportive spirit was manifest in the constant parade of visitors to the laboratory in San Francisco. As Ed Lazowska points out in a subsequent section, he had the time to help anyone he felt needed him.

Digital immortality

Bell and Gray's 2001 paper on "digital immortality" [18] marked the real beginning of Bell's MyLifeBits project for holding all of a person's life. With Jim's encouragement and sponsorship, Jim Gemmell and Roger Lueder joined Bell in exploring the use of such a database (Bell had initially rejected the suggestion to use one). Jim Gray posited this research topic as "Personal Memex: Record everything a person sees and hears, and quickly retrieve any item on request"— one of a dozen research goals presented in his 1998 ACM Turing Award Lecture and also in a Microsoft report that he wrote in 1999.[19]

MyLifeBits has remained a quest—thanks mainly to Jim's support and belief in such "lifelogging." Bell continues to speculate about digital immortality, especially with respect to Jim and his legacies, enabled by a life log and evolving technology. For example, Jim's website provides a record of his several major "professional lives," described in an extensive curriculum vita that starts in 1962 with a student co-op job and includes his positions at 10 other academic and commercial organizations. The website has 185 articles and books published since 1966, as well as 212 Microsoft and other system-related talks given since 1994. It lists 50 events in 2006 of which Jim was a part; a dozen websites that Jim operated; earned and honorary academic degrees; memberships in over 20 public boards, advisory committees, program committees, and five societies—including the National Academy of Sciences and the National Academy of Engineering.

Jim and telescope.
(Photo courtesy Microsoft.)

The website is just a glimpse of Jim's prodigious output (limited as the site is to papers, public reports, and presentations) and of the communities of which he was a part. By his vitae and the Berkeley memorial, we can see many of his active parallel lives, within the computer science, database, fault tolerance, performance, transaction processing, and systems communities, and the timelines of those lives—reflecting his tenures at IBM (11 years), Tandem Computer (10 Years), DEC (4 years), and Microsoft (12 years), among others. At Microsoft alone, his lives since 1995 included eScience and systems generally, with deep contributions to astronomy, bioinformatics, databases, oceanography, servers, and terrestrial data. Professional society and educational institutions constituted another set of lives. His friends and colleagues all cherish how each of our own lives intersected with his. For example, Bell enumerates 13,000 emails, 1,600 web pages, 100 presentations, 50 photos, 600 spreadsheets, and a host of documents that were written or coauthored by Jim or that reference him.

Positing the fourth paradigm by exemplary work with other scientific disciplines

Coming from the work on a TerraServer for earth-image data, the natural extension was to look up—hence the SkyServer. While Jim worked with scientists from many disciplines, his dominant collaboration was with Alex Szalay and the astronomy community. According to Alex, astronomers "embraced Jim as a 'card carrying member' of their community. Jim's contributions have made a permanent mark on astronomy, and eScience in general." Alex recounts collaborations with astronomers that resulted in over 20 published papers between 2000 and 2007; they even named an asteroid after him.

One of Jim's most lasting contributions is likely to be "eScience"—his recognition, description, and naming of data discovery using computers—which he called the Fourth Paradigm of Science. His last talk posited this paradigm to the National Research Council's Computer Science and Telecommunications Board.[20] Experimental science and theoretical science (the first and second paradigms) have evolved as the basis of understanding nature. In the mid-1980s, computer simulation for exploring phenomena inaccessible to analysis was named the third paradigm. Simulation results and the emergence of ever more data enabled Jim to observe a fourth paradigm, which encompasses everything from initial data creation and provenance, through intermediate analysis and visualization for extracting meaning, to long-term preservation. Jim's wide-ranging NRC/CSTB talk covered the philosophical change in data-based science, the need for databases and collaboration with computer scientists, scientific publication based on openness and peer review, and the use of wikis.

The first-published book on the fourth paradigm, with 25 case studies from eScience, begins to illustrate its scope and to hint at its potential.[21] The National Science Foundation's report on the importance of data[22] begins:

> *It is exceedingly rare that fundamentally new approaches to research and education arise. Information technology has ushered in such a fundamental change. Digital data collections are at the heart of this change. They enable analysis at unprecedented levels of accuracy and sophistication and provide novel insights through innovative information integration. Through their very size and complexity, such digital collections provide new phenomena for study.*

Life as a friend and mentor:

Ed Lazowska's tribute to Jim's mentoring[23] describes many of the attributes we all admired and enjoyed.

> *So many things are so special about Jim as a mentor. None of them are rocket science. It's that he did them all, and did them all so consistently and so well: making time; simply listening; inspiring self-confidence; lighting the way; nurturing and pushing; following the muse; connecting good people and good ideas without boundaries; promoting the young; sharing knowledge selflessly; displaying professional integrity; advocating for the field; keeping things in perspective; [and] being a friend. Let's briefly look at each of these:*
>
> ***Making time.*** *Time is the most precious gift that one can give or receive. James Hamilton, in his tribute to Jim at a Microsoft event in January, tells a story that is familiar to each of us: "Jim came over, sat down beside me, and said 'How are you doing, James Hamilton?' This is signature Jim. I'll bet nearly everyone he knows has had one of those visits during the course of a conference. He drops by, sits down, matches eyes, and you have 110 percent of his attention for the next 15 to 20 minutes."*
>
> *Even beginning graduate students benefited from this attention. Remzi Arpaci-Dusseau recalls: "What struck me was Jim's insistence on scheduling time with our graduate students. He would sit in their offices, chat about their work, give lots of advice, and generally do for them what he had done so long ago for me: make them feel like they (and the problems they were working on) were important. What better gift than that?"*
>
> ***Simply listening.*** *Listening is a difficult art to master, and Jim was a master. Remzi Arpaci-Dusseau says: "Even in those first days when I was just a young (and relatively clueless) graduate student, Jim took every idea I mentioned seriously, encouraged me to continue with my work while giving me new ideas and directions, and did something that is all too rare: he simply listened when I spoke, and treated me as a peer."*
>
> *Natassa Anastasia recalls: "I remember an event at Bill Gates's house where all the interns were gathered for dinner; Jim would walk up to all the first-year graduate student interns and say 'Hi, my name is Jim Gray.*

What are you working on? Are you having fun?'...He made everyone feel so comfortable around him, so at home....There was not a single time he said he was too busy to talk to me; I would pick up the phone and call him and tell him all that I was doing and he was just happy to listen and, when asked, provide help."

Inspiring self-confidence. *Alfred Spector writes: "Jim was my most influential graduate-school advisor. I was hugely influenced by his work. Even more importantly, I was influenced by the enormous confidence he showed in me. He somehow managed to convey that he thought I was bright, articulate, and would make significant contributions."*

Lighting the way. *Whether advising students, supporting young faculty, or collaborating with established researchers, Jim "suggested" rather than "prescrib[ed]." Johannes Gehrke writes: "Jim was not generous in a controlling way—the instructions accompanying his research grant awards were 'do good science.'" James Hamilton observes: "Jim's style is not to correct or redirect. Yet, after each conversation, I've typically decided to do something differently. It just somehow becomes clear."*

Alex Szalay adds: "Jim was incredibly patient and supportive and willing to listen—much more so than anyone else I have ever known. He would not say, 'This is what you must do'—he would gently light the way, so that people would find the path themselves."

Nurturing and pushing. *Jim knew that different career stages and different life stages required different styles of mentoring. Joe Hellerstein writes: "By 1994 when I met him, Jim's role as a leader of the field was long cemented. I saw him give a talk that year at Wisconsin, and as an ambitious grad student I raised a couple technical questions. He put my name in the acknowledgments of the subsequent paper, which eventually became a classic in the field. That was vintage Jim: always taking time to promote the next crop of young folks while chugging along on his own scientific agenda. I'll say that when I arrived at Berkeley, Jim's relationship with me changed, and it took me a while to figure it out. He seemed to become more antagonistic—questioning my direction, grilling my students, and generally pushing back on our technical agenda. I've seen this enough times with other young folks now to realize it was a* *pattern in his mentorship: he liked to gently raise promising folks up into the big leagues, and then switch tactics and turn up the heat....Very few folks in academia are as thoughtful about mentorship over the course of a career."*

Following the muse. *In his 2002 SIGMOD interview Jim said: "I don't believe in an afterlife, so I think this is it, and I'm trying to spend my time as best I can...So I have, in fact, only worked on things that I thought could really be significant...and I always tried to be in a situation where I could quit the job I was doing that very day if the need came. I think that was liberating. Of course, it made me a manager's nightmare."*

Jim inspired others to follow their muse. Jignesh Patel remembers: "I decided that I really wanted to do something that had a long-term impact on society. I decided I was going to jump into the life sciences. I asked a number of senior faculty in my department, and every single one of them told me that I was making a foolish move—that I should stay with my strengths and keep publishing in my established area...I started talking to Jim about this new direction, and he immediately saw the potential and was enthusiastically in favor of this move. ...In every paper I write on biological data management, I ask myself 'What would Jim think of this work?'"

Connecting good people and good ideas without boundaries. *Jim was an extraordinary connector. James Hamilton put it beautifully: "Anyone can talk to Jim, and an astonishing number frequently do. And because his review comments are so good, and he's so widely respected, a mammoth amount is sent his way. He receives early papers and important new results across a breadth of fields from computer architecture, operating system design, networking, databases, transaction processing, astronomy, and particle physics. The most interesting work he comes across is forwarded widely. He ignores company bounds, international bounds, bounds of seniority, and simply routes people and useful data together. Jim effectively is a routing nexus where new ideas and really interesting results are distributed more broadly."*

Promoting the young. *Jim's role as a "connector" was particularly focused on drawing attention to good work by young researchers. "Even as a busy lab director," Erik Riedel notes, "Jim managed to take the time to respond to students, to connect new researchers to the established members of the community. ...[H]e understood that a query from a student or new engineer can be just as important long-term as a message from a corporate vice president." Yannis Ioannidis adds, "Jim became my best supporter and advocate. He kept talking about my work to others."*

Sharing knowledge selflessly. *Some people hoard what they know—"knowledge as power." Jim shares with all. Mike Carey provides an example: "When I was thinking of taking the plunge from IBM (safe) to Propel (scary startup), the job I was considering sounded like something Jim would have been much better qualified for than I. On a whim, I sent Jim a note saying something like 'Hey, I'm considering a move from IBM to a job that you should be taking, not me—I don't feel like I know enough to do it' (and I explained a bit about the job). Jim replied 'Come visit me at my SF lab and I'll tell you what you need to know about that.' I did—and he did. He gave up 2 or 3 hours of 'free consulting' to tutor me—for no reason other than always being there to support younger folks in his field."*

Displaying professional integrity. *Elsewhere in this volume, David DeWitt describes Jim's masterminding of the [1985] 24-author "Anon. et al." paper, "A Measure of Transaction Processing Power," which led to the establishment of the Transaction Processing Council and established the closest thing possible to a level playing field for comparing transaction processing systems. The "Anon. et al." authorship was not merely humorous—it provided valuable air cover for the 14 authors employed by vendors (David and I were among the eight academic authors), few of whose employers would have been pleased by this activity. Only Jim had the professional integrity—and the reputation for professional integrity—to lead such an effort.*

Advocating for the field. *Jim was a tireless advocate for investment in research. In his 2002 interview in* SIGMOD Record *he said, "I go to Washington because I believe that a dollar invested in scientific research*

(Photo courtesy Donna Carnes.)

gets a payback of 10 dollars to society." His 1998 Turing Award lecture is remarkable for laying out a vision for our field, rather than focusing on Jim's own contributions. A good long-range research goal, he said, should be understandable, challenging, useful, testable, and achievable through intermediate milestones. He proposed a dozen examples: four

(Photo courtesy Donna Carnes.)

variants of the Turing test (human imitation, hearing, speech, and object recognition); a personal and a world Memex; telepresence; systems that are trouble-free, secure, always up, and scalable; and an automatic programmer. This inspirational lecture is must reading for those who have not done so, and must re-reading for those who have.

Keeping things in perspective. *I've served on the Technical Advisory Board for Microsoft Research [MSR] since its inception in 1991. MSR boasts many extraordinary achievements, but like any organization or individual, it has its infuriating moments. At one point I was blowing off steam at Jim regarding a particular aspect of MSR's university relations. Jim looked at me calmly and said, "Think of it like teenage sex. It's new to them. They're incredibly clumsy at it. They don't want to take advice. But trust me—they'll figure it out."*

Being a friend. *Alfred Spector writes: "Jim and I became personal friends over the many years. Many sailing trips, visits to his boat, visits to his and Donna's lovely home on Telegraph Hill....His encouragement and leadership inspired even my kids to do some things they hadn't done before, to know a little more about the world, and to be a bit bolder."*

It is impossible to overestimate Jim's durable impact—for example, Bell recently recycled several of the dozen challenges he posited in his 1998 ACM Turing Award Lecture as 2030 challenges for computing. We continue to hear "What would Jim say?" when we have a problem to solve; and when greeting a shared friend, Jim is the center of our conversation. This came from his being a wonderful and well-informed conversationalist, storyteller, and spokesperson for computing. We all greatly miss him, just as everyone else who knew him does.

AUTHORS' NOTE

The authors would like to thank all those who contributed their published memories of Jim and especially Ed Lazowska who captured the essence of Jim. The authors also gratefully acknowledge Jim's wife Donna Carnes who contributed photos, suggestions, and support.

NOTES

1. Labrinidis, A. (ed.). 2008. Special issue: Tribute to honor Jim Gray. *ACM SIGMOD Record* 37(2):16-82. The event that inspired this issue took place at the University of California, Berkeley, on May 31, 2008.
2. Lazowska, E. 2008. 500 Special relationships: Jim as a mentor to faculty and students. In Special issue: Tribute to honor Jim Gray. *ACM SIGMOD Record* 37(2):30-32.
3. Gray, J., and A. Reuter. 1992. *Transaction processing: Concepts and techniques*. San Mateo, CA: Morgan Kaufmann.
4. Gray, J. (ed.). 1992. *The benchmark handbook for database and transaction processing systems*, 2nd edition. San Mateo, CA: Morgan Kaufmann.
5. Gray, J. 2007. On eScience: A transformed scientific method. Presentation to the CSTB. Mountain View, CA, January 11. Online at *http://research.microsoft.com/en-us/collaboration/fourthparadigm/4th_paradigm_book_jim_gray_transcript.pdf.*
6. Hey, T., S. Tansley, and K. Tolle. 2009. The Fourth paradigm: Data-intensive discovery. Redmond, WA: Microsoft Research. Online at *http://research. microsoft.com/en-us/collaboration/fourthparadigm/.*
7. Harrison, M. 2008. Jim Gray at Berkeley. In Special issue: Tribute to honor Jim Gray. *ACM SIGMOD Record* 37(2):26-27.
8. Association for Computing Machinery (ACM). 2003. A conversation with Jim Gray. *Queue* 1(4):8-17. Online at *http://queue.acm.org/detail. cfm?id=864078.*
9. Association for Computing Machinery. A conversation with Jim Gray.
10. Stonebraker, M. 2008. Why did Jim Gray win the Turing Award? In Special issue: Tribute to honor Jim Gray. *ACM SIGMOD Record* 37(2):33-34.
11. Gray, J. 1978. Notes on database operating systems. In *Operating Systems: An Advanced Course*, edited by R. Bayer et al. New York: Springer-Verlag, 393-481.
12. Nauman, J., and W. Bartlett. 2008. Jim Gray's Tandem contributions. In Special issue: Tribute to honor Jim Gray. *ACM SIGMOD Record* 37(2):41-44.
13. Gray, *The benchmark handbook*.
14. Gray and Reuter, *Transaction processing*.
15. Reuter, A. 2008. Is there life outside transactions? Writing the transaction processing book. In Special issue: Tribute to honor Jim Gray. *ACM SIGMOD Record* 37(2):54-58.
16. Stokes, D. E. 1997. *Pasteur's quadrant: Basic science and technological innovation.* Washington, DC: Brookings Institution Press.

17. Markoff, J. 2003. In computing, weighing sheer power against vast pools of data. *New York Times*, June 1.

18. Bell, G., and J. Gray. 2001. Digital immortality. *Comm. of the ACM* 44(3):28-30.

19. Gray, J. 1999. *What next? A dozen information-technology research goals.* (1998 ACM Turing Award Lecture.) Technical Report MS-TR-99-50. Redmond, WA: Microsoft Research. Online at *ftp://ftp.research.microsoft. com/pub/tr/tr-99-50.pdf.* See also *http://research.microsoft.com/~Gray/Talks/Turing2.ppt.*

20. Gray, J. 2007. On eScience: A transformed scientific method. Presentation to the Computer Science and Technology Board of the National Research Council. Mountain View, CA, January 11. Online at *http://research.microsoft. com/en-us/collaboration/fourthparadigm/4th_paradigm_book_jim_gray_transcript. pdf.*

21. Hey, T., *et al*, ed. *The Fourth paradigm: Data-Intensive Scientific discovery.* Redmond, WA.: Microsoft Research.

22. National Science Board. 2005. *Long-lived digital data collections: Enabling research and education in the 21st century.* Washington, DC: National Science Foundation.

23. Lazowska, E. 2008. 500 Special relationships: Jim as a mentor to faculty and students. In Special issue: Tribute to honor Jim Gray. *ACM SIGMOD Record* 37(2):30-32.

SELECTED BIBLIOGRAPHY

1970 Locking. In *Proceedings of the Woods Hole Conference on Concurrent Systems and Parallel Computation*, edited by J. Dennis. New York: ACM, 169-176.

1978 Notes on database operating systems. In *Operating Systems: An Advanced Course*, edited by R. Bayer et al. New York: Springer-Verlag, 393-481.

1985 With Anon. et al. A measure of transaction processing power. *Datamation* 31(7):112-118. (Gray wrote this article with 23 coauthors from academia and the transaction-processing industry in order to establish the benchmark standard for transaction processing.)

1992 With D. DeWitt. Parallel database systems: The future of high performance database systems. *Comm. of the ACM* 35(6):85-98. Online at *http://research.microsoft.com/%7Egray/papers/CacmParallelDB.pdf.*

1992 With A. Reuter. *Transaction processing: Concepts and techniques*. San Mateo, CA: Morgan Kaufmann.

1993 With D. Bitton et al. *The benchmark handbook for database and transaction processing system*s, 2nd edition. San Mateo, CA: Morgan Kaufmann.

1998 With T. Barclay et al. *The Microsoft TerraServer*, Technical Report MSR-TR-98-17. Redmond, WA: Microsoft Research.

1999 *What next? A dozen information-technology research goals.* (1998 ACM Turing Award Lecture.) Technical Report MS-TR-99-50. Redmond, WA: Microsoft Research. Online at *ftp://ftp.research.microsoft.com/pub/tr/tr-99-50.pdf. See also http://research.microsoft.com/~Gray/Talks/Turing2.ppt.*

2001 With G. Bell. Digital immortality. *Comm. of the ACM* 44(3):28-30.

2002 With T. Barclay et al. *TerraService.NET: An introduction to web services*. Technical Report MSR TR 2002-53. Redmond, WA: Microsoft Research.

With A. S. Szalay et al. Data mining the SDSS SkyServer database. In *Distributed Data & Structures 4: Records of the 4th International Meeting*, edited by W. Litwin and G. Levy. Kitchener, Ontario: Carleton Scientific, 189-210; also Technical Report MSR-TR-2002. Redmond, WA: Microsoft Research. Online at *ftp://ftp.research.microsoft.com/pub/tr/tr-2002-01.pdf.*

2002 With A. S. Szalay, et al. Online scientific data curation, publication, and archiving. In *Proceedings of SPIE Astronomy Telescopes and Instruments*, *August 22-28, Waikoloa, Hawaii.*

With A. S. Szalay. "The World-Wide Telescope: An archetype for online science. *Comm. of the ACM* 45(11):50-54.

2003 With A. R. Thakar et al. The Sloan Digital Sky Survey Science Archive: Migrating a multi-terabyte astronomical archive from object to relational DBMS. *Computing in Science and Engineering* 5(5):16-29.

2005 A measure of transaction processing 20 years later. *IEEE Data Engineering Bulletin* 28(2):3-4.

With D. T. Liu et al. Scientific data management in the coming decade. *ACM SIGMOD Record* 34(4):35-41.

Behind the code with Jim Gray (interview on video). MSDN Channel 9, December 15. Online at *http://research.microsoft.com/apps/video/dl.aspx?id=104284.*