Journal of Universal Computer Science, vol. 13, no. 8 (2007), 1044-1046
submitted: 7/7/07, accepted: 7/7/07, appeared: 28/8/07 © J.UCS

Dedication to Jim Gray

David Lomet
(Microsoft Research, USA
lomet@microsoft.com)

As most readers undoubtedly know, Jim Gray did not return from a sailing trip he took in January of this year (2007) and has been missing since then, despite one of the most comprehensive and far reaching search operations that the world has ever seen. While the search effort was unprecedented, the willingness of people to volunteer their time to help in the effort was not surprising given both Jim's manifest and widely known technical accomplishments and the many people he has befriended, helped, and mentored over the years. Thus, we dedicate this volume, the Dagstuhl Seminar on "Atomicity: A Unifying Concept in Computer Science" to Jim Gray, whose work has done so much to shape our notion of atomicity and the many roles it can play, does play, and should not be expected to play in computer science, all topics on which Jim has published.

In 1998 Jim was awarded the ACM Turing Award for "For seminal contributions to database and transaction processing research and technical leadership in system implementation." More information about Jim and the award is available on the ACM web site `http://awards.acm.org/citation.cfm?id=7232067&srt=all&aw=140&ao=AMTURING`.

The complete Turing Award citation reads: "For fundamental contributions to database and transaction processing research and technical leadership in system implementation from research prototypes to commercial products. The transaction is the fundamental abstraction underlying database system concurrency and failure recovery. Gray's work led to the definition of the desired key transaction properties: atomicity, consistency, isolation and durability; and his locking and recovery work demonstrated how to build database systems that exhibit these properties. His fault tolerance work explained how to use transactions to achieve high system availability."

At the 1999 SIGMOD Conference, I gave a short speech at a ceremony honoring Jim and his winning the Turing Award. I'd like to reproduce that here as it hints at both the breadth of Jim's technical accomplishments and his influence in our technical community.

'It is a real pleasure for me to have an opportunity to speak here. It is rare that one of our database colleagues is recognized with a Turing Award. Indeed, it has been twenty years since Ted Codd won the award. I think it is particularly fitting that it is Jim who has won the award. While Ted contributed the relational model to our field, Jim provided us with the other great abstraction of our field, transactions. These two abstractions have been, and continue to be, central to our field.

My personal interactions with Jim go back almost 25 years, to when we were both at IBM (though we worked on opposite sides of the country). I can indeed remember when Jim was not a technical leader of our field, it was back before there was a field called relational databases, back before System R. System R was the progenitor of all subsequent relational database systems. It showed the world that relational database systems were REAL. As in so much of the subsequent history of our field, Jim was a technical leader of that effort, particularly in what became known as the RSS (relational storage subsystem) component. I recently had the opportunity, via invitation from Rick Snodgrass, to contribute my thoughts on a most influential paper. For me the choice was real easy. It was Jim's "Notes on Database Operating Systems". This paper grew out of Jim's experience with System R and served as an early bible on how to implement transaction support in database systems.

Later, when Jim was thinking of leaving Tandem, I played an enthusiastic if minor role in helping to recruit him to DEC. I can remember how we did this. First, we had to convince Jim that our company was serious about databases. That was sometimes hard. Then we had to convince our company that Jim should have a lab in San Francisco. That usually took some time. But it worked, and Jim made an enormous contribution to DEC's prominence in the TP/DB world. The most public part of Jim's impact while at DEC was AlphaSort, demonstrating that the Alpha chip's blazing clock time could be converted into blazing data processing performance. But I think Jim's contribution inside of DEC was more important. At a point when DEC's production systems strategy was bordering on catastrophe, Jim picked up the pieces, and turned it into a coherent technical AND business strategy. And when DEC was withdrawing from the software arena, Jim was a leader in the effort to keep that business going.

After DEC exited from the production software business, fate arranged for me to be hired at Microsoft. So, let me now repeat part of the previous story—I played an enthusiastic if minor role in helping to recruit him to Microsoft. I can remember how we did this. First, we had to convince Jim that our company was serious about databases. That was sometimes hard. Then we had to convince our company that Jim should have a lab in San Francisco. That usually took some time. But it worked.

And once again, Jim made an enormous contribution to (this time) Microsoft's prominence in the TP/DB world. Jim was instrumental in bringing data cube support to SQL Server, he showed SQL Server scalability via a wonderful and useful demonstration called TeraServer, demo'd last year at SIGMOD and subsequently live on the web ever since. Jim consults with the NT group on clusters, and the SQL Server group on concurrency control and recovery. In my meeting with Bill Gates on his SIGMOD 98 speech, I remember Bill saying, when asked about some database technology, that Jim had told him about the technology, and if Jim had said it, Bill had great confidence that this was correct. (It is not easy to impress Bill Gates—but Jim has.)

Finally, Jim's research does not consist only of clever solutions to hard problems. What really distinguishes Jim's work is the number of times in which he has defined what the hard problems were: recovery, transaction commit, concurrency control, why systems fail, sorting, benchmarking (he was Anon of Anon et al), OLAP (data cube). Jim always seems to be working in the future. It was Jim who defined "petabytes" for me, when I was still trying to remember what "terabytes" meant.

I believe that giving Jim the Turing Award not only honors Jim, Jim brings honor to the Award.'

So this dedication is truly fitting- and honors us while it honors him.