Remembering Michael S. Mahoney

Martin Campbell-Kelly
Warwick University

Michael S. Mahoney, professor of the history of science at Princeton University, died in 2008. Born in 1939, Mahoney was already a seasoned historian of mathematics when he became one of the first senior historians to take an interest in the history of computing. He was by no means the first: for example, individuals such as I. B. Cohen at Harvard University and Derek de Solla Price at Yale University had been interested since the 1960s. Moreover, several institutions were already actively engaged: the Smithsonian Institution, the Charles Babbage Institute (founded 1979), the Computer Museum in Boston, and similar organizations in Europe. A journal, the *Annals of the History of Computing,* was established in 1979. However, Mahoney's unique mission was not so much to *do* the history of computing, as to see it done well.

The collection of papers on the history of computing, admirably edited by Thomas Haigh, stands as a memorial to Mahoney.[1] It is arranged in three categories which effectively define Mahoney's contributions to the discipline: the shaping of the history of computing; constructing a history of software; and the structures of computation. Haigh has provided an excellent, and by no means uncritical, commentary on the collection. In this essay, I will follow Haigh's organization.

1. The Shaping of the History of Computing

Mahoney burst on to the scene with a paper "The History of Computing in the History of Technology" published in the *Annals of the History of Computing* in 1988. In this somewhat polemical paper he urged both historians and computing practitioners to take up the challenge of doing his-

1. Michael Sean Mahoney, *The Histories of Computing,* edited with an introduction by Thomas Haigh, Harvard University Press, 2011.

Perspectives on Science 2013, vol. 21, no. 3

doi:10.1162/POSC_a_00104

tory well. The paper is highly characteristic of Mahoney's rhetorical style, and to read it is to hear his voice. Mahoney loved teaching, especially undergraduate and lay audiences, and he had a voice that projected far. Unknowingly he tended to use the same rhetorical boom regardless of the size of his audience or auditorium. I always tried to sit at the back where the volume was lowest. This paper was delivered at a time when the literature of the history of computing was quite thin. Today, twenty five years later and with the publication of hundreds of scholarly articles and scores of monographs, the paper is a snapshot of an emergent discipline. Mahoney played an important part in establishing the validity of this discipline.

Mahoney's comments were often addressed to practitioners rather than historians. Indeed he had an admiration and affection for computer professionals. Typical was his advice on doing history well given to speakers at the History of Programming Languages Conference in 1996. One of his recurrent themes was expressed here quite brilliantly:

> When scientists study history, they often use their modern tools to determine what past work was "really" about; for example, the Babylonian mathematicians were "really" writing algorithms. But that's precisely what was not "really" happening. What was really happening was what was possible, indeed imaginable, in the intellectual environment of the time; what was really happening was what the linguistic and conceptual framework then would allow. The framework of Babylonian mathematics had no place for a metamathematical notion such as an algorithm. (p. 39).

As it happened, in 1972 the world's most famous theoretical computer scientist Donald Knuth had written a widely admired paper "Ancient Babylonian Algorithms" which took exactly the line that Mahoney discouraged. Not many computer scientists would have criticized Knuth, albeit obliquely; this was a powerful message. Mahoney probably never realized it, but he was quite an intimidating authority figure and his strictures had the potential to discourage amateurs from engaging with the history of their subject. This is not to say that Mahoney was personally intimidating—far from it, he was kind and encouraging in person—but he carried everywhere the prestige of a professorship in one of the world's great universities.

2. Constructing a History for Software

Mahoney was primarily interested in theoretical computer science and so-called systems software (that is, infrastructure software such as operating systems). This interest tended to constrain his view of the broad spectrum

of software, which properly includes applications software and its economic exploitation and delivery. His perception of software was also distorted by a fixation on the Unix operating system and its creator Bell Labs. Mahoney's software perspective used to put me in mind of Saul Steinberg's famous *New Yorker* cover "View of the World from Ninth Avenue." For Mahoney, the view from Princeton University seemed to have Bell Labs and Murray Hill in the foreground, while Microsoft and the like were mere dots on the horizon. Mahoney interviewed a number of the software scientists at Bell Labs with the proclaimed intention of writing a book about the software crisis of the 1960s. It comes as no surprise to those who knew him that the book never materialized. He was full of boyish enthusiasm for such grand projects but he never had the single-mindedness to see them through. His life was filled with lecturing, mentoring, administering, and socializing, and he did not have the temperament for extended periods of solitude in gloomy archives. One can hardly criticise his priorities: there are more than enough books but too few historians with Mahoney's sociability.

The papers in this section highlight a general problem with the *Histories of Computing*—the repetition of the same message to different audiences. Most historians repurpose their material in this way, of course, but when the same point is made three or four times between one pair of covers it does the author no service. To take one example, Mahoney was fascinated by the way in which software engineers used and misused historical examples to validate and inspire their practices. Practitioners frequently tried to apply Ford's mass production techniques to software and used metaphors such as "software factory" and "software components." Mahoney commented on this theme repeatedly, but never explored it in any depth and one wonders why he did not. It was one of Mahoney's singularly annoying vices that he would set agendas for others but never seemed to do any of the ground breaking.

3. The Structures of Computation

The Structures of Computation is the editor's catch-all for Mahoney's papers on the relationship between mathematics and computer science. Easily the most accessible paper of the set is "Computing and Mathematics at Princeton in the 1950s," describing John von Neumann's IAS Computer and its significance. Considering he wrote as a denizen of Princeton University, it is splendidly modest in its claims. I heartily recommend it as a corrective to the partisan claims in George Dyson's *Turing's Cathedral* (2012) which has received so much uncritical attention.

Another paper "The Search for a Discipline of Computer Science" is much less accessible. Indeed, most readers, whether or not trained in com-

puter science, will find it impenetrable. The paper originally appeared as a chapter in the published proceedings of a conference on "The Space of Mathematics: Philosophical, Epistemological, and Historical Explorations" held in 1990. To be fair, Mahoney's was by no means the only unintelligible paper at that conference. One of the difficulties of the history of mathematics is that much of it is incomprehensible to non-mathematicians. I once thought that the history of computing might end up that way, but thanks to Mahoney and many others it has matured into a broad-based and inclusive discipline, although the history of theoretical computer science inevitably caters to a very narrow interest. This particular paper is rather revealing of Mahoney's modus operandi: as with virtually all his output in the history of computing one would classify it as scholarship rather than research. The paper is wide ranging but quite shallow (which was not inappropriate for the venue at which it was read, as it happens) and it is largely based on secondary sources and the surface technical literature. What the paper lacks is any sense of engagement with primary sources. For example, a good fraction of the paper concerns the collaboration of the British computer scientist Christopher Strachey and the American logician Dana Scott in the 1960s and 1970s. This was a subject that very much interested Mahoney, and Strachey was one of his heroes. Strachey died as professor of computation at Oxford University in 1975 leaving a massive archive now housed in the Bodleian Library. There is vast research potential in this archive on the very subjects that so fascinated Mahoney. How on earth was he able to resist spending some time there?

The answer I think is clear. Mahoney was above all a communicator and teacher rather than a researcher. Indeed, it always seemed to me that he was not quite in his element at Princeton University. I felt he would have been more at home in one of the United States' great liberal arts colleges—Dartmouth or Williams College, say—where there was more emphasis on collegiality and less on research. Of course, Mahoney's sociability and collegiality was hugely valued at Princeton and he was much liked by his colleagues. This must have made his academic life one of great personal satisfaction.

So, one question remains. Who is this book for? Clearly the book stands as a memorial to one of the leading advocates for the history of computing who was held by colleagues in an equal measure of admiration and affection. But beyond these colleagues, who would want to read it? *The Histories of Computing* should certainly be required reading for anyone embarking on a study of the history of theoretical computer science. It covers the ground well, and offers valuable research questions and directions in the spirit of traditional history of mathematics. To a lesser extent the book would be useful background for studies in software technology. His dis-

cussion of the software crisis of the 1960s is well worth reading and his insights into software engineering practice raise interesting research questions.

However, for those with a broader interest in the history of computing—such as in business and scientific applications or the development of information networks—the book cannot really be recommended, except perhaps that the rising generation ought to know something about Michael Mahoney, one of the shapers of the field. Mahoney's reflections of the 1980s no longer reflect the state of the field, but they do make interesting reading; we have come a long way in 25 years. Without a doubt, Mahoney left the history of computing in better shape than he entered it.