

► Len Shustek, Column Editor

Viewpoint

Let's Not Dumb Down the History of Computer Science

Donald Knuth on the best way to recognize the history of computer science.

Editor's note: On May 7, 2014, Don Knuth delivered that year's Kailath Lecture^a at Stanford University to a packed auditorium. In it he decried the absence of technical content from the histories of computer science being written, and he made an impassioned plea for historians of computer science to get back on track, as the historians of mathematics have always been.

Both the video^b and, now, the verbatim transcript^c of that talk are online. In the January 2015 issue of *Communications*,^d historian Thomas Haigh analyzed and responded to the talk, concluding that “work of the particular kind preferred by Knuth will flourish only if his colleagues in computer science are willing to produce, reward, or commission it.”

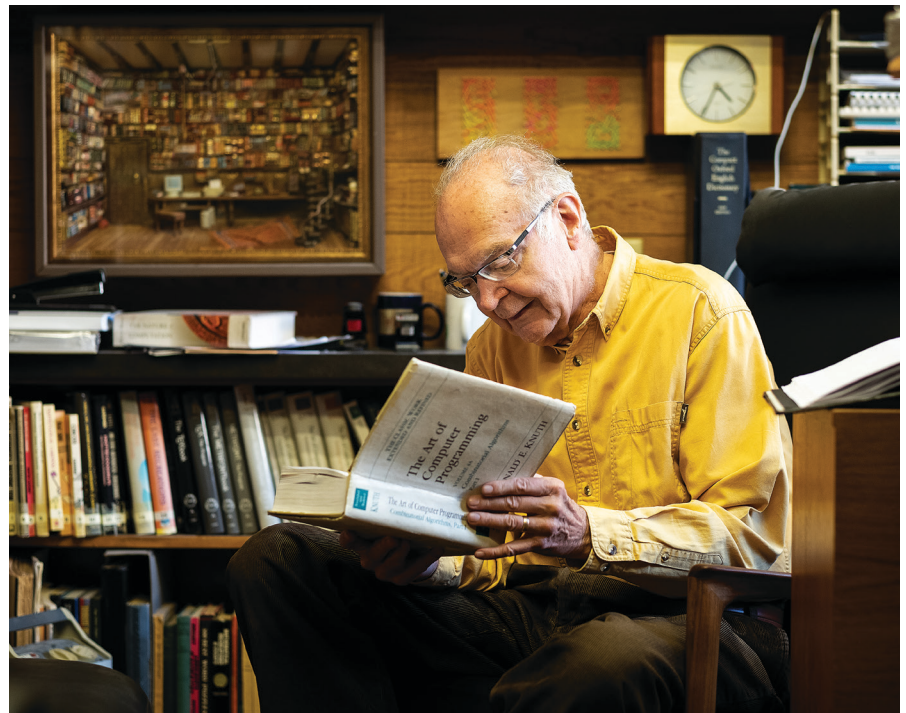
This Viewpoint, which we thank *Communications* Senior Editor Moshe Vardi for suggesting, is a condensed and highly edited transcript of the original talk that has provoked so much discussion.

a See <https://stanford.io/3qYDCce>

b See <https://bit.ly/3oTsKtY>

c See <https://stanford.io/2Wg2v4J>

d “The Tears of Donald Knuth,” Thomas Haigh, *Commun. ACM* 58, 1 (Jan. 2015), 40–44; <https://bit.ly/382aAQ7>



GIVING THIS TALK might be the greatest mistake in my life, because I'm going to talk about controversial things. I generally go out of my way to avoid argument whenever possible. But I feel so strongly about this that I just have to vent and say it.

Although there is “history” in the title, I'm not going to tell you about the history of computer science. Instead,

I'm going to talk about historians of computer science—about historiography. This is meta-history. I'm going to try to explain why I love to read works on history, and why I'm profoundly disturbed by recent trends in what I've been reading.

Why do I, as a scientist, get so much out of reading the history of science? Let me count the ways:

1. To understand the process of

discovery—not so much what was discovered, but how it was discovered. Primary sources are best: the words of somebody who discovered something, as they were discovering it. The more examples I see, the more likely I'll be able to discover something tomorrow.

2. To understand the process of failure. We learn a good deal from historical errors, not only from our own. It also helps to know that even the greatest minds are unable to grasp things that seem obvious to us. Leibniz spent much time working on combinatorics, and most of what he did was underwhelming and totally wrong.

3. To celebrate the contributions of many cultures. There are many ways of thinking, many points of view, and many independent researchers. Fibonacci numbers were discovered in India long before Fibonacci. Catalan numbers were discovered in China, a hundred years before Catalan. Many uneducated people have discovered wonderful patterns in numbers, and I can share their joy of discovery.

4. Telling historical stories is the best way to teach. It's much easier to understand something if you know the threads it is connected to. Give credit to Fibonacci, but also to Narayana in India. The complete story is of many separate individuals building a magnificent edifice with a series of small steps.

5. To learn how to cope with life. How did other scientists grow up, make friends or enemies, manage their time, find mentors, mentor others, and serve their communities? Balance is important.

6. To become more familiar with the world, and to know how science fits into the overall history of mankind. What was life like on different continents and in different epochs? The main difference between human beings and animals is that people learn from history.

I am grateful in particular to historians of mathematics. They make original source materials accessible through reprints, and through their translations of both language and notation. They scout out previously unpublished papers, letters, meeting minutes, and official records, and then link them together into a narrative. What I don't like is analysis of trends alone; I like to see the source materials up front.

So there is mostly good news from the historians of mathematics. The bad news comes from the historians of computer science.

What did it for me was an article by Martin Campbell-Kelly, a leading historian of computer science whose work I had admired. But his 2007 article on “The History of the History of Software”^e was a shock.

The centerpiece of the article was a table that classified selected works on software from 1967 to 2004 into four categories: technology; industry; applications; and institutional/social/political. At the beginning most published works are about the technology, but by the end they are mostly in the other categories. The author's description of the change is that “over time, software history has evolved from narrow technical studies, through supply-side and economic studies, to broad studies of applications.”

He thinks that is good! On the contrary, it is extremely shallow and completely non-technical. I broke down and started to cry. I finished reading it only with great difficulty because tears had made my glasses wet. I immediately dashed off a letter to Martin.^f

“I must confess that by the time I got to the last three or four pages, I was so upset that I could barely see straight. I had to force myself to read slowly, not believing you had succumbed so far to the alarming-to-me trends and fads of the moment about how history ‘ought to be’ written.

“Do you not see any blind spots in your outlook when your Table 1 shows 68% class T [technology] articles in the first 20 years, and 0% class T in the last five years ... and then you say ‘The table shows how the subject matter has broadened!’ The subject matter has not broadened; it has totally shifted. All we get nowadays is dumbed-down. Thank goodness historians of mathematics have not entirely abandoned writing articles that contain formulas or explain scientific ideas.

^e “The History of the History of Software,” Martin Campbell-Kelly, *IEEE Annals of the History of Computing* 29, 4 (Oct.–Dec. 2007); 40–51; <https://bit.ly/3oMC0jN>

^f Campbell-Kelly replied in “Knuth and the Spectrum of History,” *IEEE Annals of the History of Computing* 36, 3 (July–Sept. 2014); <https://bit.ly/3ninEXP>

“I am sure that business histories are as difficult to write as technical histories, and they are no doubt also as valuable to businessmen as technical histories are valuable to technicians. But you seem to be celebrating the fact that nobody writes technical CS history at all anymore!

“When you speak of ‘obvious holes’, you are thinking of obvious holes in business history ... the video game industry, for example. But how about the people who write video games: They invent marvelous breakthroughs in techniques about how to render scenes and pack data and do things in parallel and coordinate thousands of online users. The lack of anything even close to describing these techniques and how they were discovered and under what constraints seems to me a much more obvious hole; yet you show no inclination to admit its existence much less to suggest plugging it.”

Martin wasn't always that way. He describes in the article how, for his Ph.D. dissertation under Brian Randell at Newcastle University, he “managed to locate most of the system programs developed for the first three operational British computers—the Cambridge EDSAC, the Manchester Mark I, the National Physical Lab Pilot ACE. Studying these programs and their texts was utterly absorbing.” Absolutely! He could see why it was beautiful. He was doing the kind of history that I came to admire him for.

Then by 1976 he was starting to think about the broader picture. He didn't see how it was “concrete” the way subroutine linkage was achieved on the EDSAC, or how you got an index register in the hardware of a machine. He offers a “biographical mea culpa” and says, “what they (we) wrote looks constrained, excessively technical, and lacking in breadth of vision.” He's apologizing for what I always had admired!

Back to my letter:

“During the past 20 years, histories and expositions of mathematics for general readers have gotten dramatically better, while the analogous histories and expositions of computer science have gone downhill. With your Table 1 you could have generated a wakeup call. But instead you seem to be a pied piper for continuing the dismal trends. You have clearly lost faith

in the notion that computer science is actually scientific (as well as being related to economics and defense etc.). Yet I still cling to that old-fashioned belief ... indeed, if computer science were no longer a rich science with deep ideas, I could finish *The Art of Computer Programming* in no time, but it appears that I still have 20 years of work ahead!”

Well, that was 5 years ago,^g and I have 25 years of work ahead.

“You kindly state that it was OK and even fine for narrow-minded people like me to attempt to write history even though we have no training as historians, since there is a shortage of historians. Fair enough. But now you are encouraging professional historians to address only the masses of readers [...] and to ignore the 2% of the population who will spend their lives actually writing software. This you say is holistic and integrative. I view it as lightweight, mildly interesting; a chance to be witty and win some arguments so that another witty historian can challenge you and publish more lightweight stuff. Fine for employment of historians, but pretty much a waste of time for a reader who wants to know how to do hard science. The few papers I’ve written that have a historical component were among the most difficult I have ever done, and I greatly admire the historians who do it properly.”

I met Martin a few months later at a history meeting in England. We talked for several hours, but neither of us could get the other to agree. He keeps insisting that he wants his students to write no more books and papers of type T. Going back to my list of all the reasons why I love history, he’s saying that numbers one, two, three, and four aren’t important; only numbers five and six are of value.

I soon found out that historians of science have been debating this among themselves for a long time. They don’t call it “type T” versus something else; they talk about “internal history” versus “external history.” For them, internal history is written by and for people who are knowledgeable about some discipline, and the external histories are written for the masses. Internal histories, those of type T, have basically

^g 11 years now—Ed.

I am grateful in particular to historians of mathematics. They make original source materials accessible through reprints, and through their translations of both language and notation.

come into disrepute—except, I’m glad to say, with respect to mathematics.

How has mathematics managed to escape this so far? I suppose it’s because historians of math have always faced the fact that they won’t be able to please everybody. Historians of other sciences have the delusion that any ordinary person can understand it, or at least they pretend so.

There was one thing that Martin Campbell-Kelly and I definitely agreed on: that it would really be desirable if there were hundreds of papers on history written by computer scientists about computer science. Specialists like me are not writing the kind of papers that would fill the historical gaps. Martin says at least he wants professional historians to have some data from misguided people—like we who do the technical stuff—that they can clean up later.

He muses about why it is that there are almost no history papers being written now by computer scientists, and he says that it is probably peer pressure—that papers on history don’t get any academic points. In Britain they had the notorious “Research Assessment Exercise,” which was used to decide on salaries and promotions. History papers probably got no points in that assessment, and so nobody writes them. In America I don’t see support for such

papers either. I think it’s something that computer scientists ought to do anyway, even though it’s hard to write these historical papers, and hard to get exposure for them.

I want to end on a high note, with a tantalizing wish list about what we could do. The best way to write history is to combine breadth and depth. Not just the broad ideas from which you understand the context, but also to zoom in on a few places and provide specific examples with detailed analyses. Here are some of the many papers waiting to be written:

► **Operating Systems.** I have at home Edsger Dijkstra’s source code for the operating system he wrote in 1965. Nobody has looked at it, and we should.

► **Databases.** Early computer programs were filled with database ideas that have never really been analyzed and placed in context.

► **Rendering techniques for movies and video games.** Many great technical ideas were developed at Pixar and elsewhere, and you could make a great story about the history of the algorithms they’ve used.

► **Compilers.** In the early 1960s there were really interesting programs written at Burroughs and Computer Sciences Corporation that have never been analyzed. There was a brilliant programmer at Digitek who had completely novel and now unknown ideas for software development; he never published anything, but you could read and analyze his source code.

► **The Computer History Museum has Bill Atkinson’s source code, now released by Apple, for MacPaint and MacDraw.** They are brilliant programs, beautifully organized and structured, that are a treat to read and deserve to be annotated and studied.

And so on. There are many wonderful algorithms and source codes whose histories are completely untouched. If we technicians can study and explain them in depth, then historians will at least have material to which they can later add the breadth. ◻

Donald E. Knuth is Professor Emeritus of The Art of Computer Programming at Stanford University, Stanford, CA, USA.

Len Shustek (len@shustek.com) is Chairman Emeritus at the Computer History Museum, Mountain View, CA, USA.

Copyright held by authors.