

The History of the History of Software

Martin Campbell-Kelly
Warwick University

Just as the fields of software and hardware development have evolved, the field of software history has likewise matured. At first, the history of software was exclusively focused on technology. Later, there were historical explorations of the software industry and professions. Today the emphasis is on applications and the societal changes resulting from software.

Allow me to begin with a small piece of personal history, which I hope illustrates a more general point about software history.

Starting in 1976, I undertook a doctoral study on the early development of computer programming in Britain. This was to prove a highly technical study that traced in considerable detail the programming systems invented in Britain in the period 1945–1955. I was then employed as an instructor at Sunderland Polytechnic, a second-tier college in the north east of England. I had the good fortune to be supervised by Brian Randell at nearby Newcastle University. Randell was a technical computer scientist and software engineer with a strong interest and commitment to the history of computing—he had recently edited a seminal collection of technical papers *Origins of Digital Computers* (1972).

Note that I have used the word “technical” several times. Up to the late 1970s, software history was almost exclusively technical. My dissertation was about code and texts. In my research, I managed to locate most of the system programs developed for the first three operational British computers—the Cambridge University EDSAC, the Manchester University Mark I, and the National Physical Laboratory’s Pilot ACE. Studying these programs and their texts was utterly absorbing. I also interviewed several of the pioneers of these early systems (they were then mostly in their 50s or 60s), and I studied all of the derivative machines up to about 1955. In the end, my dissertation encapsulated in a single diagram the complex processes of invention and technology transfer. At the time I thought I had done a tolerably good job. I published four lengthy papers in the *Annals of the History of Computing*,¹ and only a change of job to Warwick University and an itch to move on to fresh fields prevented my producing a mono-

graph. When I look back at my dissertation today, however, I cannot do so without a mild flush of embarrassment. Let me explain why.

In addition to Brian Randell as my supervisor, I felt the need for a second advisor who was a professional historian of science (the history of technology was little recognized as a discipline in Britain at the time). Margaret Gowing, the recently appointed professor of the history of science at Oxford University, agreed to act in this role. Gowing was an economic historian by training, and she had been a driving force behind the civil history of World War II, and subsequently wrote a number of classic works on the development of atomic energy in Britain.

I sent Gowing the draft of a paper I had written about programming for the EDSAC (and which was subsequently the first of my papers published in the *Annals*). She wrote back to the following effect: What you have written is clearly very good. I know practically nothing about computers, but I can tell that what you have written is good history, so far as it goes. However, let me urge you to look beyond programming technology to consider the kinds of people who were using computers and the problems that they were solving.

I was completely floored by this suggestion. I could not see in 1976 how to integrate the story of users into the history of software. I was not being obtuse or myopic; I just could not see how it was possible. I remember thinking it would be like writing about a bunch of miscellaneous application programs. What would be the point? What would be the coherence? Beyond adding a few sentences about users to my paper, I buried my head in the sand, and pressed ahead with my instincts.

Looking back, some years later, I began to see the point Gowing was making. Today it seems frankly bizarre that I chose to write

about the internals of the programming systems and not about the applications. Just consider some of the tasks that the EDSAC was used for.² In 1950, J.C.P. Miller and David Wheeler used the machine to generate the world's highest known prime number. In the mid-1950s, John Kendrew used the EDSAC in the elucidation of the myoglobin module, for which he shared the Nobel Prize in 1962. Martin Ryle used the EDSAC and its successor EDSAC 2 for the reduction of radio telescope data, a crucial process in radio astronomy, which he acknowledged in his 1974 Nobel lecture. The users of the other British machines also created path-breaking applications.

For example, at Manchester, Alan Turing did his first numerical experiments on morphogenesis (the growth of living forms) and produced printouts that looked remarkably like cow-spots, and which we can now see as lifting a corner of the curtain on dynamical systems.³ Christopher Strachey wrote a truly sophisticated program for playing checkers, some years ahead of Arthur Samuels in the US. At the National Physical Laboratory, researchers were conducting some of the first experiments on digital applications, ranging from esoteric machine translation to a commission from the government to explore the potential for the automatic computation of income tax.⁴ Also at the Laboratory, Jim Wilkinson (an ACM Turing Award winner in 1971) investigated errors and stability in digital numerical methods and developed advanced matrix programs. These were to prove vital in understanding and preventing “flutter” for the British aircraft industry, which was still reeling from the de Havilland Comet air disaster of 1954.

Yet in 1976, I just could not see that these applications related to software as such. Certainly there seemed to be nothing as concrete as how subroutine linkage was achieved on the EDSAC or the B-line index register was used on the Manchester Mark I.

Well, this biographical *mea culpa* is not intended as an exercise in humility. Rather it is intended to show that the world of software history was different in the 1970s. The people who wrote software history at that time, and earlier, wrote the best history they could with the knowledge, understandings, and the background they had at the time. With 20/20 hindsight, what they (we) wrote looks constrained, excessively technical, and lacking in breadth of vision. But that's not how it seemed at the time.

I believe it was difficult to write about application software in the 1970s for two reasons. First, software was not nearly such a ubiquitous concept in the 1970s as it is today. For example, the software industry was very small—less than one-twentieth of its current size. Second, we did not think that software characterized applications in quite the way we do today. We thought of an application as comprising systems design, mathematics, fulfillment, and so on, whereas software was just code. Indeed, I don't think that books such as James McKenney's *Waves of Change* (describing the SABRE airlines reservation system) or Steven Levy's *Insanely Great* (describing the Macintosh development) would have been classified as software books at all; but today we would certainly characterize them as such, because we now see software as comprising so much more than code. These two books, and others like them, describe the process by which software came to orchestrate standard hardware in unique ways.

Evolution of software history

To portray the evolution of writings about software, in Table 1 and the “Bibliography: Software-Based Information Processing” sidebar I have listed what in my experience were the most useful or important publications on software history every year since 1967, when distinct publications on software history began to appear. This is not a scientific survey—I have simply gone through the bibliographies of my own writings on software and culled the items I personally found most useful, up to three items for each calendar year. Some years there were no publications at all; other years—rather like London buses—several came along at the same time. By the 1990s, it was sometimes difficult to choose just three, the field was so much enriched. Some might say that not all the works I have selected are about software as such—perhaps, but I certainly used them in my writings.

The bibliography is not an exhaustive reading list. It is more in the nature of a subjective sampling of the literature, with the constraint of selecting a maximum of three items per year. A few of the works are outstanding, such as Jean Sammet's *Programming Languages* (1969) and Claude Baum's *The System Builders* (1981). Others are not particularly good at all; they are simply the most useful items published in that year. Incidentally, the great majority of works are by American or British authors. There are not, as yet, many works on software history published

Table 1. Selected works on software and software-based information processing, with shortened titles. (For full titles, see the "Bibliography: Software-Based Information Processing" sidebar.)

Year	Type*	Author and Title
1967	T	Rosen, <i>Programming Systems & Languages</i>
1968		
1969	T	Sammet, <i>Programming Languages</i>
	T	Rosin, <i>Supervisory & Monitor Systems</i>
	T	Knuth, <i>Art of Computer Programming</i>
1970	T	Knuth, "von Neumann's First Program"
1971		
1972		
1973		
1974		
1975	T	Brooks, <i>Mythical Man-Month</i>
1976	T	Wexelblat, <i>History of Programming Languages</i>
	T	Wegner, "Programming Languages"
1977		
1978		
1979	S	Phister, <i>Data Processing Economics</i>
1980	T	Knuth, "Early Programming Languages"
	S	Welke, "Origins of Software"
1981	S	Baum, <i>System Builders: Story of SDC</i>
1982	T	Campbell-Kelly, "Foundations of Computer Programming"
1983	S	Fisher et al., <i>IBM & US Data Processing Industry</i>
1984	S	Frieberger & Swaine, <i>Fire in the Valley</i>
	T	<i>Annals</i> , special issue on Fortran
1985	S	Forman, <i>Computer's Promise: History of Informatics</i>
	T	<i>Annals</i> , special issue on Cobol
1986	T	Bashe et al., <i>IBM's Early Computers</i>
1987	S	MacNeal, <i>MacNeal-Schwendler Corp.</i>
1988	T	Goldberg, <i>History of Personal Workstations</i>
1989	S	Walker, <i>Autodesk File</i>
1990		
1991	S	Ichbiah & Knepper, <i>Making of Microsoft</i>
	I	Pugh, <i>IBM's 360 & 370 Systems</i>
1992	S	Burton Grad, <i>Evolution of Packaged Software</i>
	I	O'Neill, <i>Evolution of Interactive Computing</i>
1993		
1994	S	Petersen, <i>AlmostPerfect</i>
	S	Levy, <i>Insanely Great</i>
	S	Dunphy, <i>Unix Industry</i>
1995	A	Yates, "Application Software for Insurance"
	A	McKenney, <i>Waves of Change</i>
	S	Cusumano & Selby, <i>Microsoft Secrets</i>
1996	S	Mowery, <i>International Software Industry</i>
	T	Bergin et al., <i>History of Programming Languages II</i>
	I	Norberg & O'Neill, <i>Transforming Computer Technology</i>
1997	T	Glass, <i>Recollections of Software Pioneers</i>
	S	Herman, <i>Phoenix: Fall and Rise of Videogames</i>
1998	S	Torrisi, <i>International Study of Software Industry</i>
	S	Cusumano & Yoffie, <i>Competing: Netscape v. Microsoft</i>
1999	S	Leibowitz & Margolis, <i>Winners, Losers & Microsoft</i>
	S	Mindell, <i>Rise of Relational Databases</i>
	S	Graham, <i>Legal Battles</i>

(continued on next page)

Table 1. (Continued from preceding page)		
Year	Type*	Author and Title
2000	I	Bardini, <i>Bootstrapping</i>
	I	Gillies & Cailliau, <i>How the Web Was Born</i>
	S	Meissner, <i>SAP</i>
2001	I	Akera, <i>Fruits of Collaboration: SHARE</i>
	S	Moody, <i>Rebel Code</i>
	I	Lohr, <i>GoTo</i>
2002	S	<i>Annals</i> , special issue on software industry
	I	Hashagen et al., <i>Software Issues</i>
2003	S	Campbell-Kelly, <i>Sonic the Hedgehog: Software Industry</i>
	S	Norberg, "Software at Eckert-Mauchly"
	A	Bourne & Hahn, <i>Online Information Services</i>
2004	A	Cortada, <i>Digital Hand</i>
	I	Weber, <i>Open Source</i>
	I	Frana, "Before the Web there was Gopher"

* Classification Type:

T = Technology

S = Supply-side industry

A = Applications

I = Institutional, social, political

in languages other than English and still fewer have been translated.⁵ However, a number of edited volumes have international contributors: these include the proceedings of the *History of Programming Languages* conferences edited by Richard Wexelblat (1981) and Tim Bergin et al. (1996), and David Mowery's *International Software Industry* (1996).

To indicate how software writing has changed over time, in Table 1 I have classified each item as "technology," "supply-side industry," "applications," or "institutional, social, political." These are subjective classifications, of course, and not all works were easily pigeon-holed. The table shows how the subject matter has broadened. In the 1960s and 1970s, people wrote about technology—code and software engineering practices. Starting in the 1980s, people began to write about software as an economic activity—primarily the supply-side industry. Some of these studies also began to look tentatively at issues of labor supply and organization. In the 1990s, especially, we began to see books that set software in a much broader institutional, social, or political setting—for example, Emerson Pugh's fine books about IBM, and Arthur Norberg and Judy O'Neill's excellent history of computing at DARPA.

It is only in the past 10 years that scholars have begun to look at applications—a path that was pioneered by JoAnne Yates's 1995 study of insurance-industry software. Thus, over time, software history has evolved from

narrow technical studies, through supply-side and economic studies, to broad studies of applications. Software history is a cumulative exercise. To write about broad software applications, it was necessary to know something about the history of the software industry and the institutions with which it interacted. And to write about the software industry, it was necessary to understand the technologies and practices of software design and engineering. What we write today depends heavily on those who went before.

One could equally well have classified the works in Table 1 by their authors' qualifications. This would show that most of the early works were written by software practitioners. In the middle period, journalists, especially business writers, made many contributions. After that, historians and the academy in general began to add many high-quality studies to the field. All these authors have been essential to the field's development, and it is always appropriate for historians to acknowledge the contributions of individuals for whom history is a passion, not a profession.

A lack of proportionality

As scholarly works on the history of software left the publishing mill, it would have been an interesting experiment to put the accumulation of material on programming languages and systems on the left-hand pan of a set of scales and everything else on the right-hand pan. I think it would not have been until about

Bibliography: Software-Based Information Processing

This bibliography lists selected works on software and software-based information processing.

- Akera, "Voluntarism and the Fruits of Collaboration: The IBM User Group, Share," *Technology and Culture*, vol. 42, 2001, pp. 710-736.
- Annals of the History of Computing*, special 25th anniversary issue on Fortran, vol. 6, no. 1, 1984.
- Annals of the History of Computing*, special 25th anniversary issue on Cobol, vol. 7, no. 4, 1985.
- T. Bardini, *Bootstrapping: Douglas Englebart, Coevolution, and the Origins of Personal Computing*, Stanford Univ. Press, 2000.
- C.J. Bashe et al., *IBM's Early Computers*, MIT Press, 1986.
- Baum, *The System Builders: The Story of SDC*, SDC, 1981.
- T.J. Bergin, *History of Programming Languages, Volume 2*, R.G. Gibson and R.G. Gibson Jr., eds., Addison-Wesley, 1996.
- C.P. Bourne and T. Bellardo Hahn, *A History of Online Information Services, 1963-1976*, MIT Press, 2003.
- F.P. Brooks Jr., *The Mythical Man-Month: Essays on Software Engineering*, Addison-Wesley, 1975.
- Evolution of the U.S. Packaged Software Industry*, Burton Grad Associates Inc., 14 Oct. 1992.
- M. Campbell-Kelly, "Foundations of Computer Programming in Britain 1945-1955," *Annals of the History of Computing*, vol. 4, no. 2, pp. 121-131, 1982.
- M. Campbell-Kelly, *From Airline reservation to Sonic the Hedgehog: A History of the Software Industry*, MIT Press, 2003.
- J. Cortada, *The Digital Hand, Volume 1*, Oxford Univ. Press, 2004.
- M.A. Cusumano and R.W. Selby, *Microsoft Secrets*, The Free Press, 1995.
- M.A. Cusumano and D.B. Yoffie, *Competing on Internet Time: Lessons from Netscape and its Battle with Microsoft*, The Free Press, 1998.
- E. Dunphy, *The Unix Industry*, 2nd ed., Wiley & Sons, 1994.
- F.M. Fisher, J.W. McKie, and R.B. Mancke, *IBM and the U.S. Data Processing Industry: An Economic History*, Praeger, 1983.
- R.L. Forman, *Fulfilling the Computer's Promise: The History of Informatics 1962-1982*, Informatics General Corp., 1985.
- P.L. Frana, 2004, "Before the Web There Was Gopher," *IEEE Annals of the History of Computing*, vol. 26, no. 1, 2004, pp. 20-41.
- P. Frieberger and M. Swaine, *Fire in the Valley: The Making of the Personal Computer*, Osborne/McGraw-Hill, 1984; 2nd ed., McGraw-Hill, 1999.
- J. Gillies and R. Cailliau, *How the Web Was Born: The Story of the World Wide Web*, Oxford Univ. Press, 2000.
- R.L. Glass, ed., *In the Beginning: Personal Recollections of Software Pioneers*, IEEE Computer Soc. Press, 1997.
- L.D. Graham, *Legal Battles that Shaped the Computer Industry*, Quorum Books, 1999.
- Goldberg, ed., *A History of Personal Workstations*, ACM Press, 1988.
- U. Hashagen, R. Keil-Slawik, and A. Norberg, eds., *History of Computing—Software Issues*, Springer-Verlag, 2002.
- L. Herman, *Phoenix: The Fall and Rise of Videogames*, Rolenta Press, 1997.
- Ichbiah and S.L. Knepper, *The Making of Microsoft*, Prima Publishing, 1991.
- IEEE Annals of the History of Computing*, special issue on the early software industry, vol. 24, no. 2, 2002.
- D.E. Knuth, *The Art of Computer Programming, Volume 1*, Addison-Wesley, 1969.

(continued on next page)

1990 that the right-hand pan began to weigh heavier.

There are several reasons why programming languages and systems, especially the former, have received such disproportionate attention. First, in the 1970s the history of computing field was largely populated by practitioners, and their vision was constrained by what they knew—primarily the technical attributes of software. Second, as Jean Sammet has noted, several hundred programming languages existed, of which several dozen were in widespread use.⁶ The study of programming languages was in and of itself an activity of considerable breadth and scope. Third, the

study lent itself to a form of historical textual analysis somewhat like literary criticism, which required little in the way of archival access or oral history programs.

My comments on the study of programming languages are not intended to belittle the endeavors of the pioneers in software history. Indeed, some works were written by outstanding technical experts who also took it on themselves to kick-start the software history field—Jean Sammet, Donald Knuth, and Peter Wegner among them. Nonetheless, these early studies of programming languages were of the low-hanging-fruit variety. Even in the realm of systems software, other aspects were compar-

(continued from preceding page)

- D.E. Knuth, "John von Neumann's First Computer Program," *Computing Surveys*, vol. 2, no. 4, 1970, pp. 247-260.
- D.E. Knuth, "The Early Development of Programming Languages," *A History of Computing in the Twentieth Century*, N. Metropolis, J. Howlett, and G.-C. Rota, eds., Academic Press, 1980, pp. 197-273.
- S. Levy, *Insanely Great: The Life and Times of Macintosh, the Computer that Changed Everything*, Viking, 1994.
- S.J. Liebowitz and S.E. Margolis, *Winners, Losers and Microsoft: Competition and Antitrust in High Technology*, The Independent Institute, 1999.
- S. Lohr, *Go To: The Story of The Programmers Who Created the Software Revolution*, Basic Books, 2001.
- R.H. MacNeal, *The MacNeal-Schwendler Corporation: The First Twenty Years*, MacNeal-Schwendler Corporation, 1987.
- J.L. McKenney, *Waves of Change: Business Evolution through Information Technology*, Harvard Business School Press, 1995.
- G. Meissner, *SAP: Inside the Secret Software Power*, McGraw-Hill, 2000.
- Mindell, "The Rise of Relational Databases," *Funding a Revolution: Government Support for Computing Research*, National Research Council, National Academy Press, 1999, pp. 159-168.
- G. Moody, *Rebel Code: Linux and the Open Source Revolution*, Perseus, 2001.
- D.C. Mowery, ed., *The International Computer Software Industry*, Oxford Univ. Press, 1996.
- A.L. Norberg and J.E. O'Neill, *Transforming Computer Technology: Information Processing for the Pentagon, 1962-1986*, Johns Hopkins Univ. Press, 1996.
- A.L. Norberg, "Software Development at the Eckert-Mauchly Computer Company between 1947 and 1955," *Iterations*, vol. 2, 31 Dec. 2003; see <http://www.cbi.umn.edu/iterations/norberg.pdf>.
- J. O'Neill, "The Evolution of Interactive Computing Through Time Sharing and Networking," doctoral dissertation, Univ. of Minnesota; available from Univ. Microfilms International, Ann Arbor, Mich., 1992.
- W.E. Petersen, *Almost Perfect: How a Bunch of Regular Guys Built WordPerfect Corporation*, Prima, 1994.
- M. Phister Jr., *Data Processing: Technology and Economics*, Digital Press and Santa Monica Publishing Co., 2nd ed., 1979.
- E.W. Pugh, L.R. Johnson, and J.H. Palmer, *IBM's 360 and Early 370 Systems*, MIT Press, 1991.
- S. Rosen, *Programming System and Languages*, McGraw-Hill, 1967.
- R.F. Rosin, "Supervisory and Monitor Systems," *Computing Survey*, vol. 1, no. 1, 1969, pp. 37-54.
- J.E. Sammet, *Programming Languages: History and Fundamentals*, Prentice-Hall, 1969.
- S. Torrisi, *Industrial Organisation and Innovation: An International Study of the Software Industry*, Edward Elgar, 1998.
- J. Walker, ed., *The Autodesk File: Bits of History, Words of Experience*, 3rd ed., New Riders, 1989; the latest edition can be downloaded from <http://www.fourmilab.to/autofile/>.
- S. Weber, *The Success of Open Source*, Harvard Univ. Press, 2004.
- P. Wegner, "Programming Languages—The First 25 Years," *IEEE Trans. on Computers*, vol. C-25, Dec. 1976, pp. 1207-1225.
- L. Welke, "The Origins of Software," *Datamation*, Dec. 1980, pp. 127-130.
- R.L. Wexelblat, ed., *History of Programming Languages*, Academic Press, 1981.
- J. Yates, "Application Software for Insurance in the 1960s and Early 1970s," *Business and Economic History*, vol. 24, no. 1, 1995, pp. 123-134.

actively neglected. Today there is still a dearth of good historical accounts of operating systems and databases.

In the early 1980s, people started to write about the software industry. To date, the emphasis has been on large proprietary software firms. I would single out Claude Baum's excellent *The System Builders* (1981), the history of the Systems Development Corporation (SDC). This book, together with the superb collection of SDC records in the Charles Babbage Institute (CBI) that complement it, is one of the real gems of software history. Another fine work is Richard Forman's history of Informatics, *Fulfilling the Computer's Promise*

(1985). This little-known, privately published work was commissioned by Walter Bauer, the founder of Informatics and a trustee of the CBI.

As a genre, histories of the software industry are very much like business histories of other industries. There is good and bad, but most books are surprisingly useful. The majority were not written by professional historians but by journalists, company founders and executives, and business school academics. For example, Baum, the author of the SDC history, was "not a writer by profession"—he was an SDC employee of more than 20 years' standing; he worked with a team of four other senior

employees who had access to the company archives and between them conducted more than 100 interviews.⁷

Forman, the author of the Informatics history, was a graduate student who made extensive use of the company archives, conducted many interviews, and was assisted by Frank W. Wagner, a cofounder of the company and one of the founding editors of the *Annals of the History of Computing*. Many of these histories are very good indeed, and our field is highly dependent on them—there will never be enough trained historians to do the kind of job one might like.

One of the great strengths of the CBI under Arthur Norberg has been the way it has encouraged the efforts of diverse computer history enthusiasts. For example, the CBI has worked closely with the Software History Center, founded by former software industry executives Burt Grad and Luanne Johnson.

Perhaps the one aspect of the business history of software that is most troubling today is the disproportionate number of studies of the 100 biggest firms, compared with the negligible number of studies of the 100,000 small firms. And among the largest firms there has been a disproportionate interest in Microsoft. If one were to put the books about Microsoft on the left-hand pan of our scales, and all the other books on the right-hand pan, we can guess what the result would be. These cavils apart, however, I would say that the business history of software is in fair shape. But this leaves many unfilled spaces in the software history tapestry.

Voids in software history

In an attempt to stimulate the filling of some of the holes in software history, the CBI and the Heinz Nixdorf MuseumsForum organized a conference, History of Computing: Software Issues, at Paderborn, Germany, in April 2000. The preface to the published proceedings gives an excellent vignette of software history at that time.⁸

As the organizers viewed it, software history had focused on software technologies and early set-piece projects such as the SAGE air defense system and the Bank of America's ERMA system. The organizers wanted to broaden the field but without setting an agenda that was so ambitious it was unworkable. After much deliberation, they agreed on a set of five topics; for each, a leading software historian was invited to prepare a position paper and two or three expert commentators

were invited to make a response. The topics chosen were as follows:

- Software as Science
- Software as Engineering
- Software as Reliable Artifact
- Software as Labor Process
- Software as Economic Activity

As a member of the organizing committee, I was somewhat uncomfortable with some of these choices and the balance between them, although we reached a consensus. I'm quite sure my co-organizers had their misgivings too. For the fact that we did reach a consensus, we have to thank Arthur Norberg, Ulf Hashagen, and Reinhard Keil-Slawik who led the discussions. We ended up with a program of a set of excellent orthogonal discussions of the literature and possible research directions.

As I now look back on the topics, I am conscious that six years is a long time in software history. What, I wonder, could have induced us to give "Software as Dependable Artifact" weight equal to software engineering or the software industry? Perhaps we were still in thrall of the Y2K problem, and this had made us overly sensitive to issues of software quality. More likely we were influenced by the fact that we knew of an outstanding scholar and speaker, Donald MacKenzie of Edinburgh University, who we wanted to give a platform.

Why did we not include applications software as one of our topics? Why did we not include any cultural aspects of software such as videogaming, national styles in software, or the open-source community? The answer is worth stating. We were a group of experts seeking to reach a consensus and move the field forward a few paces, not to push it off the edge of a cliff. This is just the kind of well-judged leadership we have come to expect from the CBI and Arthur Norberg.

The year 2000 was an excellent time for software history. Six months after the Paderborn meeting, the CBI and the Software History Center jointly organized a conference on the early years of the packaged-software industry at the Xerox Palo Alto Research Center. With the inspired title *Unbundling History*, the conference focused on the enterprise software products industry that flourished after IBM's unbundling decision in 1970. At that time this "old" software industry was being eclipsed in the media by Microsoft and the other personal-computer software makers, and the meeting was a timely corrective.

Table 2. Selected works on offices and office-based information processing, with shortened titles. (For full titles, see the "Bibliography: Office-Based Information Processing" sidebar.)

Decade	Type*	Author and Title
1920s	T	Turck, <i>Modern Calculating Machines</i>
	T	Martin, <i>Die Rechenmaschinen</i>
	T	Typewriter Topics, <i>History of the Typewriter</i>
1930s	S	Fortune magazine, "IBM"
	T	Couffignal, <i>Calculating Machines</i>
1940s	S	Fortune magazine, "IBM"
1950s	I	Mills, <i>White Collar</i>
	S	Englebourg, <i>IBM</i>
	S	Bliven, <i>Wonderful Writing Machine</i>
1960s	I	Lockwood, <i>Black Coated Worker</i>
	I	Machlup, <i>Production and Distribution of Knowledge</i>
	T	Richards, <i>History of Typewriter</i>
	A	Truesdell, <i>Punch Card Tabulation</i>
	S	Engler, <i>Typewriter Industry</i>
1970s	I	Holcombe, <i>Victorian Ladies at Work</i>
	T	Beeching, <i>Century of the Typewriter</i>
	I	Anderson, <i>Victorian Clerks</i>
	I	Eyler, <i>William Farr</i>
1980s	I	Davies, <i>Woman's Place is at the Typewriter</i>
	T	Yates, <i>From Pigeonhole to Vertical Filing</i>
	A	Beniger, <i>Control Revolution</i>
	S	Anderson, <i>American Census</i>
	I	Campbell-Kelly, <i>ICL</i>
1990s	I	Yates, <i>Control through Communication</i>
	T	Campbell-Kelly, "Punched Card Machinery"
	A	Norberg, "Punched Card Machinery in Business"
	S	Cortada, <i>Before the Computer</i>
	H	Campbell-Kelly, "Victorian Data Processing"
2000s	H	de Wit, <i>Shaping of Automation</i>
	H	Headrick, <i>When Information Came of Age</i>
	H	Agar, <i>Government Machine</i>
	H	Yates, <i>Structuring the Information Age</i>

* Classification Type:

T = Technology

S = Supply-side industry

A = Applications

I = Institutional, social, political

H = Holistic

Speakers came from academia and industry in equal numbers, building a lasting bridge between the two communities.

For reasons that will become apparent in my closing section, I think it is too early to set a research agenda for software history. In any case some of the obvious holes are already being plugged. For example, application software is getting much-needed attention in Jim Cortada's path-breaking trilogy *The Digital Hand*. There is a glimmering of interest in video games and software cultures. Open

source has received attention from both economic and cultural historians.

Software history records

In the early 1980s, I was one of many people Arthur Norberg consulted about the kinds of records that historians would find useful and that the CBI might collect. One issue at the time was the "manuals problem." I think we all agreed that indiscriminately collecting computer manuals was not the answer to software history or any other kind

Bibliography: Office-Based Information Processing

This bibliography lists selected works on offices and office-based information processing.

- J. Agar, *The Government Machine: A Revolutionary History of the Computer*, MIT Press, 2003.
- M.J. Anderson, *The American Census: A Social History*, Yale Univ. Press, 1988.
- G. Anderson, *Victorian Clerks*, Manchester Univ. Press, 1976.
- "International Business Machines," *Fortune*, Jan. 1932, pp. 34-50.
- "International Business Machines," *Fortune*, Jan. 1940, pp. 36-43, 124, 126, 128, 132.
- W.A. Beeching, *A Century of the Typewriter*, Heinemann, 1974.
- J.R. Beniger, *The Control Revolution: Technological and Economic Origins of the Information Society*, Harvard Univ. Press, 1986.
- B. Bliven Jr., *The Wonderful Writing Machine*, Random House, 1954.
- M. Campbell-Kelly, *ICL: A Business and Technical History*, Oxford Univ. Press, 1989.
- M. Campbell-Kelly, "Punched Card Machinery," *Computing Before Computers*, W. Aspray, ed., Iowa State Univ. Press, 1990.
- M. Campbell-Kelly, "The Railway Clearing House and Victorian Data Processing," *Information Acumen: The Understanding and Use of Knowledge in Modern Business*, L. Bud-Frierman, ed., Routledge, 1994, pp. 51-74.
- J.W. Cortada, *Before the Computer: IBM, NCR, Burroughs, and Remington Rand and the Industry They Created, 1865-1956*, Princeton Univ. Press, 1993.
- L. Couffignal, "Calculating Machines: Their Principles and Evolution," 1933, extracts reprinted, *The Origins of Digital Computers: Selected Papers*, 2nd ed., B. Randell, ed., Springer-Verlag, 1982, pp. 145-154.
- M.W. Davies, *Woman's Place is at the Typewriter: Office Work and Office Workers, 1870-1930*, Temple Univ. Press, 1982.
- D. de Wit, *The Shaping of Automation: A Historical Analysis of the Interaction between Technology and Organization, 1950-1985*, Rotterdam Univ., 1994.
- S. Engelbourg, *International Business Machines: A Business History*, Arno Press, 1976. Originally published as a PhD thesis, Columbia Univ., 1954.
- G.N. Engler, *The Typewriter Industry: The Impact of a Significant Technological Revolution*, UCLA, 1969. Available from Univ. Microfilms International, Ann Arbor, Mich.
- J.M. Eyler, *Victorian Social Medicine: The Ideas and Methods of William Farr*, Johns Hopkins Univ. Press, 1979.
- D.R. Headrick, *When Information Came of Age: Technologies of Knowledge in the Age of Reason and Revolution, 1700-1850*, Oxford Univ. Press, 2000.
- L. Holcombe, *Victorian Ladies at Work: Middle Class Working Women in England and Wales, 1850-1914*, Newton Abbot: David and Charles, 1973.
- D. Lockwood, *The Black Coated Worker*, Allen and Unwin, 1958.
- F. Machlup, *The Production and Distribution of Knowledge in the United States*, Princeton Univ. Press, 1962.
- E. Martin, *The Calculating Machines (Die Rechenmaschinen): Their History and Development*, 1925; P. Aldrich Kidwell and M.R. Williams, trans. and ed., Charles Babbage Reprint Series for the History of Computing, vol. 16, MIT Press, 1992.
- C.W. Mills, *White Collar: The American Middle Classes*, Oxford Univ. Press, 1951.
- A.L. Norberg, "High-Technology Calculation in the Early 20th Century: Punched Card Machinery in Business and Government," *Technology and Culture*, vol. 31, no. 4, 1990, pp. 753-779.
- G.T. Richards, *The History and Development of Typewriters*, London: HMSO, 1964.
- L.E. Truesdell, *The Development of Punch Card Tabulation in the Bureau of the Census, 1890-1940*, US Dept. of Commerce, 1965.
- J.A.V. Turck, *Origin of Modern Calculating Machines*, Western Soc. of Engineers, 1921.
- Typewriter Topics, *The Typewriter: An Illustrated History*, 1924, reprinted, Dover, 2000.
- J. Yates, "From Press Book and Pigeonhole to Vertical Filing: Revolution in Storage and Access Systems for Correspondence," *J. Business Communication*, vol. 19, no. 3, 1982, pp. 5-26.
- J. Yates, *Control Through Communication: The Rise of System in American Management*, Johns Hopkins Univ. Press, 1989.
- J. Yates, *Structuring the Information Age: Life Insurance and Technology in the Twentieth Century*, Johns Hopkins Univ. Press, 2005.

of history. There was a story circulating at the time that since the launch of System/360, IBM had become, in terms of the number of titles issued, the largest publisher in the US apart from the federal government. This may have been an urban myth, but it certainly made the point. Collecting manuals was about what you

could collect, not what you *should* collect. Lay people often misunderstand records collection: quality and selection is paramount; we can't keep every thing, and choices have to be made.

When I wrote my history of the software industry, I became aware that one of the most

potentially useful sources was the tens of thousands of industry reports produced since about 1970 by firms such as IDC, ICP, INPUT, Forrester Research, Datapro, and several others. My research fellow and I wrote to all the firms that still existed, asking what reports they had kept and if we could see them. Most of those that troubled to reply gave essentially the same answer: they did not keep reports going back more than 10 years, and they could not give us access to those they had retained. For us, it was a case of disappointment quickly followed by relief: how could we process thousands of reports anyway? Moreover, those familiar with the genre I think will agree that industry reports are not the most compelling reading.

Almost as my manuscript was being shipped to MIT Press, two analysts in fact decided to make their archives available to historians. This was a direct outcome of the Unbundling History conference, and it stands as a testimony to the way in which the CBI and the Software History Center have built bridges to the practitioner community. First, Larry Welke donated the ICP archive to the CBI. I was able to draw on this before my book shipped to the publisher. Actually, I may be overstating the case—there were several bankers' boxes of material and I had just two days to explore them. Second, Peter Cunningham of INPUT stated he would make his firm's archive available to historians. This was a truly generous offer, but my book was in press before I could take advantage of it. But again, I'm not sure how a lone historian could have used such an enormous volume of material.

It may be that the time has come to re-evaluate what records computer archives keep. This is not an issue for Arthur Norberg, but rather his successor Tom Misa. Most of the archives I use are still in the pencil-and-yellow-pad era. Recently, however, I was engaged as a historical expert for a software lawsuit. In many ways lawyers do the same job as historians, though to a shorter timescale, and perhaps with less objectivity. Lawyers have lots of money, and this has enabled them on occasion to take a lead in the application of information technology. For example, Lexis and WestLaw were among the first pioneering suppliers of online information.⁹ What best-practice lawyers are currently doing is ahead of where historians are today. In the case I was involved in, there were several million pages of testimony and subpoenaed documents. These had all been scanned, OCR processed, and resided in a database in Los Angeles. Over the

Internet, from Warwick University, I was able to interrogate this mass of material with a powerful query language. For someone more used to a pencil, notebook, and a photocopy request form, it was quite an eye-opener. Particularly as born-digital records emerge, all modern-records archives will need to implement electronic searching and remote access. It would be very appropriate if the CBI were to be in the vanguard of this transition.

What software history may become

All computer historians would agree that the state of software history is not what we would wish. It is easy enough to articulate particular faults (as I have done here); it is much more difficult to suggest remedies. Below I quote from the social historian Harold Perkins who put his thumb on the problem rather well. I do this with some hesitation because the quotation comes from a review of my history of the software industry *From Airline Reservations to Sonic the Hedgehog* (2003), which appeared in the *Times Literary Supplement*:

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. 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. 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 take for granted how they changed the world.¹⁰

They say there is no such thing as bad publicity. I hope so. Actually, I think Perkins' criticisms are overstated, and he was perhaps not the right reviewer for my particular book. The book I wrote was a fairly standard, competent if undistinguished, business history, and most business historians would recognize it as such. Nonetheless, Perkins is surely right in characterizing the kind of software history we would all like to see. Let's read again

his key sentence: “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.”

The fact is, we are years away from writing a book like this. I would hazard 10 or 15 years. What is it that stops us? To answer this question, allow me to digress to consider the history of the office.

In Table 2 and the “Bibliography: Office-Based Information Processing” sidebar, I have listed some of the major works on the history of offices and office-based information processing. What we see is an evolution in publications not unlike that which is currently unfolding in software history. First, in the 1920 and 1930s, there was a flurry of books about office machine technologies—the typewriter, adding machines, and punched card machines. After a rather lean period during the Depression years and World War II, in the late 1940s and 1950s there appeared books and articles about the office machine industry—IBM, NCR, and others—and this business-history tradition continues in the writings of Cortada and others. The 1950s and 1960s saw a major strand of literature about office work and office workers; in the 1970s and 1980s this aspect of office history engaged with gender studies. In the 1980s there was a growing interest in office applications, such as the census and banking. To write a holistic history of the office, it was necessary to draw on all these different genres—technology, industry, applications, labor history, and the social and institutional contexts.

In the 1990s, especially the past few years, this holistic literature has started to appear. I would single out JoAnne Yates’s *Structuring the Information Age* (2005; a longitudinal history of information processing in the insurance industry) and Jon Agar’s *Government Machine* (2003; a study of British bureaucracy). Both books cover an enormous time span—Yates starting in the 1890s and Agar starting even earlier, until recent times. Both books integrate the histories of office machinery from manual methods through punched cards to computers, the supply-side industries, labor and employment issues, as well as complex social, governmental, and regulatory issues. If we replace the words “computer” and “software” by the concepts of “office” and “clerks,” the Perkins sentence might read: “The office and its clerical 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 sentence captures perfectly the scale and scope of Yates’s and Agar’s books. But it is only in the past decade or so, following 70 or 80 years of cumulative historical activity, that it has been possible to write such histories.

My hypothesis is that the history of software is following a similar trajectory. Software history began with narrow but essential technical studies in the 1960s and 1970s. We then looked at the supply-side industry in the 1980s and 1990s. We have only just begun to study applications. The study of institutional, social, and political issues has barely begun. Software history is heading in the right direction, but we need a solid base of secondary literature before it is possible to write the major holistic works that will do our profession justice.

Finally, when one comments upon the shortcomings of software histories, it is important to keep in mind that the history of software is dynamic and still unfolding. I do not mean, of course, that the history of the office has ended. But it has evolved through a distinct set of changes of location and process from which we can construct solid trajectories that give us a sense of where we have come from and perhaps where we are going. Thus, office locations have evolved from small to large, and from decentralized to centralized. Office processes have evolved from manual techniques, through business machinery, such as typewriters and punched-card machines, to highly automated computer systems. These trajectories are clearly reflected in the Yates’s and Agar’s narratives.

Software has not yet passed through comparably long trajectories. For example, in terms of process, we have evolved from a craft discipline to software engineering, but not much beyond. We have evolved from bespoke software programs to software products, but, again, not much beyond. These trajectories lack enough data points to clearly indicate where we are headed. Indeed, there is a sense that software is still turbulent and evolving. For example, as I write, many software companies are in a state of transition from products to services: such software-as-a-service companies deploy software on their own servers on behalf of users, instead of users licensing software products as such.

If a software-as-a-service company is an enterprise that writes, hosts, and deploys

software over the Internet, then what is an online bank? It too is software driven—users store financial data and make transactions in a way not dissimilar, for example, to Web-based software. To take a less provocative example, consider eBay, which is very much a software creation. In a technical sense, eBay is a software platform¹¹—it publishes its Application Program Interfaces, and hundreds of firms exist by offering software and Web-service complements.¹²

It is possible that we are observing a convergence between ordinary information businesses and software enterprises. As software firms migrate to computer services, they will become increasingly indistinguishable from other information businesses that conduct their operations over the Internet. In 20 or 30 years' time, the outcome of these changes will be known, and software may have entered a settled state, rather like today's offices. It will then at last be possible to write satisfactorily about "a revolution in human development as significant as the steam engine, the automobile or the aeroplane."

References and notes

1. M. Campbell-Kelly, "Programming the EDSAC: Early Programming Activity at the Univ. of Cambridge," *Annals of the History of Computing*, vol. 2, no. 1, 1980, pp. 7-36; M. Campbell-Kelly, "Programming the Mark I: Early Programming Activity at the Univ. of Manchester," *Annals of the History of Computing*, vol. 2, no. 2, 1980, pp. 133-162; M. Campbell-Kelly, "Programming the Pilot ACE: Early Programming Activity at the National Physical Laboratory," *Annals of the History of Computing*, vol. 3, no. 2, 1981, pp. 133-168; M. Campbell-Kelly, "The Development of Computer Programming in Britain 1945-1955," *Annals of the History of Computing*, vol. 4, no. 2, 1982, pp. 121-131.
2. J.M. Wheeler, "Applications of the EDSAC," *IEEE Annals of the History of Computing*, vol. 14, no. 4, 1992, pp. 27-33.
3. A. Hodges, *Alan Turing: The Enigma*, Burnett Books, 1983.
4. D.M. Yates, *Turing's Legacy: A History of Computing at the National Physical Laboratory, 1945-1995*, Science Museum, 1997.
5. An exception is G. Meissner's *SAP: Inside the Secret Software Power* (McGraw-Hill, 2000), a translation of *SAP: die heimliche Software-Macht* (Wilhelm Heyne Verlag, 1997).
6. J.E. Sammet, *Programming Languages: History and Fundamentals*, Prentice Hall, 1969.
7. C. Baum, *The System Builders: The Story of SDC*, SDC, pp. vii-ix.
8. U. Hashagen, R. Keil-Slawik, and A. Norberg, eds. *History of Computing—Software Issues*, Springer-Verlag, 2002.
9. C.P. Bourne and T. Bellardo Hahn, *A History of Online Information Services, 1963-1976*, MIT Press, 2003, pp. 300-304, 330-331.
10. H. Perkins, "Revenge of the Anoraks," *Times Literary Supplement*, London, 10 Oct. 2003, p. 28.
11. D.S. Evans, A. Hagiu, and R. Schmalensee, *Invisible Engines: How Software Platforms Drive Innovation and Transform Industries*, MIT Press, 2006, pp. 349-355.
12. See for example, J.P. Mueller, *Mining eBay Web services: Building Applications with the eBay API*, Sybex, 2004.



Martin Campbell-Kelly is a professor in the Department of Computer Science at the University of Warwick, where he specializes in the history of computing.

Readers may contact Campbell-Kelly at m.campbell-kelly@warwick.ac.uk.

For further information on this or any other computing topic, please visit our Digital Library at <http://www.computer.org/csdl>.