DOI:10.1145/2786922

Vinton G. Cerf

Milestones

Last month I wrote about how much we owe to John White, not only for his 17 years of service to ACM as CEO, but also his earlier services in various ACM volunteer positions,

including president. Now I want to draw attention to the fact that ACM will be 70 years old in 2017. It is not too early to be thinking about how we might usefully recognize this milestone. We had a remarkable celebration we called ACM 1 in 1997 as we reached the 50th year of our existence and while 70 is not the same as, say, 100, it is a significant moment and worthy of some serious thought. Such moments encourage all of us to think about what we want to achieve in the future, even as we look back on the many years of our existence and accomplishments.

One immediate thought is to try to compose a list of, say, 25 problems in computer science that remain unsolved—rather like David Hilbert's 23 problems in mathematics published in 1900.^a What might be on our list? I am not sure how to formulate this, but one thing on my list is to ask whether we can ever get to the point where we can write bug-free code in a reliable fashion! There are other questions having to do with computability and computer design. Will new machine architectures permit some form of computation that goes beyond the reach of the Turing Machine? Are there limits to quantum computation? Scott Aaronson tackles that question in a 2008 *Scientific American* article.^b What do we not know about the fundamentals of computation and pro-

gramming, in their largest sense, that we should seek to understand? What other questions would you put on such a list?

Another thing we might consider is how we think computing (and communications) will change over the next 70 years. The Internet, the World Wide Web and the Internet of Things are already a part of daily life for about 40% of the world's population and smartphones for an even larger fraction. Can we try to imagine what the next major changes might be? Will nano devices, programmable and autonomous robots, machine learning and artificial intelligence have as large or even greater impact than what has gone before? Will programs learn to evolve themselves to pose and solve new problems? Do we face an optimistic future in which software artifacts are seen as partners or are the darker predictions of the replacement of humanity with smarter robotic software more credible?

What will be the state of affairs for computing in 2017 and how should we help to shape its future? What is the role of ACM and its members?

Should we be concerned about the increasingly pervasive role of software in our lives? Should we be learning about and teaching some form of “responsible programming?” Should there be incentives for software makers and providers to assure that credible measures have been taken to maximize safety and minimize risk to those using or dependent on software? What might those incentives look like? What should we do about preserving digital content including the software needed to correctly interpret it? What are the implications for intellectual property protection and for the interests of the general public? What is the future of ACM's Digital Library and the concept of open access to publications, data, and software?

The notion of “stock taking” seems very applicable here. What will be the state of affairs for computing in 2017 and how should we help to shape its future? What is the role of ACM and its members? What about the millions of professionals (and amateurs) with an interest in computing who may not be members but are affected by ACM's work? It is not too early to begin thinking about 2017 and, as we wish John White a fond farewell and his successor a warm welcome, let us begin the process of formulating a meaningful and substantive recognition of that milestone year. I await your suggestions with great anticipation!

Vinton G. Cerf is vice president and Chief Internet Evangelist at Google. He served as ACM president from 2012–2014.

Copyright held by author.

a http://en.wikipedia.org/wiki/Hilbert%27s_problems

b Scott Aaronson, The Limits of Quantum Computers, *Scientific American* 298, (2008) 62–69; doi:10.1038/scientificamerican0308-62.