Stanford University

2014 Kailath Lecture

“Let's Not Dumb Down the History of Computer Science”[1]

Professor Donald Knuth

May 7, 2014

The title of my talk today is “Let's Not Dumb Down the History of Computer Science,” and I have to start out essentially by saying that this might be the greatest mistake in my life to give this talk today, because I’m going to talk about controversial stuff. The truth is I actually hate polemics. I go out of my way to avoid arguments whenever possible. The worst kind of argument is about matters of taste, when everybody should be free to disagree. So how can I argue about something where a lot of people can have different ideas and taste? On the other hand, I feel so strongly about this that I just had to vent and say it. So anyway, here it goes. Of course it looks like there are enough people here that some of you might agree with me, anyway. [audience laughs]

Computer people have a word for what I’m going to do: we call it “flaming.” [audience laughs] I have to tell you in advance that I’m 76 years old now, and therefore I’m no doubt getting grumpier every minute.

My talk is going to be non-standard also in another way. There’s “history” in the title, but I’m not going to tell you about the history of computer science. I’m not going to tell you about what cool ideas are or who invented them or what. No,

[1] https://www.youtube.com/watch?v=gAXdDEQveKw

instead of talking about the history, I'm talking about *historians* of computer science. I'm talking about historiography. So this is meta-history. I'm going to try to explain why I personally love to read works on history and why I continue to spend considerable time reading historical works, and why I'm profoundly disturbed by recent trends in the things that I've been reading.

Now in order to get you all into a historical mood, you can see I'm using old-fashioned technology here. [audience laughs] I'm on purpose using an overhead projector instead of a computer. I'm not going back to punch cards, but… [audience laughs]

I want to start out upbeat, and so the first thing I want to do is give a testimonial for the historians who are doing it right. Why do I, personally, as a scientist, get a lot out of reading history of science? Let me count the ways.

First of all – and this is really number one for sure – to understand the process of discovery. Now some people would call it "invention" instead of "discovery"; I don't want to get into that kind of a debate. But what I mean is I want to know not so much *what* was discovered but *how* it was discovered. The best thing for me is to read primary sources – if possible, to find the words of somebody who discovered something as they were discovering it. The way they described it at the time. The original words. The more examples of this that I see about how somebody discovered something, the more likely I'll be able to discover something tomorrow. It helps my own research all the time in order to have more and more case studies of how people discovered things.

Number two is a little harder to read, but I'll read it to you – understanding the process of failure. This is almost equally important in a way. We learn a good deal from errors, not only our own errors but also just to see historical errors as well as successes. I myself have to learn not to be afraid of making mistakes, and then how to recover from mistakes and use them for guidance.

Also, another part of this which is not often mentioned is that when we see that some of the greatest minds of all time are unable to grasp things that seem obvious to us today, this gives us a better way to calibrate the intrinsic difficulty of ideas. Let's take Leibniz, for example. He spent a lot of his time trying to do combinatorics. And most of what he did when he was working on combinatorics was not only very underwhelming, but it was totally wrong. So I can be a much better teacher and writer if I can understand why it is that Leibniz didn't get it.

Number three: celebrating the contributions of many cultures. It's important to understand that there are many, many different ways of thinking. Also that particular concepts can be perceived from many, many different points of view. For example, Fibonacci numbers were discovered a few hundred years before, in India, long before the time of Fibonacci. Catalan numbers were discovered in China, a hundred years before the time of Catalan. Each discovery added to the other, though in different in parts of the world. Because they had different approaches, they had different motivations. Also, lots and lots of illiterate people from around the world have discovered wonderful patterns, and I can share with them the evident joy that they had when they found these patterns. So I love that aspect as well.

Number four: storytelling. Telling stories is really the best way to teach, to explain something. One of my main goals throughout my life has been as a teacher and writer, and so I want to reach other people and pass along the ideas that excite me. What's the best way to teach them? Human beings over the centuries have evolved to communicate with each other best by telling stories to each other. History gives us the stories. It's much easier to understand something if you have some thread that it belongs to. It lodges in your brain much better if you have the story to go with it. Now one of the motivations in this respect is to give credit where credit is due. For example, Fibonacci gets credit, but also so does Narayana in India.

But that's not the main thing about these stories. The main thing I try to understand is that the history of every field I've looked at is really not so much the story of great milestones and "What are the top 10 discoveries in this area?" But instead, the real story is that many, many separate individuals are working in collaboration and together they're building a magnificent edifice by a series of small steps. You think of the Great Wall of China, and lots and lots of stones went into that wall. That's the way there's a "Great Wall of Computer Science" and so on happening. So that's a great story.

Number five: learning how to cope with life. Besides understanding technical details about scientific things, of course I also want to see how other scientists grew up, how they made friends, how they maybe made some enemies. How did they manage their time? Did they like music? What did they do for hobbies? Who did they get mentorship from? Who did they mentor? What service did they do to community? All these things. Science is a major part of my life, but it's less than half. Balance is important.

Number six out of six here is: become more familiar with the world. This is just in general to know how science fits in with the overall history of mankind. What was life like for people on different continents in different epochs of time? It seems to me that the main difference between human beings and animals is that people like us learn from history. We have a sense of history, so we know what happened in past times much more than Tom's dog did, for example. [audience laughs]

So I'm grateful to historians, especially historians of mathematics, because of the things that they do better than anybody else. The first thing I'm really thankful for is that they can make source materials accessible. As I say, I like to see the words of the originators -- what they're doing. A lot of times, or most of the time, if you go back in history, it’s not in English. So somebody has to do a good job of translating. I can learn maybe German and French and Russian, but I don’t have time to learn Hungarian and Japanese and all these other languages, not to mention Indian languages, which are about a hundred of them. So historians can bring that, make it available to scientists like me.

Also, notational barriers. The people didn’t only use different words, but they used different symbols. It’s a trick to learn. We can sort of put ourselves into the mind of these inventors, even though they were using different symbols, but somehow not being totally unable to understand because the symbols are so different from ours. So historians can do this great.

Then, scouting out archival documents. Here there’s a great talent for finding letters, finding unpublished papers, other archives, the civil records, medical records, financial records, minutes of meetings. All kinds of scraps of information exist somewhere in the world, and historians have been really good at smoking these things out.

Then, linking things together. Also they know a lot about what else is going on at the same time in previous times, and so a good historian can do this.

The thing that I don’t like particularly is if a whole paper is just analysis of trends. After they smoke this out, I still like to see source materials. I like to see exact words -- what did somebody say to somebody in this letter -- instead of just an overall thing. The things that historians probably work hardest on in order to analyze these trends are the things that are of least interest to me. But I love the ability to bring these other source materials up front.

Okay, so this is the good news. Now I have to turn to the bad news. The good news came mostly from historians of mathematics, and the bad news is going to come from the history of computer science.

Here’s what sort of did it for me. [audience laughs] A shock. Five years ago, I was reading this paper. You can see the title: “The History of the History of Software.” This paper was actually published seven years ago, but I’m a little slow reading the literature. Anyway, one day I came across this article and I thought, “Oh good. It’s by Martin Campbell-Kelly.” I’ve always liked him, liked his work. He’s kind of the leading historian of computer science.

Well, he didn’t write this paper for people like me. He wrote it for historians. It got published later, because somebody said, “Oh, historians liked it,” and so they said, “Okay, we’ll publish this in the journal *IEEE Annals of the History of Computing*.” Anyway, this gives you some insight as to what historians say when they’re talking to historians instead of when they’re talking to people like me.

So that was a shock because the main thing, sort of the centerpiece of this article, is a table that he gave. I don't expect you to be able to read this, but I just want to show you the table. It appears on two pages. Can you sort of see? He takes excerpts from up to three papers from each year and mentions some of the key articles that were written about history of computer science, or history of software in this case actually, in that year.

The point is he classified these papers into four types. He had type T, which is technical. He had type S, which stands for supply-side; it means industries, the study of industrial things, economics. Type A was applications; what people have actually done with the software. And then type I was stories of institutions; social, and political aspects.

We start out in 1967. We have one paper called type T. In '69, he has three papers, all type T, and so on. T, T, T, T, T, T, S, T, T, and so on. Now we get down to 2004, at the end, and you see what it is – I, I, S, I, S, I, S, I, S, S, A, A, I, I. So you see, there's been a difference in the way people wrote history of computer science.

And how does he describe that difference in his paper? Well, here in the left column he says, "This table shows how the subject matter has broadened. In the 1960s and '70s people wrote about technology, code, software engineering practices. Starting in the '80s people began to write about software as economic activity," and so on. Then he gets on, and he says, "Over time, software history has evolved from narrow technical studies, through supply-side, economic studies, to broad studies of applications."

The point, the thing is, he thinks that's good. [audience laughs] I grant that it's broad, but it's also extremely shallow. I mean, it's completely non-technical. I have to admit that when I was reading this page, I broke down and started to cry. I finished reading it only with great difficulty because tears had made my glasses wet.

So I immediately dashed off a letter. Actually my glasses are still wet when I wrote this letter. [audience laughs] But I'll read you part of it now.

"Dear Martin, I had always greatly admired your writing about CS history until today when I came across your recent article '(The History of)^2 Software'." [audience laughs] Yeah. Don't laugh. I'm writing this in tears, you have to understand. [audience laughs] "Alas, I must confess that by the time I got to the last three or four pages, I was so upset I could barely see straight." I didn't want to tell him about the tears. [audience laughs] "I had to force myself to read slowly, not believing you had succumbed to the alarming-to-me trend 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 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. It has not broadened; it has totally shifted. All we get nowadays is dumbed-down."

Then I say, "Thank goodness historians of mathematics have not entirely abandoned writing articles that contain formulas or explain scientific ideas rather than just sticking to things like the strategies a mathematician has used in order to get into academy or something.

"I'm 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, and you mention the video game industry for example." To me, okay, there's lots of great papers that ought to be written about video games. "The people who write these video games have invented marvelous breakthroughs in techniques about how to render scenes, how to pack data in a small space, how to do lots of things in parallel, how to coordinate thousands of online users. All of these are brilliant ideas that make great stories. But the lack of anything even close to describing these techniques and how they were discovered and under what constraints they were discovered seems to be a much more obvious hole, but you've shown no indication even to admit its existence and much less to suggest plugging it."

So I wrote this letter. I think I'm going to read you a little more of his article just to show you, again, what I was complaining about.

At the very beginning of his article, Martin says, "I have used the word 'technical' several times. Up to the late '70s software history was almost exclusively technical. My dissertation…" – his PhD work under Brian Randell at Newcastle – "…I 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 loved it at that time. He could see why it was beautiful. "I also interviewed several of the pioneers of these early systems (they were in their 50s or 60s), I studied all of the derivative machines up to '55." In other words, he was doing the kind of history that I came to admire him for.

Then he gets to the second page – and remember he's talking to historians – and he says by the time 1976 came along, at the time when he was starting to think about the so-called broader picture, he didn't see how it was "concrete" the way "subroutine linkage was achieved on EDSAC" or how you got an index register in the hardware of a machine.

Then he says, “Well, this biographical *mea culpa* is not intended as an exercise in humility.” He’s apologizing. He’s saying, “You know, I’m guilty of writing this technical history. You have to understand, I was just a young man. I was making like foolish, you know?” He writes, “With 20/20 hindsight what they (and we) wrote looks constrained, excessively technical, lacking in breadth of vision.” Okay, so here he’s apologizing for what I always had admired.

Later on in the article he gives an example of what he should have done. It turns out that he wrote a book that came out in 2003 that is called *From Airline Reservations to Sonic the Hedgehog*. There’s a guy named Mr. Perkins, Harold Perkins, who gave kind of a poisoned review in the *Times Literary Supplement*. I think Martin was wounded by this. But anyway, here’s what the reviewer said.

“Campbell-Kelly is a master of technical detail and the alphabet soup of acronyms. But like most specialists in an arcane activity, he has tunnel vision and provides little social context.” This man Perkins, by the way, I guess is a social historian. [audience laughs] “He does ‘internalist’ history, rather like old-fashioned art history or history of science -- full of innovators and heroes driven by creative opportunism. The impact of the computer industry on society, on the way people live and communicate, is largely left to the reader’s imagination.”

Actually this book doesn’t really get much into technical details either. It still doesn’t have much depth. Anyway, “Even the state and military applications are touched on, rather than explained. The computer and its software nervous system brought a revolution in human development as significant as the steam engine, the automobile or the aeroplane, and even more effective in shrinking the planet. This technically expert book is rather like old railway history written by railway buffs who know the number of wheels and the horsepower, the names of the engineers and companies, but [they] take for granted how [railroads] changed the world.”

Of course! Do I want to buy a book that tells me that “Hey, gosh, computers have changed the world”? What else is new? [audience laughs] Why don’t you tell me something interesting?

It’s nice to know that people like Mr. Perkins are starting to get it -- that there’s something interesting going on -- but it’s obvious to me that he just didn’t have a clue what this book was about. He has his – what do you call it? – his dignity, so he wants to show that he can do it better, even though he doesn’t know anything about the field, by just saying, “Well…” It’s just like people who say, “I never was very good at mathematics; it must be that mathematics isn’t important –I got through.”

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 wake-up call. But instead, you seem to be a pied piper for continuing these dismal trends. You have clearly lost faith in the notion that computer science is actually scientific, as well as being also coincidentally related to economics and defense and so on. Yet I myself still cling to that old-fashioned belief. Indeed, if it were true that computer science were no longer a rich science with deep ideas, I could finish *The Art of Computer Programming* in no time. [audience laughs] But it appears that I still have 20 years of work ahead!" Well, that was 5 years ago and I have 25 years of work ahead. [audience laughs]

"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 like Mr. Harold Perkins", and to ignore the scientists who are going to make the thing live for a few more years.

I do have another page to show you here. [audience laughs] I talk about "the 2% of the population who will spend their lives actually writing software." Here I'm talking about John Hennessy and others like us who are drawn to this mindset that happens to correlate with computer science. "This you say is holistic, 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. That's fine for employment of historians, but it's pretty much a waste of time for a reader who wants to know how to do hard science.

"The few papers that I've written myself that have a historical component were among the most difficult I've ever done in my life, and I greatly admire the historians who do it properly. (In mathematics I've just purchased superb sourcebooks by Jackie Stedall, and Eleanor Robson, Victor Katz and so on.) I think Steven Levy's books on *Hackers* and *Artificial Life* are wonderful. I hope you will come back to technicalities that displease Mr. Perkins once in a while, because you are one of the few who are able to handle them well.

"There are millions of lines of brilliant code in existence" and so on and so on.

So Martin replied to me. I soon learned, and in talking with other colleagues, that… Well, okay. By the way, I also met him a few months later at a history meeting in England, and we talked for several hours, but neither of us could get the other one to agree; he keeps insisting that he does want his students to write no more books of type T, and no more papers of type T. Going back to my earlier slide then where I listed all the ways why I love history, he's saying number 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, these type T, have basically come into disrepute except, I'm glad to say, with respect to mathematics.

This is where Martin says these type T things have a limited value. But, well, everything has a limited value. Have you ever read something that has *unlimited* value? I mean what the heck. [audience laughs] Okay.

For example, this has really infected all of history of science. The main journal in history of science is *Isis*, founded long, long ago. At the time that I did this, I went to the library and I looked up *Isis* from 50 years ago and then I compared it with *Isis* from 2008. There was a sea change that had gone on. If you look in 1958, the articles actually went into a lot of technical details in whatever it was, chemistry, biology. But now we go in 2008, it was almost always about strategies, funding, something like this.

Here's an example from 2008: the review of a book called *Beyond Artificial Intelligence*. The reviewer says, "History of science awaits a comprehensive history of the decision sciences – in which artificial intelligence is an important actor, but which also include cybernetics, information theory, management science, and operations research. For the time being, we will have to make what we can of the limited perspective provided by practitioners' accounts such as *Beyond AI*." Now, I don't know -- I leave it as an exercise why on earth anybody would really want to combine cybernetics, information theory, management science, operations research, and artificial intelligence into a single book, but that's because of my cramped perspective.

Now, 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 actually an ordinary person can understand physics, or at least they pretend it.

Last weekend I read an obituary of a distinguished mathematical historian, Christoph Scriba, who died last July. Scriba was probably the leading expert on John Wallis, a 17th century mathematician who was, I think there's no doubt, the closest to becoming a computer science of anybody in the 17th century. If you learn more about John Wallis, you'll see that he sort of speaks our language in those days. Scriba was a disciple of Joseph Hofmann, a Leibniz scholar, and in 1971 Scriba wrote an essay dedicated to Hofmann about the trends in the "History of Mathematics in the Mirror of Time." He's considering whether or not history of mathematics should shift over to be more of a social history.

The obituary says that “While recognizing the value of social history and the history of ideas, Scriba remained largely wedded to the approach he had inherited from his academic mentors.” Thank God. Now Scriba did write that he often had trouble making his work comprehensible to uninitiated. But he also complained that a lot of mathematicians who could understand it also couldn’t care less about what he wrote. But that’s another problem. That’s shame on us for not appreciating science the way we should.

Now there was one thing that Martin Campbell-Kelly and I definitely agreed on, and that is that it would really be desirable if there were hundreds of papers written now by computer scientists about computer science. It seems that actually specialists like me are not writing the kind of papers that would fill the gaps, and we’re somehow assuming that the historians of mathematics should all of a sudden learn how to do this -- learn computer science instead of just learning how to scout through archives. Martin says at least he wants professional historians to have some data that they can clean up from these misguided people like we are who do the technical stuff.

He muses: why is it that there are almost no papers being written by computer scientists now? He says that probably it’s peer pressure, that the papers on history don’t seem to get any points. In Britain, they have something notorious called “British Research Assessment Exercise,” which is used to decide about salaries and promotions, and somebody who writes a history paper probably gets no points in that, and so nobody writes it. In America I really don’t see support for such papers either. I think that’s something that computer scientists ought to do, even though it’s hard to write these historical papers; we should face that fact, even though there’s no conference on it or something, that we can get it into.

I want to end on a high note again – with, I hope, a tantalizing wish list about what we could do. What do I think is really the best way to write history? I’ve always tried to combine breadth and depth. Not just shallow and broad, and not just deep, but a little of both. Somebody made a wonderful definition of a liberal education once, that said that you know something about everything and you know everything about something. You’ve got to have the breadth, you’ve got to understand context of stuff. But then in order to really understand, you got to zoom in one or two places and just look at that as an example from which people have something to hang onto. I used this; for example, I wrote a book called *3:16* where I try to say that the Bible is a really complicated subject, but we can learn a lot more about it if we zoom into 15 sort of randomly chosen parts and see what people have said about those parts over the years. So I like this idea of “Breadth + Depth.”

Now I want to mention just some -- scratch the tip of the iceberg on some -- of the many papers that are waiting to be written.

The concept of operating systems. I have at home: Edsger Dijkstra sent me the source code for the operating system that he wrote in 1965. Nobody has looked at that. You want to see exactly what kind of code was written in those days.

Concepts of databases -- the same thing. Early computer programs were filled with ideas that have not ever been really analyzed and set in context.

Rendering techniques for motion pictures, your video games, all these things. So many great ideas are being used at Pixar and all over. We don't want to just see the movies. Computer people, you can make a story that you can explain to ordinary people, and zoom in and show some of the things that go beyond the effects that you see on the screen. You don't just have to see the cool workspaces that these people work in, you can also see the algorithms they're using.

I've got lots and lots of software that I collected over the years. In the early '60s there were really, really interesting programs written for compilers at Burroughs Corporation group, and another group at Computer Sciences Corporation, especially the one they did for LARC at Livermore. There was a brilliant guy – he never published anything, except you can read his source code -- who was at Digitek Corporation. He had completely different ideas for software development that deserve to be resurrected today, I think.

The Computer History Museum has Bill Atkinson's source code released by Apple for MacPaint and MacDraw. Really brilliant programs, beautifully organized and structured. A treat for anybody to read.

On and on and on. There are so many wonderful source codes that are completely untouched. If we technicians can explain these, then maybe some historians will come along and add the breadth that we lack.

That's my message today. Thank you very much for listening. [audience applauds]

[end of recording]