THE HISTORY OF THE HISTORY OF SOFTWARE

by

Martin Campbell-Kelly
Warwick University

Introduction

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 which 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 Professor 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 *The Origins of Digital Computers* (1972).

One notes 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 the course of 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*,[1] and only a change of job to Warwick University and an itch to move on to fresh fields prevented my producing a monograph. 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). I wrote to several individuals, all but one of whom declined to be involved. They declined politely—it was just that none of the people I approached felt comfortable with computers and some did not work on post-Second World War science anyway. Very fortunately one person did agree to help me—Margaret Gowing, the recently appointed professor of the history of science at Oxford University. Margaret was an economic historian by training, and she had been a driving force behind the civil history of the Second World War, and subsequently wrote a number of classic works on the development of atomic energy in Britain. Margaret invited me to Oxford where I told her about what I was planning. I was more than a little in awe of Margaret but she instantly put me at my ease, and offered to read what I wrote when I wrote it, and that I should feel free to call on her at any time I needed help. Anxious not to over-play my hand, I did so on just two or three subsequent occasions during the next few years. The fact that she endorsed the topic of my research was the encouraging message I took away from that meeting.

Several months later I sent Margaret 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

[1] M. Campbell-Kelly, "Programming the EDSAC: Early Programming Activity at the University of Cambridge," *Annals of the History of Computing*, vol. 2 (1980), pp. 7-36. M. Campbell-Kelly, "Programming the Mark I: Early Programming Activity at the University of Manchester," *Annals of the History of Computing*, vol. 3 (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

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, many years later, I began to see the point Professor 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, and which was an essential stepping stone on the path to the discovery of DNA by Watson and Crick. 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. Similarly for the other British machines. 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 United States. 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] 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,

---

(1981), pp. 133-168. M. Campbell-Kelly, "Foundations of Computer Programming in Britain 1945-1955," *Annals of the History of Computing*, vol. 4 (1982), pp. l2l -131.

[2] Joyce M. Wheeler, "Applications of the EDSAC," *IEEE Annals of the History of Computing*, vol. 14, no. 4 (1992), pp. 27-33..

[3] Andrew Hodges, *Alan Turing: The Enigma* (London: Burnett Books, 1983).

which was still reeling from the de Havilland Comet air disaster of 1954. Yet in 1976, I just could not see that there was anything here that 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 McKenny'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.

The Evolution of Software History

To portray the evolution of writings about software, in Table 1 and the Bibliography 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 began to appear. This is not a scientific survey—I have simply gone through the bibliographies of my own writings on software and I have culled the items I personally found most useful,

[4] David M. Yates, *Turing's Legacy: A History of Computing at the National Physical Laboratory, 1945-1995* (London: Science Museum, 1997).

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.

To indicate how software writing has changed over time, in Table 1 I have classified each item as "technology", "software 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 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 last 10 years that scholars have begun to look at applications—a path that was pioneered by JoAnne Yates' 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. In order 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 is heavily dependent on those who went before.

One could equally well have classified the works in Table 1 by the qualifications of their authors. 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 of these authors have been essential to the development of

the field, 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 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, there were several hundred programming languages of which several dozen were in widespread use.[5] 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, that 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 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 comparatively 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,* the history of the Systems Development Corporation

[5] Sammet, Jean E., *Programming Languages: History and Fundamentals* (Englewood Cliffs, N.J: Prentice Hall, 1969).

(SDC). This book, together with the superb collection of SDC records in the 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*. This little known 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. 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 one-hundred biggest firms, compared with the negligible number of studies of the one-hundred thousand 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 set of 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.[6]

[6] Ulf Hashagen, Reinhard Keil-Slawik, and Arthur Norberg, eds., *History of Computing—Software Issues*, Berlin: Springer-Verlag, 2002.

As the organizers saw 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 that it was unworkable. After much deliberation they agreed on a set of five topics, for each of which 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:

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

I was a member of the organizing committee, and although I was somewhat uncomfortable with some of these choices and the balance between them, 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 definite 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 Reliable Artifact" equal weight 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, to whom 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 video gaming, 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 perfect location of the Xerox Palo Alto Research Center. With the inspired title of *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 PC software makers, and the meeting was a timely corrective. 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-braking 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 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 United States apart from the 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 everything 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 records they kept and if we could see them. Most of those that troubled to reply gave essentially the same answer: They did not keep records going back more than ten 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, if you are familiar with the genre I think you 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 archive of ICP to the CBI. I was able to draw on this before my book was 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 firms 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 for 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 an 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 Nexis and WestLaw were among the first pioneering suppliers of online information.[7] What best-practice lawyers are currently doing is someway ahead of where historians are today. In the case I was involved in, there were several million pages of testimony and subpoenaed documents.

[7] Bourne, Charles P. and Trudi Bellardo Hahn, *A History of Online Information Services, 1963-1976* (Cambridge, MA: MIT Press, 2003).

These had all been scanned, OCR processed, and resided in a database in Los Angeles. Over the Internet, from Warwick, 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.

Emulation and Simulation

One of the intriguing opportunities of computer history is the possibility of executing old software by emulating the computer on which it originally ran. This seems to be largely the preserve of retired software engineers, computer preservationists, and video game collectors.

How people choose to spend their recreational time is rightly no business of mine. I think, however, that it is legitimate to analyze whether emulation serves any useful historical purpose. In my opinion, software revivalists belong to a long tradition of hands-on restorers of old technology. Their activity is comparable to restoring vintage automobiles or railway engines. Perhaps the closest analogy is with the restoration of musical automata and phonographs—analogous in the sense that restoring the hardware enables the original "software" to be re-experienced. These activities can be very useful in museums for creating interesting exhibits and in motivating volunteers. The question I ask here, however, is what they bring to software history.

Writing history is a cumulative process. In my own case, I have relied very heavily on secondary sources. Of the sources listed in Table 1 and the Bibliography, every one has played some part in the books and articles I have written. And I like to think that my writings will eventually repay some of that debt, as they in their turn become secondary sources for historians further up the evolutionary chain.

I have never had occasion to find a use for restored software, as such, in anything I have written. Certainly I have found a use for articles based on emulation studies useful, but these are very few in number compared with the total number of emulation projects. Those who emulate all too rarely publish their achievements in a way that is useful to historians. To repeat, it is not for me to lecture software enthusiasts how they spend their

leisure time; in any case, having met some, I have noticed that listening is not their most conspicuous attribute. However, it is perhaps as well they understand that what they do has no real impact on the greater historical project, unless they emulate historically significant artifacts and communicate their results.

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 in this paper); it is much more difficult to suggest remedies. Below I quote from the social historian Harold Perkins who put his finger 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*, 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.[8]

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

[8] Harold Perkins, "Revenge of the Anoraks," *Times Literary Supplement* (Oct. 10, 2003), p. 28.

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 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 last few years, this holistic literature has started to appear. I would single out JoAnne Yates' *Structuring the Information Age* (a longitudinal history of information processing in the insurance industry) and Jon Agar's *Government Machine* (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”, 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.

I think this sentence captures perfectly the scale and scope of Yates’ and Agar’s books. But it is only in the last 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. I conclude that software history is heading in the right direction, but we need to accumulate a solid base of secondary literature before it is possible to write the major holistic works that will do our profession justice.

Bibliography

I Works on Software and Software-based Information Processing

Akera, Atsushi. 2001. Voluntarism and the Fruits of Collaboration: The IBM User Group, Share. *Technology and Culture,* vol. 42, pp. 710-736.

Annals of the History of Computing, vol. 6, no. 1. January 1984. A special 25th anniversary issue on FORTRAN.

Annals of the History of Computing, vol. 7, no. 4. October 1985. A special 25th anniversary issue on COBOL.

Bardini, Thierry, 2000. *Bootstrapping: Douglas Englebart, Coevolution, and the Origins of Personal Computing*. Stanford: Stanford University Press,.

Bashe, Charles J., Lyle R. Johnson, John H. Palmer, and Emerson W. Pugh. 1986. *IBM's Early Computers*. Cambridge, MA: MIT Press.

Baum, Claude. 1981. *The System Builders: The Story of SDC.* Santa Monica: SDC.

Bergin, Thomas J. 1996. Richard G. Gibson and Richard G., Jr Gibson, eds., *History of Programming Languages, Volume 2.* Reading, Mass.: Addison-Wesley.

Bourne, Charles P. and Trudi Bellardo Hahn, 2003. *A History of Online Information Services, 1963-1976*. Cambridge, MA: MIT Press.

Brooks, Frederick P. Jr., 1975 *The Mythical Man-Month: Essays on Software Engineering*. Reading MA: Addison-Wesley.

Burton Grad Associates Inc., 1992. *Evolution of the U. S. Packaged Software Industry.* Tarrytown, NY: Burton Grad Associates Inc., October 14.

Campbell-Kelly, Martin, "Foundations of Computer Programming in Britain 1945-1955", *Annals of the History of Computing* **4**,. 1982): 121 -131.

Campbell-Kelly, Martin. 2003. *From Airline reservation to Sonic the Hedgehog: A History of the Software Industry*. Cambridge, MA: MIT Press.

Cortada, James. 2004. *The Digital Hand, Volume 1*. Oxford: Oxford university Press.

Cusumano, Michael A. and Richard W. Selby. 1995. *Microsoft Secrets*. New York.

Cusumano, Michael A. and David B. Yoffie. 1998. *Competing on Internet Time: Lessons from Netscape and its Battle with Microsoft*. New York: The Free Press.

Dunphy, Ed. 1994. *The Unix Industry*. New York: Wiley, 2nd ed.

Fisher, Franklin M., James W. McKie and Richard B. Mancke. 1983. *IBM and the U.S. Data Processing Industry: An Economic History*. New York: Praeger.

Forman, Richard L. 1985. *Fulfilling the Computer's Promise: The History of Informatics 1962-1982*. Woodland Hills CA: Informatics General Corp..

Frana, Philip L. 2004. "Before the Web There Was Gopher," *IEEE Annals of the History of Computing,* vol. 26, no. 1., pp. 20-41.

Frieberger, Paul and Michael Swaine. 1984. *Fire in the Valley: The Making of the Personal Computer*. Berkeley, CA: Osborne/McGraw-Hill; 2nd ed., McGraw-Hill, 1999.

Gillies, James and Robert Cailliau. 2000. *How the Web Was Born: The Story of the World Wide Web.* Oxford: Oxford University Press.

Glass, Robert L., ed. 1997. *In the Beginning: Personal Recollections of Software Pioneers*. Los Alamitos CA: IEEE Computer Society Press.

Graham, Lawrence D. 1999 *Legal Battles that Shaped the Computer Industry*. Westport, CT: Quorum Books.

Goldberg, Adele ed. 1988. *A History of Personal Workstations*. New York: ACM Press.

Hashagen, Ulf, Reinhard Keil-Slawik, and Arthur Norberg, eds. 2002. *History of Computing—Software Issues*. Berlin: Springer-Verlag.

Herman, Leonard. 1994. *Phoenix: The Fall and Rise of Videogames*. Union, NJ: Rolenta Press; 2nd ed. 1997.

Ichbiah, Daniel and Susan L. Knepper. 1991. *The Making of Microsoft*. Rocklin, CA: Prima Publishing.

IEEE Annals of the History of Computing, vol. 24, no. 2, 2002. A special issue on the early software industry.

Knuth, Donald E. 1969. *The Art of Computer Programming, Volume 1*. Reading, MA: Addison-Wesley.

Knuth, Donald E. 1970. John von Neumann's First Computer Program. *Computing Surveys* **2**, no. 4, pp. 247-260.

Knuth, Donald E. 1980 The Early Development of Programming Languages. In Metropolis, N., J. Howlett, and G.-C. Rota, eds., *A History of Computing in the Twentieth Century*. New York: Academic Press, pp. 197-273.

Levy, Steven. 1994. *Insanely Great: The Life and Times of Macintosh, the Computer that Changed Everything.* New York.

Liebowitz, Stanley J. and Stephen E. Margolis 1999. *Winners, Losers and Microsoft: Competition and Antitrust in High Technology.* Oakland, CA: The Independent Institute.

Lohr, Steve. 2001. *Go To: The Story of The Programmers Who Created the Software Revolution*. New York: Basic Books.

MacNeal, Richard H. 1987. *The MacNeal-Schwendler Corporation: The First Twenty Years*. Los Angeles, CA: MacNeal-Schwendler Corporation.

McKenney, James L. 1995. *Waves of Change: Business Evolution through Information Technology*. Boston MA: Harvard Business School Press.

Meissner, Gerd. 2000. *SAP: Inside the Secret Software Power*. New York: McGraw-Hill.

Mindell, David. 1999. The Rise of Relational Databases. In National Research Council, *Funding a Revolution: Government Support for Computing Research.* Washington DC: National Academy Press, pp. 159-168.

Moody, Glyn. 2001. *Rebel Code: Linux and the Open Source Revolution.* New York: Perseus.

Mowery, David C., ed. 1996. *The International Computer Software Industry*. New York: Oxford University Press.

Norberg Arthur L., and Judy E. O'Neill. 1996. *Transforming Computer Technology: Information Processing for the Pentagon, 1962-1986*. Baltimore: Johns Hopkins University Press.

Norberg Arthur L., Dec. 31, 2003. Software Development at the Eckert-Mauchly Computer Company Between 1947 and 1955. *Iterations*, vol. 2 at http://www.cbi.umn.edu/iterations/norberg.pdf

O'Neill, Judy, *The Evolution of Interactive Computing Through Time Sharing and Networking.* 1992. Ph.D. diss., University of Minnesota; available from University Microfilms International, Ann Arbor, Mich.

Petersen, W. E. 1994. *AlmostPerfect: How a Bunch of Regular Guys Built WordPerfect Corporation*. Rocklin, CA: Prima.

Phister, Montgomery Jr., 1979. *Data Processing: Technology and Economics*. Santa Monica, CA: Digital Press and Santa Monica Publishing Company, 2nd ed.

Pugh, Emerson W., Lyle R. Johnson and John H. Palmer. 1991. *IBM's 360 and Early 370* Systems. Cambridge, MA: MIT Press.

Rosen, Saul. 1967. *Programming System and Languages*. New York: McGraw-Hill.

Rosin, Robert F. 1969. Supervisory and Monitor Systems. *Computing Survey*, vol. 1, no. 1, pp. 37-54.

Sammet, Jean E. 1969. *Programming Languages: History and Fundamentals*. Englewood Cliffs, N.J.: Prentice-Hall.

Torrisi, Salvatore. 1998. *Industrial Organisation and Innovation: An International Study of the Software Industry*. Cheltenham: Edward Elgar.

Walker, John, ed. 1989. *The Autodesk File: Bits of History, Words of Experience.* Thousand Oaks, CA: New Riders Publishing, 3rd ed.; the latest edition can be downloaded from http://www.fourmilab.to/autofile/.

Weber, Steve. 2004. *The Success of Open Source*. Cambridge, MA: Harvard University Press.

Wegner, Peter. Dec. 1976. Programming Languages—The First 25 Years. *IEEE Trans. on Computers*, vol. C-25, pp. 1207:1225.

Welke, Larry. December 1980. "The Origins of Software," *Datamation*. pp. 127-130.

Wexelblat, Richard L., ed. 1981. *History of Programming Languages*. New York: Academic Press.

Yates, JoAnne. 1995. Application Software for Insurance in the 1960s and Early 1970s. *Business and Economic History*, vol. 24, no. 1, pp. 123-134.

II Works on Offices and Office-based Information Processing

Agar, Jon. 2003. *The Government Machine: A Revolutionary History of the Computer* Cambridge, MA: MIT Press.

Anderson, Margo J. 1988. *The American Census: A Social History.* New Haven, Conn.: Yale University Press.

Anderson, Gregory. 1976. *Victorian Clerks*. Manchester: Manchester University Press.

Anon. January 1932. "International Business Machines." *Fortune*, pp. 34-50.

Anon. January 1940. "International Business Machines." *Fortune*, pp. 36-43, 124, 26, 128, 132.

Beeching, Wilfred A. 1974. *A Century of the Typewriter*. London: Heinemann.

Beniger, James R. 1986. *The Control Revolution: Technological and Economic Origins of the Information Society.* Cambridge, Mass.: Harvard University Press.

Bliven Jr., Bruce. 1954. *The Wonderful Writing Machine.* New York: Random House.

Campbell-Kelly, Martin. 1989. *ICL: A Business and Technical History.* Oxford: Oxford University Press.

Campbell-Kelly, Martin. 1990. Punched Card Machinery. In, William Aspray, ed., *Computing Before Computers.* Ames: Iowa State University Press.

Campbell-Kelly, Martin. 1994. The Railway Clearing House and Victorian Data Processing. In Bud-Frierman Bud-Frierman, Lias, ed. *Information Acumen: The*

Understanding and Use of Knowledge in Modern Business. London and New York: Routledge, pp. 51-74.

Cortada, James W. 1993. *Before the Computer: IBM, NCR, Burroughs, and Remington Rand and the Industry They Created*, 1865-1956. Princeton, NJ: Princeton University Press.

Couffignal, L. 1933. Calculating Machines: Their Principles and Evolution. In Brian Randell. ed., *The Origins of Digital Computers: Selected Papers*, 2nd ed., Berlin: Springer-Verlag, 1982, pp. 145-154.

Davies, Margery W. 1982. *Woman's Place is at the Typewriter: Office Work and Office Workers, 1870-1930.* Philadelphia: Temple University Press.

de Wit, Dirk. 1994. *The Shaping of Automation: A Historical Analysis of the Interaction between Technology and Organization, 1950-1985.* Hilversum: Rotterdam University.

Engelbourg, Saul. 1976. *International Business Machines: A Business History*. New York: Arno Press. Orig. pub. as a Ph.D. thesis, Columbia University, 1954.

Engler, George Nichols. 1969. *The Typewriter Industry: The Impact of a Significant Technological Revolution*. Los Angeles: UCLA. Available from University Microfilms International, Ann Arbor, Mich.

Eyler, John M. 1979. *Victorian Social Medicine: The Ideas and Methods of William Farr*. Baltimore: The Johns Hopkins University Press.

Headrick, Daniel R. 2000. *When Information Came of Age: Technologies of Knowledge in the Age of Reason and Revolution, 1700-1850.* Oxford: Oxford University Press.

Holcombe, Lee. 1973. *Victorian Ladies at Work: Middle Class Working Women in England and Wales, 1850 1914*. Newton Abbot: David and Charles.

Lockwood, David. 1958. *The Black coated Worker*. London: Allen and Unwin.

Machlup, Fritz. 1962. *The Production and Distribution of Knowledge in the United States.* Princeton: Princeton University Press.

Martin, Ernst, translated and edited by Peggy Aldrich Kidwell and Michael R. Williams. 1992. *The Calculating Machines. Die Rechenmaschinen): Their History and Development. 1925),* Charles Babbage Reprint Series for the History of Computing Volume 16. Cambridge, MA: MIT Press.

Mills, C. Wright. 1951. *White Collar: The American Middle Classes.* New York: Oxford University Press.

Norberg, Arthur L. 1990. "High-Technology Calculation in the Early 20th Century: Punched Card Machinery in Business and Government." *Technology and Culture* 31, no. 4, pp. 753-79.

Richards, G. Tilghman. 1964. *The History and Development of Typewriters*. London: HMSO.

Truesdell, Leon E. 1965. *The Development of Punch Card Tabulation in the Bureau of the Census, 1890-1940.* Washington, D.C.: U.S. Department of Commerce.

Turck, J. A. V. 1921. *Origin of Modern Calculating Machines.* Chicago: Western Society of Engineers.

Typewriter Topics, *The Typewriter: An Illustrated History*. Mineola, New York: Dover, 2000; reprint of 1923 original.

Yates, JoAnne. 2005. *Structuring the Information Age: Life Insurance and Technology in the Twentieth Century*. Baltimore, MD: Johns Hopkins University Press.

Yates, JoAnne. 1982. From Press Book and Pigeonhole to Vertical Filing: Revolution in Storage and Access Systems for Correspondence. *Journal of Business Communication* 19. Summer), pp. 5-26.

Yates, JoAnne. 1989. *Control Through Communication: The Rise of System in American Management.* Baltimore, MD: Johns Hopkins University Press.

1967	Rosen, Programming Systems & Languages		
1968			
1969	Sammet, Programming Languages	Rosin, Supervisory & Monitor Systems	Knuth, Art of Computer Programming
1970	Knuth, von Neumann's First Program		
1971			
1972			
1973			
1974			
1975	Brooks, Mythical Man Month		
1976	Wexelblat, HOPL	Wegner, Programming Languages	
1977			
1978			
1979	Phister, Data Processing Economics		
1980	Knuth, Early Programming Languages	Welke, Origins of Software	
1981	Baum, System Builders [SDC]		
1982	Campbell-Kelly, Foundations of Programming		
1983	Fisher et. al, IBM & U.S. D.P. Industry		
1984	Frieberger & Swaine, Fire in the Valley	Annals, Special Issue on FORTRAN	
1985	Forman, Computer's Promise [Informatics]	Annals, Special Issue on COBOL	
1986	Bashe, IBM's Early Computers		
1987	MacNeal, MacNeal-Schwendler Corp.		
1988	Goldberg, History of Personal Workstations		
1989	Walker, Autodesk File		
1990			
1991	Ichbiah & Knepper, Making of Microsoft	Pugh, IBM 360 & 370 Systems	
1992	Burton Grad, Evolution of Packaged Software	O'Neil, Evolution of Interactive Computing	
1993			
1994	Herman, Phoenix [Video Games]	Levy, Insanely Great	Dunphy, Unix Industry
1995	Yates, Application Software for Insurance	McKenny, Waves of Change	Cusumano & Selby, Microsoft Secrets
1996	Mowery, International Software Industry	Bergin et al, HOPL II	Norberg & O'Neil, ARPA's IPTO
1997	Glass, Recollections of Software Pioneers		
1998	Torrisi, International Software Industry	Cusumano & Yoffie, Netscape v. Microsoft	
1999	Leibowitz & Margolis, Microsoft Economics	Mindel, Relational Database	Graham, Legal Battles
2000	Bardini, Bootstrapping	Gillies & Cailliau, How Web Was Won	Meissner, SAP
2001	Akera, SHARE	Moody, Rebel Code	Lohr, GoTo
2002	Annals, Special Issue on Software Industry	Hashagen, R-K, & Norberg, Software Issues	
2003	Campbell-Kelly, Software Industry	Norberg, Eckert-Mauchly Software	Bourne & Hahn, Online Services
2004	Cortada, Digital Hand	Weber, Open Source	Frana, Before the Web

KEY:
TECHNOLOGY
SUPPLY-SIDE INDUSTRY
APPLICATONS
INSTITUTIONAL, SOCIAL, POLITICAL

Table 1. Literature on software and software-based information processing

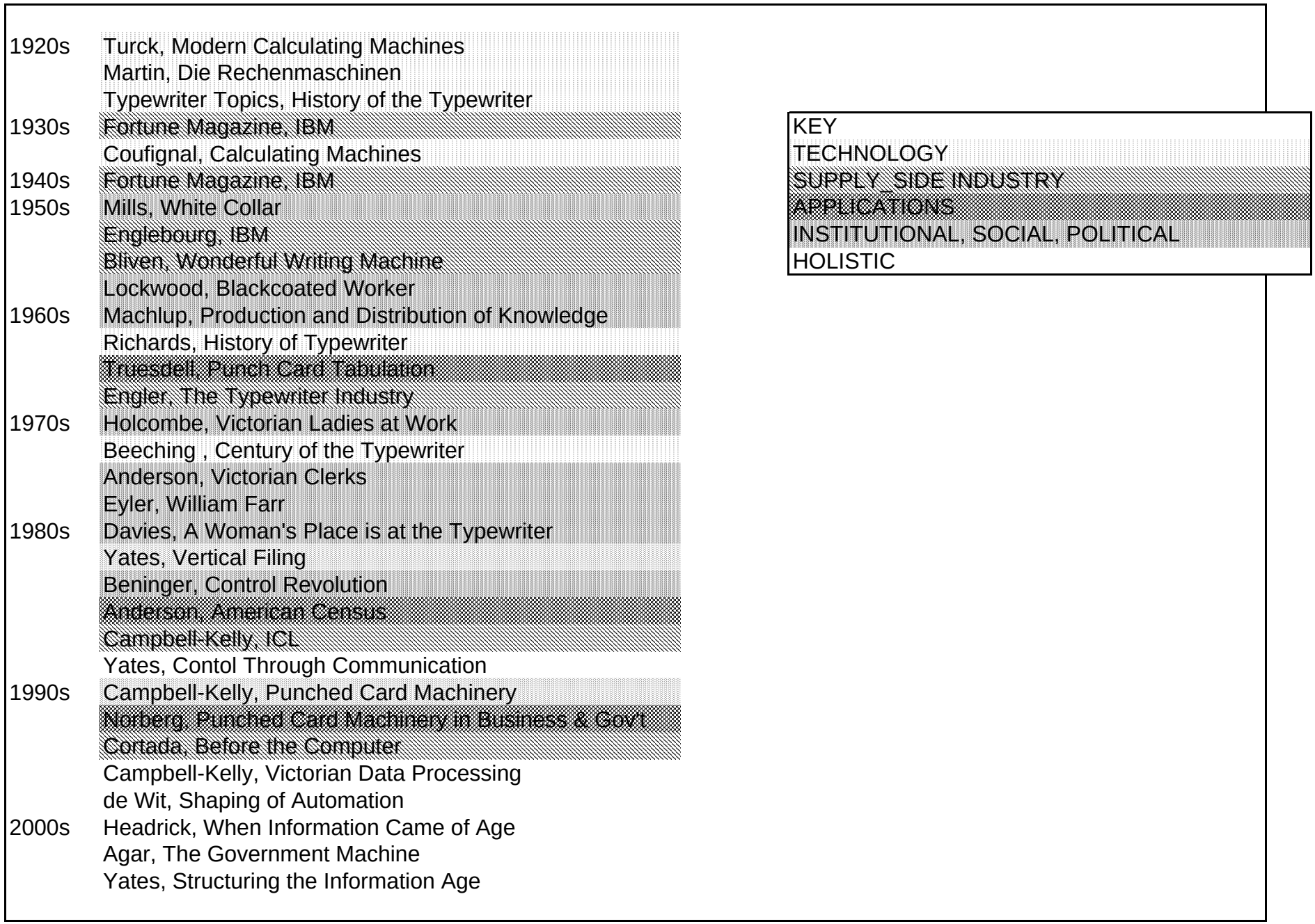

Table 2. Literature on offices and office-based information processing