# Communities of Computing

## *Computer Science and Society in the ACM*

**Thomas J. Misa**

Copyrighted material

# Communities of Computing

## *Computer Science and Society in the ACM*

**Thomas J. Misa, Editor**

*University of Minnesota*

*ACM Books #13*

Copyrighted material

Copyright © 2017 by the Association for Computing Machinery
and Morgan & Claypool Publishers

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopy, recording, or any other except for brief quotations in printed reviews—without the prior permission of the publisher.

Designations used by companies to distinguish their products are often claimed as trademarks or registered trademarks. In all instances in which Morgan & Claypool is aware of a claim, the product names appear in initial capital or all capital letters. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

*Communities of Computing*

Thomas J. Misa, Editor

books.acm.org
www.morganclaypoolpublishers.com

ISBN: 978-1-97000-187-7 hardcover
ISBN: 978-1-97000-184-6 paperback
ISBN: 978-1-97000-185-3 ebook
ISBN: 978-1-97000-186-0 ePub

Series ISSN: 2374-6769 print 2374-6777 electronic

DOIs:

| | | | |
|---|---|---|---|
| 10.1145/2973856 | Book | 10.1145/2973856.2973864 | Chapter 7 |
| 10.1145/2973856.2973857 | Preface | 10.1145/2973856.2973865 | Chapter 8 |
| 10.1145/2973856.2973858 | Chapter 1 | 10.1145/2973856.2973866 | Chapter 9 |
| 10.1145/2973856.2973859 | Chapter 2 | 10.1145/2973856.2973867 | Chapter 10 |
| 10.1145/2973856.2973860 | Chapter 3 | 10.1145/2973856.2973868 | Chapter 11 |
| 10.1145/2973856.2973861 | Chapter 4 | 10.1145/2973856.2973869 | Chapter 12 |
| 10.1145/2973856.2973862 | Chapter 5 | 10.1145/2973856.2973870 | Chapter 13 |
| 10.1145/2973856.2973863 | Chapter 6 | 10.1145/2973856.2973871 | References |

A publication in the ACM Books series, #13
Editor in Chief: M. Tamer Özsu, *University of Waterloo*
Area Editor: Thomas J. Misa, *University of Minnesota*

First Edition

10 9 8 7 6 5 4 3 2 1

Copyrighted material

# Contents

DEFINING THE DISCIPLINE 25

Copyrighted material

Copyrighted material

Copyrighted material

Copyrighted material

# Preface

I like to think the origins of this volume stretch back to a visit that archivist Bruce Bruemmer made to Edmund Berkeley at his home in Newton, Massachusetts. Berkeley was well along in a busy and tumultuous life, as an author, activist, publisher, computer visionary, and co-founder of the Association for Computing Machinery. By my reckoning, he had then published three dozen books. He is of course most famous for the popular and readable *Giant Brains, or Machines That Think* (1949), but his first major work was published more than a decade earlier in 1937 by the American Institute of Actuaries. It was a 40-page treatise bearing the unwieldy title of “Boolean algebra (the technique for manipulating ‘and’, ‘or’, ‘not’, and conditions) and applications to insurance.” Remarkably, it was promptly reviewed in the *Journal of Symbolic Logic* by none other than Alonzo Church. Church was the eminent Princeton mathematician who, just a few months earlier, had recognized the unusual insight contained in a paper on computable numbers by a young Alan Turing (three years Berkeley’s junior). In his review, Church praised Berkeley’s exposition of Boolean algebra “with illustrative examples worked out in detail” and connected to insurance applications. He favorably compared Berkeley’s examples to those given by logician John Venn. Church closed with a warm observation, noting the work’s “novelty lies in the practical application of this technique in a hitherto unsuspected direction.”[1]

I imagine that archivist Bruemmer might have entertained dreams of finding faded correspondence with Church or Venn lurking in Berkeley’s basement, or perhaps page proofs from *Giant Brains,* or possibly his proposal to computerize Prudential Insurance, likely a note from Grace Murray Hopper from their Harvard Mark I days, or even a snippet from Berkeley’s indefatigable anti-war activism. Surely he

---

[1] E. C. Berkeley. 1937. Boolean algebra (the technique for manipulating ‘and’, ‘or’, ‘not’, and conditions) and applications to insurance. *The Record of the American Institute of Actuaries*, 26, part 2, (54): 373–414; reviewed by A. Church. 1938. *J. Symb. Logic*, 3(2): 90.

Copyrighted material

hoped for insight on the early days of ACM. But that day, as Bruemmer later related the story, "he assured me that he had no records." In the archival business, there's no room for impatience. Bruemmer continued his unwearied contacts and was rewarded when, after Berkeley died in 1988, "I personally carried over 100 boxes out of his basement."[2] This was the first significant cache of ACM-related archival records that made their way into a publicly accessible archive; in subsequent years, the Charles Babbage Institute assembled records from ACM officers and awardees culminating with the transfer in 2008 of the ACM headquarters records.

Close study of historical records demands that there be records that are accessible and worthwhile for the historian's attention. In this respect, the first set of acknowledgments for this volume on ACM history is to the donors of valuable historical records as well as the archivists who sought them out. History draws eclectically on many different kinds sources, including photographs, artifacts, oral histories, and published accounts. But the bedrock for historical analysis is the rich detail provided by archival records: the day-by-day, or month-by-month set of correspondence, memos, committee reports, and the like that give deep and particular insight into the past.[3]

A second category of acknowledgments goes to the far-seeing ACM members who created a designated ACM History Committee in 2004. David Wise, Richard Snodgrass, and others used their understanding of ACM to lodge history in the organization's ongoing operation. Since then, the History Committee has commissioned oral history interviews, expanded the A. M. Turing Award website, and sponsored research fellowships that are the immediate cause of the scholarship assembled in this volume. It's been a personal pleasure to serve as chair of this committee and to work with Mary Hall, in one transition of the committee chair, as well as with in-coming History Committee chair Chuck House. The History Committee supported the research appearing in this volume as well as a special workshop at the 2015 annual meeting of the Society for the History of Technology and its history of computing interest group SIGCIS. Many of the contributors benefitted from presenting their research and getting valuable audience feedback in Albuquerque, New Mexico.

---

[2] B. H. Bruemmer and E. Kaplan. 2001. Realizing the concept: A history of the CBI Archives. *IEEE Ann. Hist. Comput.*, 23(4): 29–38, quote on p. 31.

[3] For additional ACM archival records, see ACM Research Materials posted at http://history.acm.org/content.php?do=links; also see T. Haigh, E. Kaplan, and C. Seib. 2007. Sources for ACM history. *Commun. ACM*, 50(5): 36–41.

Copyrighted material

Finally, at the Charles Babbage Institute, I have a number of personal acknowledgments. To begin, I am periodically amazed at the archival treasures on ACM and other computing topics collected over the years by the succession of professional archivists. Bruce Bruemmer collected the Berkeley papers; his successors including Beth Kaplan, Karen Spilman, Carrie Seib, Stephanie Crowe, and Arvid Nelsen have significantly augmented CBI's holdings of ACM related materials. Arvid also followed up his curiosity about ACM's role in educational programs and contributed one of the chapters in this book. Associate director Jeffrey Yost took time from his work to give sage counsel and valuable suggestions. And once again I owe a special thanks to Katie Charlet who assisted capably with the production of this book manuscript (and kept other parts of CBI so well organized that I imagine sometimes that it's a machine that would run itself—except that I know better!).

With the ACM Books venture, I have enjoyed shaping this volume through conversations with editor-in-chief Tamer Özsu, who enthusiastically supported a book on ACM history (once he was assured that this was no unedited conference volume), and with Morgan & Claypool's Diane Cerra and her capable production staff. We have the hope, unusual for a historical volume, of having actual publication within 12 months from the SHOT–SIGCIS conference papers that were the origin of several of these chapters. With good patience and close attention to the calendar, authors have now endured three rounds of revisions, and at last they have my word that this will be the "last" "final" version.

Thomas J. Misa
ACM History Committee (chair 2014–2016)

Copyrighted material

Copyrighted material

# 1

# ACM and the Computing Revolution

## Thomas J. Misa

The Association for Computing Machinery and the computing revolution grew up together, and this book is the first to bring together essential chapters in this remarkable story. It begins in the year 1946, a seminal moment in computing, when the legion of secret wartime efforts in radar, cryptography, advanced electronics, and high-speed communication—on both sides of the Atlantic—began to gel as a public movement to foster the emerging field of digital electronic computing. Britain's wartime proto-computers at Bletchley Park were still under the tight wrap of state security, and Germany's leading wartime effort was buried by a bomb in Berlin. The United States, with its cities undamaged and its economy intact, accordingly took an early lead in computing. In February 1946 the U.S. Army staged a high-profile public display of the ENIAC, a vacuum-tube wonder built at the University of Pennsylvania's Moore School of Engineering that is frequently hailed at the world's first electronic computer. That spring the two men most responsible for the technical design of ENIAC left the University of Pennsylvania to found what became the much-storied Univac company. A few months later in the summer, 50 or so of the world's computing experts spent eight weeks in Philadelphia at the Moore School learning about ENIAC and the exciting plans for a successor machine called EDVAC that promised true programmability.[1]

The computing community soon expanded to several hundred engineers, mathematicians, military officers, scientists, and a few notable pioneer programmers, who assembled in the spring of 1947 at notable computing conferences at Harvard

---

[1] F. M. Verzuh. Moore School of Electrical Engineering lecture notes (July 8–August 31, 1946), Charles Babbage Institute (CBI 51), http://purl.umn.edu/41379; T. Haigh, M. Priestley and C. Rope. 2016. *ENIAC in Action: Making and Remaking the Modern Computer.* MIT Press, Cambridge, MA.

Copyrighted material

and at MIT, as well as a meeting convened by the New York section of the American Institute of Electrical Engineers (AIEE). That fall in mid-September, Columbia University hosted the inaugural meeting of the "Eastern Association for Computing Machinery" where it proclaimed a lofty aim to "advance the science, development, construction, and application of the new machinery for computing, reasoning, and other handling of information."[2]

The "principal person in the founding of ACM" was Edmund Berkeley, an inveterate organizer, gadfly, and computer enthusiast, soon to publish a popular book with the provocative and anthropomorphic title *Giant Brains, or Machines That Think* (1949), and he brought together 57 people for that inaugural meeting at Columbia.[3] Within a few months, as soon as enough members from across the country signed up, the qualification "eastern" was dropped and the simply named Association for Computing Machinery expanded to 400 members. Frances Holberton, one of the original six ENIAC programmers, recalled that ACM's founding was "one of the things that started the ball rolling . . . the beginning of everybody talking to each other."[4]

With computers the size of offices, and the entire world's computing community then able to fit inside a modest high-school gymnasium, it was at the time impossible to see the future whose foundations were being formed. Today, with computers small enough to put into your pocket—even if the largest floors of web servers or supercomputing complexes are still of immense physical size—the ACM membership numbers 100,000 members around the world. Its largest special interest groups, or SIGs, are each the size of a respectable professional society. Its flagship journal, *Communications of the ACM*, is the go-to publication for monthly news about the computing profession and far-reaching technical advances. And the ACM's Turing Award, named for Alan Turing and first bestowed in 1966 to Alan Perlis, is widely acknowledged to be computing's "Nobel Prize."

In the seven decades between those founding events and today, it goes without saying that the landscape of computing has been transformed. Desktop and laptop computers alone are estimated to consume a 3% share of North America's entire

---

[2] A. Akera. 2007. Edmund Berkeley and the origins of ACM. *Commun. ACM*, 50(5): 30–35. DOI: 10.1145/1230819.1230835; B. Longo. 2015. *Edmund Berkeley and the Social Responsibility of Computer Professionals.* Association for Computing Machinery and Morgan & Claypool, New York.

[3] P. J. Denning. 1988. Edmund C. Berkeley—ACM founder. *Commun. ACM*, 31(6): 781–782.

[4] UNIVAC Conference, Charles Babbage Institute, May 17–18, 1990, Washington, DC, Charles Babbage Institute, OH 200, quotes pp. 100–101. Available at http://purl.umn.edu/104288.

Copyrighted material

Figure 1.1 **Computer visionary and ACM founder Edmund Berkeley.** (Image courtesy of the Charles Babbage Institute Archives, University of Minnesota Libraries)

electricity supply, with giant data centers and server farms taking at least as much or more. The computer-based world of video gaming has spawned an industry that is larger than the movie and music industries combined. Virtual and augmented realities along with the promises of artificial intelligence and the internet of things are bringing about deep social and cultural changes. The recurrent concerns about the automation of muscle and brain work speak also to the economic and cultural

Copyrighted material

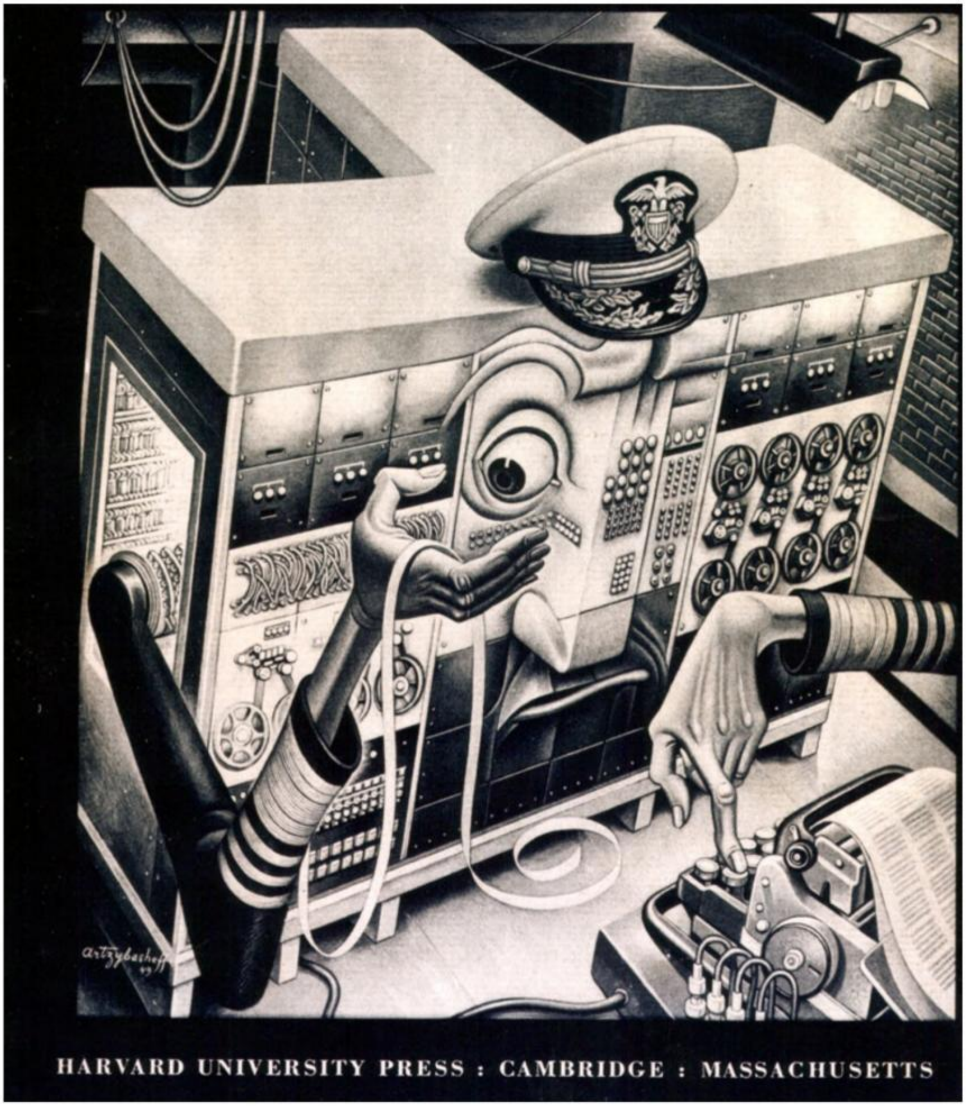


**Figure 1.2** **Harvard Mark 3 computer as anthropomorphic "giant brain," following Berkeley.** (Image courtesy of the Charles Babbage Institute Archives, University of Minnesota Libraries)

Copyrighted material

impact of computing. We are still responding to a brave new world where, by some estimates, an iPhone 4 in your pocket has roughly the processing muscle of a 1980s Cray 2 supercomputer (respectively, 1.6 and 1.9 giga-flops).[5]

Besides the technical landscape, the institutional landscape of computing has been transformed; computing directly engages a large share of the world's professional and technical brainpower. For organizing this brainpower ACM's chief competitor today is the IEEE Computer Society, a successor to that early New York meeting of the AIEE. These two societies—ACM and IEEE Computer Society—tower over the more specialized or regional-based computing societies. They are the long-term winners in defining computing as a professional enterprise and scientific field. Several other organizations were for a time undeniably successful. These include the Data Processing Management Association (DPMA), with a huge membership from 1962–1996 when it was reorganized as the Association of Information Technology Professionals (AITP), still active today. The American Federation of Information Processing Societies (AFIPS), in which ACM cooperated with the American Institute of Electrical Engineers (AIEE) and Institute of Radio Engineers (IRE), ran the most important national computer conferences for 25 years (1962–1987) and its international parent, the International Federation for Information Processing (IFIP), is active today and, indeed, figures in several chapters in this book. AFIPS also took a prominent role in shaping the early history of computing by sponsoring *Annals of the History of Computing*, the field's preeminent scholarly journal, and for a time supporting the Charles Babbage Institute, a leading computer history and archiving center. (AIEE merged with IRE to form the IEEE in 1963 and its computing activities evolved to form the IEEE Computer Society in 1971.)

Several other professional and scientific organizations remain prominent in computing today. These include the Society for Industrial and Applied Mathematics (SIAM), founded in 1952, the same year as the predecessor to the DPMA held its first annual conference; the Canadian Information Processing Society, tracing its origins to 1958 and examined in Chapter 8 of this book; and a bevy of organizations founded in the 1970s: the Computing Research Association, a network of the leading computer science departments in North America; USENIX, originally a UNIX users group and today a sponsor of lively conferences; and the Association for Women in Computing which, along with the Anita Borg Institute for Women in

---

[5] L.-B. Desroches, et al. 2014. Computer usage and national energy consumption: Results from a field-metering study, Lawrence Berkeley National Laboratory, December 1. Available at http://eetd.lbl.gov/sites/all/files/computers_lbnl_report_v4.pdf; Processing Power Compared at http://pages.experts-exchange.com/processing-power-compared (accessed June 2016).

Copyrighted material

**Figure 1.3** **ACM membership booth at Spring Joint Computer Conference (c. 1978).** (Image courtesy of the Charles Babbage Institute Archives, University of Minnesota Libraries)

Technology and the ACM's Committee on Women in Computing, among others, are key advocates for women in computing. The Anita Borg Institute and ACM are partners in producing the annual Grace Hopper Celebration of Women in Computing (see Chapter 7).

## 1.1 History in Computing

Despite a common perception that computer scientists lack historical awareness, history featured surprisingly large within ACM long before it founded a designated History Committee in 2004. The periodic History of Programming Languages conferences (1978, 1993, 2007) were high-profile efforts organized by ACM's special-interest group in programming languages, or SIGPLAN. The "prime mover" behind the HOPL effort according to Tim Bergin was Jean Sammet (see Figure 6.1), who as early as 1966 had published "The Use of English as a Programming Language" in

Copyrighted material

*Communications of the ACM*. Sammet, already known as the developer of FORMAC and a leading figure in SIGPLAN, went on to write a pioneering book, *Programming Languages: History and Fundamentals* (1969), as well as serve as ACM president (1974–1976). Intriguingly, Sammet's early computing career can be traced to her work for Metropolitan Life Insurance Company in the 1950s, a neat parallel to Edmund Berkeley's work for Prudential Insurance. It is worth pointing out that Bergin was not only a participant in (and historian of) the HOPL conferences; later he was also editor-in-chief of *Annals of the History of Computing*.[6]

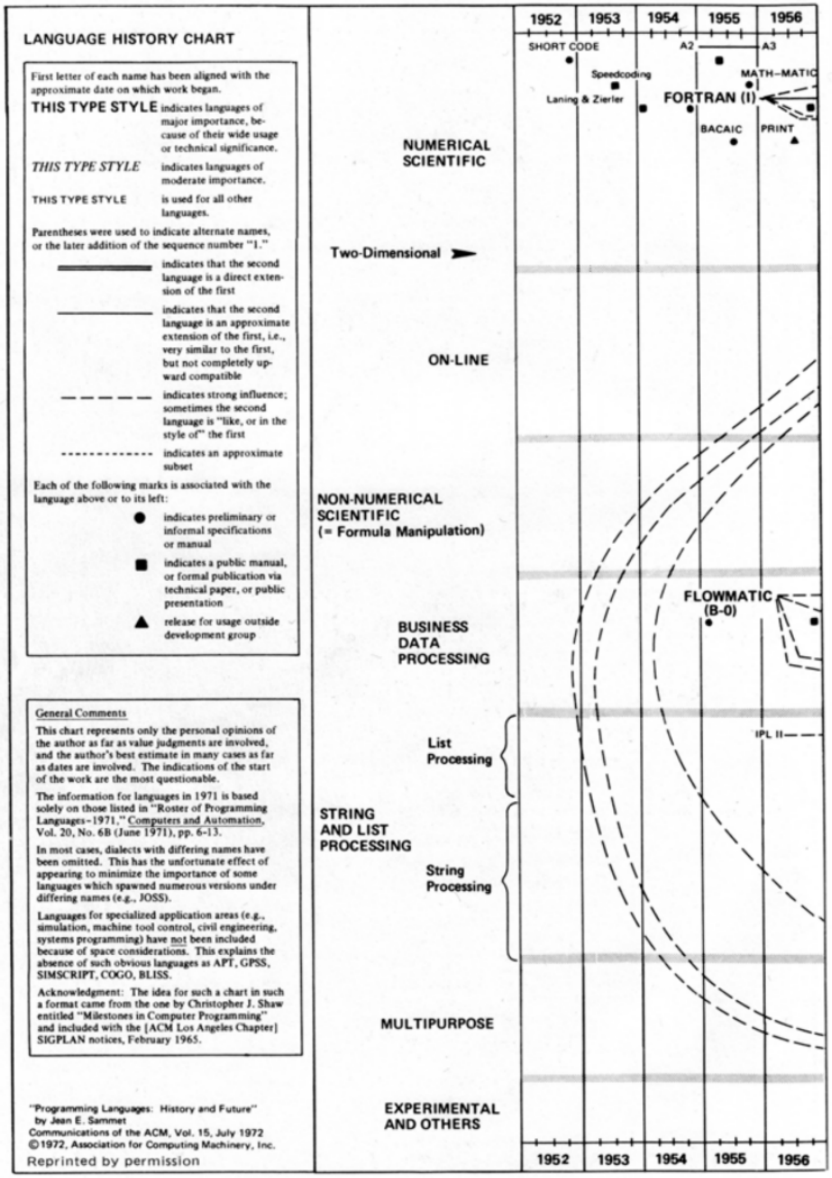
The recipe for HOPL's success with these conferences was simple: assemble a notable gallery of well-known programming-language developers, ask them to carefully structure their presentations, let them relate their captivating histories, and leaven the event with cameo appearances by such figures as Grace Hopper and Fred Brooks. The inaugural HOPL featured 13 such languages with talks by John Backus on FORTRAN, Alan Perlis and Peter Naur on ALGOL, John McCarthy on LISP, and Jean Sammet on COBOL, among other luminaries; the conference closed by announcing the launch of *Annals of the History of Computing*. HOPL 2 featured Niklaus Wirth on PASCAL, Barbara Liskov on CLU, Alan Kay on Smalltalk, Dennis Ritchie on C, and Bjarne Stroustrup on C++. HOPL 2 also featured input and participation by Michael Mahoney, the Princeton historian of mathematics and computing who served as conference historian. Mahoney presided over a closing panel—"The History of Programming: Does Our Present Past Have a Future?"—designed to elicit greater awareness of the different types of and diverse viewpoints on history.[7] Some of HOPL's featured languages, even though carefully chosen and deemed central at the time, are not so well known today; for HOPL 1 the language

---

[6] J. E. Sammet. 1966. The use of English as a programming language. *Commun. ACM*, 9(3): 228–230. DOI: 10.1145/365230.365274; T. J. Bergin. 2007. A history of the history of programming languages. *Commun. ACM*, 50(5): 69–71. DOI: 10.1145/1230819.1230841; History of Programming Languages Conference Records, Charles Babbage Institute (CBI 19), http://purl.umn.edu/40668. For a deeper history, see D. Nofre, M. Priestley, and G. Alberts. 2014. When technology became language: the origins of the linguistic conception of computer programming, 1950–1960. *Technol. Cult.*, 55(1): 40–75.

[7] Mahoney played a key role in developing *Annals of the History of Computing* as a scholarly publication. His key publications are collected in T. Haigh, editor. 2011. *Histories of Computing.* Harvard University Press, Cambridge, MA. Mahoney's HOPL panel echoes a programmatic essay by historian of science Arnold Thackray, "Science: Has Its Present Past a Future?" in R. H. Stuewer, editor. 1970. *Minnesota Studies in the Philosophy of Science*, vol. 5, pp. 112–127. University of Minnesota Press, Minneapolis, MN. See the HOPL materials in Michael S. Mahoney Papers, Charles Babbage Institute (CBI 213), http://purl.umn.edu/92154 and W. Aspray. 2014. Michael Sean Mahoney (1939–2008), *IEEE Ann. Hist. Comput.*, 36(3): 70–79.

Copyrighted material



LANGUAGE HISTORY CHART

First letter of each name has been aligned with the approximate date on which work began.

| | |
|---|---|
| **THIS TYPE STYLE** | indicates languages of major importance, because of their wide usage or technical significance. |
| *THIS TYPE STYLE* | indicates languages of moderate importance. |
| THIS TYPE STYLE | is used for all other languages. |

Parentheses were used to indicate alternate names, or the later addition of the sequence number "1."

| | |
|---|---|
| ═══════ | indicates that the second language is a direct extension of the first |
| ─────── | indicates that the second language is an approximate extension of the first, i.e., very similar to the first, but not completely upward compatible |
| — — — — — | indicates strong influence; sometimes the second language is "like, or in the style of" the first |
| - - - - - - - - - - - - | indicates an approximate subset |

Each of the following marks is associated with the language above or to its left:

| | |
|---|---|
| ● | indicates preliminary or informal specifications or manual |
| ■ | indicates a public manual, or formal publication via technical paper, or public presentation |
| ▲ | release for usage outside development group |

General Comments

This chart represents only the personal opinions of the author as far as value judgments are involved, and the author's best estimate in many cases as far as dates are involved. The indications of the start of the work are the most questionable.

The information for languages in 1971 is based solely on those listed in "Roster of Programming Languages-1971," Computers and Automation, Vol. 20, No. 6B (June 1971), pp. 6-13.

In most cases, dialects with differing names have been omitted. This has the unfortunate effect of appearing to minimize the importance of some languages which spawned numerous versions under differing names (e.g., JOSS).

Languages for specialized application areas (e.g., simulation, machine tool control, civil engineering, systems programming) have not been included because of space considerations. This explains the absence of such obvious languages as APT, GPSS, SIMSCRIPT, COGO, BLISS.

Acknowledgment: The idea for such a chart in such a format came from the one by Christopher J. Shaw entitled "Milestones in Computer Programming" and included with the [ACM Los Angeles Chapter] SIGPLAN notices, February 1965.

"Programming Languages: History and Future"
by Jean E. Sammet
Communications of the ACM, Vol. 15, July 1972
©1972, Association for Computing Machinery, Inc.
Reprinted by permission

**Figure 1.4** **Timelines of Programming Languages from HOPL I.** (Image courtesy of the Charles Babbage Institute Archives, University of Minnesota Libraries)

Copyrighted material

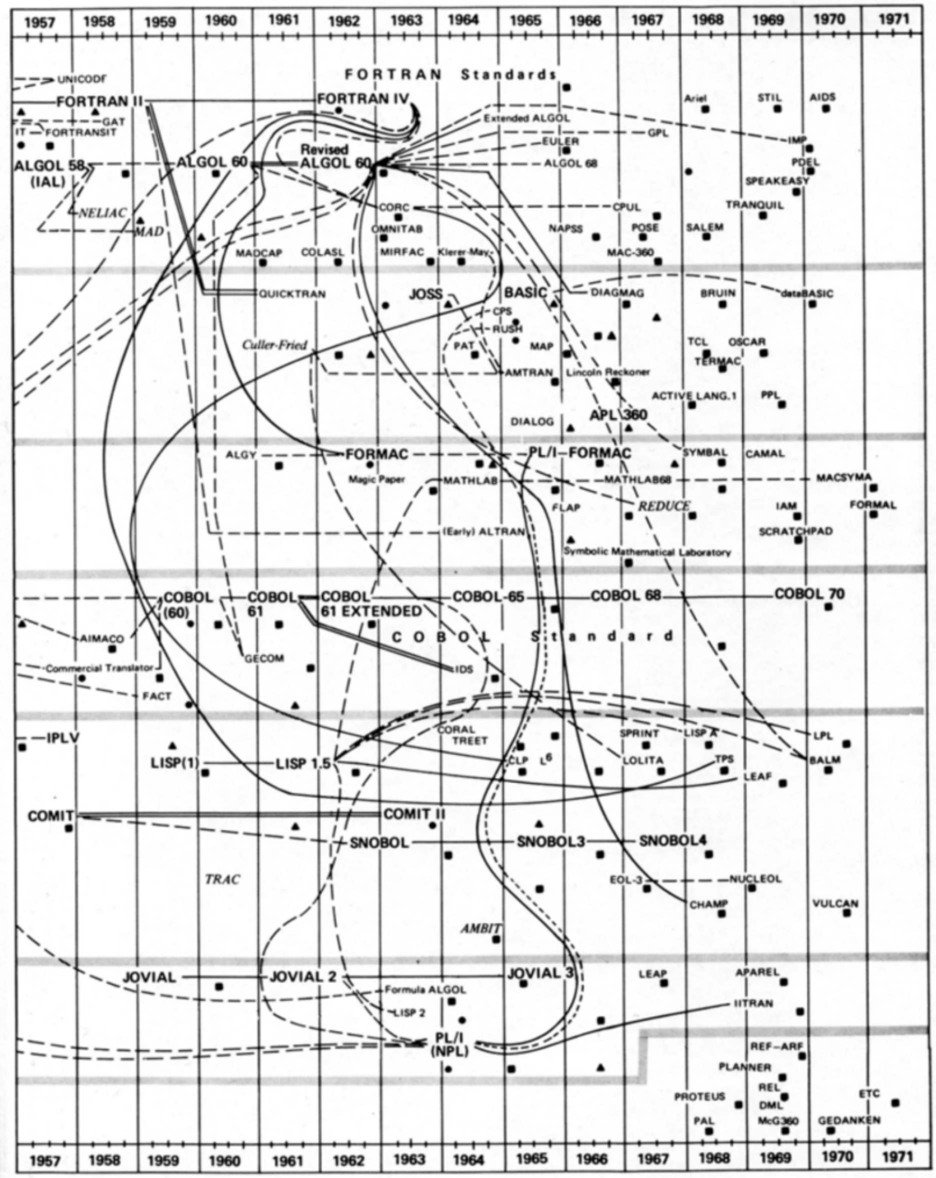


**Figure 1.4** *(continued)*

Copyrighted material

JOVIAL was chosen as an outgrowth of ALGOL 58 and as the "first language to include significant numerical, logical, and data handling capabilities." This is a gentle reminder that history is always written at the present moment, whenever that might be, and reflects the concerns and perspectives of that moment.

Another of ACM's larger special interest groups that actively cultivated its history was SIGGRAPH, which was organized in 1969 by Andy van Dam and held periodic conferences beginning in the 1970s with a notably successful conference in San Jose in 1977 that brought together academic researchers with the emerging graphics industry (see Chapter 11). Doing the math on SIGGRAPH's 25th anniversary in *1998*, you get a ready illustration of the varied perspectives on history: while SIGGRAPH was organized as a full-blown special-interest group in 1969, and its 1977 conference was a breakthrough moment, Carl Machover and the other SIGGRAPH organizers of the 25th anniversary chose instead the origin to be the 1973 conference, a signal that the first in a series of annual conferences is often understood as an origin moment in computing.[8]

## 1.2 History of Computing

The history of computing as a distinct field with scholarly activity and practitioner interest also took form during the HOPL years. As noted above, historian of science Michael Mahoney played a key role in connecting the HOPL conferences with the community of professional historians. Mahoney, originally a historian of science at Princeton University and author of a biographical study of mathematician Pierre Fermat, was in his career increasingly drawn toward the history of computing and the history of technology. Another practitioner–historian who shaped the field was the University of Michigan's Bernie Galler, the first editor of *Annals of the History of Computing*. (Sponsorship of the journal passed to the IEEE Computer Society in 1992.[9]) Moreover, Galler authored one of two finalist proposals for hosting of the Charles Babbage Institute, which had been organized by Erwin Tomash and associates in California in the late 1970s as a research and archiving center. The winning proposal to host Charles Babbage Institute (CBI), as it turned out, was

---

[8] See files on SIGGRAPH '98 in Carl Machover Papers, Charles Babbage Institute (CBI 206), http://purl.umn.edu/98190.

[9] *Annals* editors-in-chief include Bernard A. Galler, 1979–1987; J.A.N. Lee, 1987–1995; Michael R. Williams, 1996–2000; Tim Bergin, 2000–2003; David A. Grier, 2004–2007; and Jeffrey R. Yost, 2008–2011.

Copyrighted material

created by historian of science Roger Stuewer at the University of Minnesota.[10] In subsequent years, under the direction of Arthur Norberg, CBI took a keen interest in archival records created by ACM's officers and luminaries, including such figures as Edmund Berkeley, Alan Perlis, Charlie Bachman, and others, as well as the notable HOPL conferences, and in 2008 became the public repository of the ACM headquarters archives.[11] This aspect of ACM's history is consequential for this present volume, since access to archival records creates an essential precondition for historical scholarship, and many of this book's chapters draw on these publicly accessible ACM records.

The 1980s was a notable decade for the history of computing, even beyond the early years of *Annals of the History of Computing* and the CBI. CBI was one of several so-called disciplinary history centers, which at the time formed something of a "cohort" and a real-time experiment in connecting academic history to the concerns of professionals and practitioners. Compared with AFIPS, ACM played a rather modest and somewhat distant role in this development. Within a few short years four such disciplinary history centers were launched. In addition to CBI these included centers formed by the IEEE, parent of the IEEE Computer Society; by the American Institute of Physics; and by the American Chemical Society in cooperation with the American Institute of Chemical Engineers. Partnerships with prominent research universities emerged as a common attribute of these centers: these were, respectively, Minnesota, Rutgers, Maryland, and Pennsylvania. NASA had created a NASA history office a year after its founding in 1958, and it too expanded in the 1980s "to support a much wider range of scholarly activities than at any time in the past."[12] All these efforts included archival collecting, oral history programs, and a varied mix of research and commemorative activities.

---

[10] For the founding and early years of CBI, see *CBI Newsletter* articles by A. Norberg. Fall 2003. Twenty-Five Years of the Charles Babbage Institute. *CBI Newsletter* 26(1). Available at www.cbi.umn.edu/about/nsl/v26n1graphics.pdf#page=3; and T. Misa. Fall 2012 and Spring 2013. Charles Babbage Institute: Glimpses of the founding. *CBI Newslett.*, 34(2) and 35(1). Available at www.cbi.umn.edu/about/nsl/v34n2.pdf#page=4 and www.cbi.umn.edu/about/nsl/v35n1.pdf#page=19.

[11] See CBI holdings of ACM headquarters records at http://purl.umn.edu/51982; other ACM units at http://tinyurl.com/jran22g; ACM members at http://tinyurl.com/gvx9eq3; and C. Bachman papers at http://purl.umn.edu/40732. See also "ACM Research Materials" at http://history.acm.org/content.php?do=links as well as T. Haigh, E. Kaplan, and C. Seib. 2007. Sources for ACM history. *Commun. ACM*, 50(5): 36–41. DOI: 10.1145/1230819.1230836.

[12] R. D. Launius. 1999. NASA history and the challenge of keeping the contemporary past. *Pub. Hist.*, 21(3): 63–81, quote p. 69.

Copyrighted material

For the history of computing, the 1980s was a notable decade also in the museum world. Work was underway at the National Museum of American History on the landmark exhibit "Information Age: People, Information and Technology," which opened in May 1990 and led to a series of Smithsonian interviews with such notable pioneers as Seymour Cray, Bill Gates, and Steve Jobs. In 1984 the Computer Museum in Boston opened its doors to the public in a high-profile location on the Boston waterfront. Supported originally by the Digital Equipment Corporation, one of the mainstays of Boston's high-technology Route 128, the Computer Museum felt the nation's shifting fortunes as the Boston area's minicomputer industry waned and as California's Silicon Valley grew in economic importance. In 1999–2000, the Computer Museum in Boston was closed and most of its collection shipped to NASA Ames's Moffett Field in California. In 2002, the newly organized Computer History Museum acquired its present location in nearby Mountain View and prepared the way for its main public exhibit, "Revolution: The First 2000 Years of Computing," which opened to great acclaim in 2011.

In 2004, at the prompting of historically minded members David Wise and Rick Snodgrass, the ACM Council created a History Committee and charged it with "foster[ing] preservation and interpretation of the history of the ACM and its role in the development of computing." After a period of organizational work bringing together ACM members with professional historians from the academic and museum worlds, the History Committee embarked on commissioning a set of oral histories with ACM officers and prize winners. Greater visibility of the A. M. Turing Award prompted a multi-year project to expand the prize's website as a means to garner visibility for the awardees and expand recognition of the ACM's role in computing, an effort that has been further expanded with the commissioning of a comprehensive set of oral histories with all living Turing awardees. Another committee activity that directly shaped this present volume is an annual program of fellowships to facilitate historical research on ACM's history. Nearly two dozen fellowships so far have supported a wide variety of research projects done by computer scientists, educators, and academic historians. The results include five completed doctoral dissertations, two published books, and numerous conference presentations, journal articles, and, not least, the chapters in this book. Two hands-on workshops organized by the committee—on archival principles and practices (2014) and on oral-history methods (2016)—have connected ACM members and other computing practitioners with up-to-date professional practices in history. A third research-oriented workshop, held in conjunction with the annual meeting of the Society for the History of Technology (2015) and in cooperation with its special interest group

Copyrighted material

in computing history, or SIGCIS, was an opportunity to present, discuss, and refine nine of the chapters in this book.[13]

## 1.3 Chapters of ACM History

This book deals with ACM's organizational history, with its historic role in defining "computer science" in the 1960s and its evolution in subsequent decades, with some of its notable special interest groups, and with several prominent research activities. Many of the chapters deal with "science and society" issues, including the debates about how ACM should deal with controversial social and political issues. It is not, however, a formal history of ACM viewed from a unified perspective.[14] Instead, readers will find here a set of chapters that tells the story of intriguing aspects of computing history—where, for each, the ACM played a prominent role. The perspective is a mosaic rather than a linear timeline; the book presents compelling episodes rather than a single cohesive narrative.

The book's first theme, "Defining the Discipline," examines the 1960s and 1970s as the field of computer science was taking something like its presently recognizable form. At the time, computing projects in government, industry, and academia were conducted by people with multiple and distinct disciplinary identities, including mathematics, engineering, physical science, library science, and data processing. Appropriately enough, Janet Abbate's chapter 2 situates the emergence of computing as a proper discipline within the National Science Foundation. The problem was that computing figured in *multiple* NSF programs, including the agency's funding and support for specific computing projects in varied scientific fields, for general campus computing resources, and for mathematical research. Abbate follows the engagement in these debates of ACM presidents Alan Perlis, George Forsythe, and Anthony Oettinger, as NSF evolved initially its dedicated Office of Computing Activities by 1967 and, eventually, the full-scale directorate

---

[13] M. Hall. 2012. Understanding ACM's past. *Commun. ACM* 55(12): 5. DOI: 10.1145/2380656.2380657; and T. J. Misa. 2015. Computing is history. *Commun. ACM*, 58(10): 35–37. DOI: 10.1145/2814845.

[14] Formal histories of several of the engineering societies include B. Sinclair. 1980. *A Centennial History of the American Society of Mechanical Engineers, 1880–1980*. University of Toronto Press, Toronto; T. S. Reynolds. 1983. *75 Years of Progress: A History of the American Institute of Chemical Engineers, 1908–1983*. American Institute of Chemical Engineers, New York; A. M. McMahon. 1984. *The Making of a Profession: A Century of Electrical Engineering in America*. Institute of Electrical and Electronics Engineers, New York.

Copyrighted material

for Computer and Information Science and Engineering (CISE). She also assesses ACM's 1968 model curriculum for computer science.

It is worth recalling that the various terms in use, including computer sciences, information sciences, data processing, and the singular "computer science," were diversely defined, and that a lively debate ensued about whether computing was best considered to be a useful scientific tool, a bounded scientific domain, or a set of novel phenomena calling out for scientific understanding. "Computer science is the study of computers . . . the computer is not just an instrument but a phenomenon as well, requiring description and explanation," asserted three Carnegie Institute of Technology luminaries—Allen Newell, Alan Perlis, Herbert Simon—while Edsger Dijkstra, a friendly rival, countered that "Computer science is no more about computers than astronomy is about telescopes." Donald Knuth, author of the monumental and multi-volume *Art of Computer Programming* (1968 et seq.), offered a definition that explicitly aligned with his Carnegie colleagues: "computer science is the study of algorithms . . . one of the 'phenomenon surrounding computers'."[15] This spirited discussion, one might observe, involved *five* current or future Turing awardees (and with Simon also a Nobel Prize).

For its part, the ACM "Curriculum 68" report pragmatically defined computer science not with any single essence—formal theory, computer programming, and algorithms were possible contenders[16]—but rather by describing three dimensions of the emerging field. "Information structures and processes" centered on data structures, programming languages, and models of computation. "Information processing systems" pointed to computer design and organization, translators and interpreters, and computer operating systems. And "methodologies" enumerated applications such as numerical mathematics, symbol manipulation, data processing and file management, graphics, simulation, information retrieval, and artificial intelligence that featured common structures, processes, and techniques.[17] Text-

---

[15] A. Newell, A. J. Perlis, and H. A. Simon. 1967. Letters: computer science. *Science*, 157(22): 1373–1374 at http://www.jstor.org/stable/1723308; P. J. Denning and C. H. Martell. 2015. *Great Principles of Computing*, quote p. 1 (Edsger Dijkstra). MIT Press, Cambridge, MA; D. E. Knuth. 1974. Computer science and its relation to mathematics. *The American Mathematical Monthly*, 81:323–343, on 324.

[16] On the decades-long quest for the "essence of computing as a discipline," see M. Tedre. 2015. *The Science of Computing: Shaping a Discipline*, quote p. 5. CRC Press, Boca Raton.

[17] W. F. Atchison, et al. 1968. Curriculum 68: Recommendations for academic programs in computer science: A report of the ACM Curriculum Committee on computer science. *Commun. ACM*, 11(3): 151–197, quotes on 154–155. DOI: 10.1145/362929.362976.

Copyrighted material

books and encyclopedias largely followed this inclusive and enumerative definition of the field.[18]

With attention to these national policy discussions, including high-profile reports by the National Academy of Science and other blue-ribbon bodies, Abbate narrates the emergence of computing as a legitimate scientific discipline, worthy of research funding and justified on its own terms. Her discussion is essential background for understanding ACM's role in defining computer science and its continuing role in shaping computer science education. It also serves as a balance to historical accounts focusing on the prominent role of the Defense Department's legendary Information Processing Techniques Office in supporting computer graphics, time sharing, artificial intelligence, and networking.[19]

Joseph November's chapter 3 complements and extends Abbate's. Forsythe was the founder of computer science at Stanford University (1965) as well as ACM president during the years 1964–66. "He, more than any other man, is responsible for the rapid development of computer science in the world's colleges and universities," according to Donald Knuth. November details Forsythe's shifting discipline- and institution-building strategies, as he himself made a career transition from mathematics and meteorology to computing and computer science. As late as 1960, a textbook he co-authored on differential equations envisioned computers as a tool for mathematics rather than an object of scientific study in its own right. His participation the following year in a new "Division of Computer Science" within Stanford's math department raised the question of how to justify and consolidate the fledging endeavor, especially since some influential Stanford faculty thought computing was too much like "plumbing" and lacked respectability as an intellectual field. Stanford's "Computer Science Department" was formally created in 1965 by the legendary Fred Terman, hailed as the founder of Silicon Valley, as a means to expand computing and re-orient the university toward externally funded research. Forsythe was its founding chair. It's odd to hear today, but Stanford faced obstacles to hiring John McCarthy and Edward Feigenbaum into the mathematics department, and so the new department became a free-standing entity in the School of Humanities and Sciences (it moved to the engineering school in 1985). Forsythe's early

---

[18] See S. V. Pollack, editor. 1982. *Studies in Computer Science*, pp. 31, 35–48. Mathematical Association of America, Washington, DC; A. Ralston and C. L. Meek, editors. 1976. *Encyclopedia of Computer Science*, s.v. "computer science," pp. 314–318. Petrocelli/Charter, New York.

[19] A. L. Norberg and J. E. O'Neill. 1996. *Transforming Computer Technology: Information Processing for the Pentagon, 1962–1986.* Johns Hopkins University Press, Baltimore.

Copyrighted material

and unexpected death in 1972, after the first part of his vision was clearly achieved, in November's view hampered achievement of his second and wider vision, which was computer science as a thoroughly interdisciplinary and university-spanning activity, connected to many existing departments and research centers.

Forsythe's daughter, an anthropologist, observed that computer science at Stanford during these years was distinctly inhospitable to women. Irina Nikivincze's chapter 4 picks up exactly this theme through a collective biography of American women who received their doctoral degrees in computer science between 1970 and 1976, years when women received around 7% of the new field's doctoral degrees (and around 15% of master's degrees and a slightly greater proportion of undergraduate degrees). Nikivincze sets computer science into the wider context of women's participation in other scientific fields. She has identified 30 women receiving Ph.D. degrees in computer science from 1970–1976 and subsequently pursuing academic careers (an additional 60 women doctorates went into industry or pursued other careers). Of the 30, she personally did interviews with 7 (3 additional women had been previously interviewed), providing detailed information about a sample of women at an early and formative period of computer science. She found them to be "extraordinary group driven by their passion for the subject and the desire to succeed." The chapter reports on their diverse experiences as they negotiated graduate school and early careers, developed lines of research, struggled for professional recognition, and (for a number) raised families in the midst of busy lives. Membership in, and activities with, ACM and other professional societies are a prominent feature. Her chapter's empirical findings complement a recent book by Abbate on women in computing as well as interviews conducted by the Charles Babbage Institute on women in the computing industry.[20]

ACM played a central role in defining "computer science" as an educational activity; sometimes the field's emergence is even dated to the ACM model curriculum issued in 1968. Sebastian Dziallas and Sally Fincher's chapter 5, "The History and Purpose of Computing Curricula (1960s to 2000s)," makes clear that ACM's curriculum-shaping activities continued far beyond the 1960s and helps in understanding the transformations of computing education during these decades. Their chapter examines the ten major reports issued by ACM and, beginning in 1983–84, in parallel with and then jointly with the IEEE Computer Society. While providing

---

[20] J. Abbate. 2002. *Recoding Gender: Women's Changing Participation in Computing.* MIT Press, Cambridge, MA; existing interviews with women by the Charles Babbage Institute (available at: http://tinyurl.com/jufd2zs) and a further set of 40 CBI interviews supported by the Alfred P. Sloan Foundation.

Copyrighted material

an excellent overview of the evolution of computer-science education, their chapter is the first in the book to consider the inner workings of ACM—especially the need to coordinate the legion of volunteer committee members as they go about the painstaking business of investigating, developing, revising, and issuing these much-anticipated curriculum reports. Of course model curricula do not evolve in a vacuum, and their chapter considers the recurrent questions about accreditation, including shifting relationships with the Accreditation Board for Engineering and Technology, or ABET, the engineering-oriented accreditation body. To balance the natural visibility of Stanford, MIT, Carnegie Mellon, Purdue, and other research-oriented programs, they devote attention to the ACM's Special Interest Group on Computer Science Education (SIGCSE) and its Committee on Computing Education in Liberal Arts Colleges, where many computer-science faculty teach and where many students take undergraduate degrees.

The book's second organizing theme, "Broadening the Profession," looks outward into the wider society as ACM members and the organization itself engaged with social and political issues—and struggled with balancing a focus on scientific issues with an engagement with the wider world. ACM members did not always agree on the merits or importance of each issue, of course; but it is notable that a range of viewpoints was actively considered, especially compared with other professional engineering societies. In several instances, notably women in the computing profession and extending computing into economically disadvantaged communities, the ACM stands apart from its sister engineering societies.[21] Janet Toland's chapter 6 examines the ACM's efforts in the turbulent years of the 1960s and 1970s to retain its scientific stature (and tax-free status) with some members' desires to have ACM more squarely confront the Vietnam war, the women's movement, and the human rights of Soviet computer scientists. Toland traces the emergence and internal working of ACM's Special Interest Group on Computers and Society, or SIGCAS, seemingly designed to discuss and engage social issues involving computing, and describes its relations with the ACM leadership and the society's public policy committee. ACM founder Edmund Berkeley and ACM member Daniel McCracken each lobbied strenuously for ACM's adopting an overly political stance about the use of computing in warfare. Peter Denning, Jean Sammet, and other ACM leaders at the time aimed to keep the ACM focused on scientific and technical matters

---

[21] Compare with A. M. McMahon. 1984. *The Making of a Profession: A Century of Electrical Engineering in America*, pp. 253–263. Institute of Electrical and Electronics Engineers, New York; and M. Wisnioski. 2012. *Engineers for Change: Competing Visions of Technology in 1960s America*, pp. 111–121. MIT Press, Cambridge, MA.

Copyrighted material

rather than political issues. Drawing on many personal papers with behind-the-scenes correspondence, Toland's account is full of colorful dialog and delightful detail. By and large ACM engaged with "political issues" only where there was a clear connection to computing, for example with computer-related privacy issues. Her account neatly frames later chapters in this section dealing with other questions of social and political importance.

The women's movement resonated deeply, if unevenly, across the ACM membership as the internal and national debate about the Equal Rights Amendment certainly indicates. Amy Sue Bix's chapter 7 adds an institutional and comparative perspective. Bix, like Nikivincze, sets her account within a rich history of women in science and engineering. Her chapter presents an extended comparison of the Association for Women in Computing (AWC) with ACM's Committee on Women in Computing (ACM-W). In examining the AWC's activities in the late 1970s and 1980s, Bix finds that the fledgling organization, independent from other professional bodies, struggled to focus its attention and resources. Beginning in the 1990s, ACM-W was an informal lobby within ACM for women's issues and in 2006 it was elevated to full Council status; it benefitted mightily from the deep resources of the ACM and the institutional clout that came through its auspices. She indicates a contrast between "AWC's informal enthusiasm sparked by members' personal commitments" compared with "the ACM-W's more institutionally embedded momentum." While pointing to a host of successful efforts in education, conferences, outreach, and institutional mobilization—not least is the mammoth annual Grace Hopper Celebration of Women in Computing—Bix remains aware of signal shortcomings. Although the computer-engineer themed "Barbie" doll initially seemed a positive model for girls, the accompanying 2013 book, *I Can Be a Computer Engineer*, offered a sorry lesson in tone-deaf sexism. With gender equity in computing remaining elusive, it is clear that much remains to be done; Bix's chapter is an essential primer for comprehending the past decades' varied lessons in advocacy efforts for women in computing.

The ACM and the international world are the subjects of the next two chapters, each demonstrating that the ACM was stretched by as well as enriched through the extension of professional relationships to countries beyond the United States. Scott Campbell's chapter 8 deals with an impossibly close neighbor. It provides a capsule history of the several varieties of computer professionalism in both Canada and the United States, dealing with the rise and fall of different professional organizations—some more scientific in their bent, others more attuned to practical bread-and-butter issues. ACM was quick off the mark in Canada, holding its first-ever international meeting in Toronto in 1952. In the 1950s both ACM and

Copyrighted material

the DPMA, noted above, tried to drum up attractive activities, organizational structure, and membership appropriate for a professional organization. In the 1960s these two organizations contended over accreditation in Canada, which is intimately linked to education and the workforce. Canadian Kelly Gotlieb served as editor-in-chief of *Communications of the ACM* during 1962–1965 and of *Journal of the ACM* during 1966–1968, while many Canadians worked closely with the international minded IFIP. Yet attractive programs are not the whole story. Specifically *Canadian* organizations also sprouted up in the Computing and Data Processing Society of Canada, later evolving into the Canadian Information Processing Society or CIPS. CIPS proclaimed itself as the national "industry spokesman," which as Campbell notes is a key attribute of professionalization. The "Canadian Problem," expressed through conflicts about postage stamps and publications and national identity, led Canadians ultimately to split from the ACM-rival DPMA and found their own national "data processing" society. It so happened, all the same, that accreditation standards north and south of the U.S.-Canada border converged neatly. Toronto was and remains one of the larger ACM chapters in North America.

IFIP appears also as a key institutional actor in Ksenia Tatarchenko's chapter 9. It is built around meticulous study of a single episode: the visit of noted Soviet computer scientist Andrei Ershov to the United States in 1965, at the height of the Cold War. Ershov intended to attend the IFIP congress in New York, but these plans were derailed by a delay in his U.S. visa. He did have notable visits with (George Forsythe's) department of computer science at Stanford University and another lecture to the Los Angeles chapter of ACM, in the very heart of the government-dominated aerospace industry. As a counterweight to the traditional image of the Cold War, where implacable opposition reigned between the two superpowers, Tatarchenko provides a striking image of Ershov—chalk in hand (Figure 9.1)—giving a lecture at RAND, the archetypical Cold War think tank! Tatarchenko's archival sources make abundantly clear that deeply personal connections and intense intellectual relationships crossed the Cold War political divide. "It is really true that you are the Novosibirsk branch of the ACM," observed Edward Feigenbaum. Her chapter suggests that these international ties were crucial means for advancing professionalism: demonstrating that one had esteemed colleagues on the other side of the "iron curtain" was a means for showing that the young profession really mattered. Ershov came to the U.S. while Feigenbaum, Forsythe, and John McCarthy, among others, made reciprocal visits to the Soviet Union. After his visit to Stanford, Ershov in a letter related to Forsythe his distinct pleasure in the "singular commonality of our problems and research interests." Her case study joins

Copyrighted material

other recent work showing that computing was one of the bridges in the Cold War, yet to be fully acknowledged.[22]

Arvid Nelsen's chapter 10 forms a neat complement with Toland's chapter 6 on social issues. He investigates a set of practical community-oriented educational efforts designed to realize the promise of computing as a lever of advancement for society. Like Toland's account, Nelsen deals with ACM members advocating awareness of and engagement with these non-college educational programs. While it was relatively easy for Stanford to adopt an elite stance (prioritizing externally funded research over undergraduate education, for example), it proved a great challenge to bring computing education to "disadvantaged" communities in poor neighborhoods. The archival papers of Berkeley and other ACM members help Nelsen piece together an elusive yet gripping story. His institutional focus is the ACM Special Interest Committee on the Social Implications of Computing, or SIC$^2$ (which traces its antecedents to a Washington, DC group organized in the mid 1960s around the automation question) and the short-lived Committee on Computing and the Disadvantaged, active in the late 1960s and early 1970s. The 18 local educational-training programs that Nelsen identifies were located across the U.S., with concentrations on the East coast and in California. Even with its wariness about being overly political, ACM did devote attention to these efforts with prominent articles appearing in *Communications of the ACM* (its "ACM News" sections are a prime source) and attention by local ACM chapters as well as the special interest group on Computer Personnel Research, or SIGCPR. Ultimately, however, the ACM Committee on Computing and the Disadvantaged, despite organizing an impressive number of activities, lost support and funding (around 1974), and the ACM lost a mechanism to remain connected to these educational and outreach efforts. Like Bix's chapter 7 on women, Nelsen's on educational outreach can serve as a cautionary tale and primer for today's enthusiasts of efforts for computing for all.

The book's third section, "Expanding Research Frontiers," profiles three areas of research activity where ACM members and ACM itself shaped notable advances in computing. Jacob Gaboury's chapter 11 forms an intellectual prequel to the notable activities of ACM SIGGRAPH. His story follows the notable career of David C. Evans, who founded the University of Utah's storied computer science department

---

[22] S. Donig. 2010. Appropriating American technology in the 1960s: Cold War politics and the GDR computer industry. *IEEE Ann. Hist. Comput.*, 32(2): 32–45; K. Tatarchenko. 2010. Cold War origins of the International Federation for Information Processing. *IEEE Ann. Hist. Comput.*, 32(2): 46–57; N. Lewis. 2016. Peering through the curtain: Soviet computing through the eyes of Western experts. *IEEE Ann. Hist. Comput.*, 38(1): 34–47.

Copyrighted material

and shaped it into a world-leading center for early computer graphics. While at the Bendix Corporation in southern California during the 1950s, one of his colleagues was Harry Huskey who had worked with Alan Turing on the Pilot ACE project. Evans, after a brief stint at University of California–Berkeley, returned to his native Utah in 1965–1966 to found a computer science division within the university's School of Engineering. Funding for a nascent research program in "Graphical Man/Machine Communication" was arranged by IPTO's Ivan Sutherland who, after a brief time at Harvard, also came to the University of Utah. Evans, with a brand-new department, was free to focus both research and education on computer graphics; students chose well-defined "problems" in computer graphics, received top-level support in addressing them, and often enough did research that had field-creating results. Among the Utah graphics alumni are such figures as Ed Catmull, Nolan Bushnell, Alan Kay, John Warnock, and others well-known to graphics experts. Evans and Sutherland formed a company that helped create the computer graphics industry. Gaboury's focus is on Utah and yet he acknowledges the "professionalization of computer graphics through the ACM and its SIGGRAPH special interest group." After the notable 1973 SIGGRAPH conference connected academic research more closely to industry, computer graphics research spread outward from Utah—initially to Xerox PARC, NASA's Jet Propulsion Laboratory, the New York Institute for Technology—often through the movement of Utah graduates.

Rebecca Slayton's chapter 12 takes an up-close focus on ACM-specific activities, including conferences and publications, in creating a new sub-field of computing. Computer security was a concern of at least *six* different ACM SIGs which each pursued its own framing or conceptualization of computer security, leading, she suggests, to an early balkanization of the field. Privacy was a leading concern of SIGCAS (Computers and Society) as Janet Toland's chapter 6 in this volume also documents. Protection of computer resources was a focus of researchers in SIGOPS (Operating Systems). Databases were a joint concern for SIGMOD, SIGIR, and SIGBDP (respectively, Management of Data, Information Retrieval, and Business Data Processing). In the early 1980s SIGSAC (Security, Audit, and Control) emphasized managerial solutions to the problem of computer crime. Some integration occurred in the 1980s when networking and privacy debates encouraged a search for commonality in security framings. The IEEE sponsored the annual Symposium on Security and Privacy, beginning in the 1980s, while the ACM sponsored the Computers, Freedom, and Privacy conferences beginning in the 1990s. ACM members also played important roles in the wide-ranging discussion of "critical infrastructures." In all, computer security (to slightly expand a quote) "appears to be a very hard problem to solve."

Copyrighted material

Inna Kouper's chapter 13 explores how ACM members responded to changing research frontiers and emerging technologies. Her chapter is focused on a twin transformation: on the technical level, research focused on hypertext to research on the World Wide Web, which was one instantiation of hypertext; and on the organizational level, on the evolution of SIGLINK to SIGWEB, signally a shift to fully embrace the Web and to expansively conceptualize hypertext. Kouper begins with an intellectual history of "hypertext" as it emerged in the visions of Vannevar Bush, Douglas Engelbart, and Theodore Nelson. ACM took a prominent role in sponsoring the landmark *Hypertext '87* workshop at Chapel Hill, North Carolina. At the time, compelling instances of hypertext—"Xerox's NoteCards, Owl's Guide, and especially Apple's HyperCard" as keynoter Andries van Dam put it—existed in a distant, parallel universe with internetworking. Hypertext standards were developed by the Dexter group which authored a three-layer model. Hypertext was at the time a coherent field of research. Notably, a paper proposal by none other than Tim Berners-Lee to the 1991 hypertext conference was outright *rejected*: "as being weak on research and . . . not [containing] references to the relevant work in the field." Rapid growth in Berners-Lee's World Wide Web in the 1990s coexisted with SIGLINK's persisting "vision of hypertext research" of "standalone systems that allowed [users] to explore architectural, aesthetic, and design complexities" as well as exciting work in non-web-based "digital libraries." Eventually SIGLINK, focused on the hypertext domain, broadened its agenda in 1998 to become SIGWEB. Alas, the WWW conferences organized by Berners-Lee's organization soaked up much of the energy and resources of the burgeoning web community, even leading to the odd misperception among some that the WWW was "the first hypertext." Kouper's close study of papers, panels, awards, and conference themes, played out alongside the institutional transformation of an ACM SIG, provides a rewarding perspective on the history of a vital aspect of contemporary computing.

This book is the first to explore the history of ACM as a community of computing professionals. In several ways, it reflects the state-of-the-art in historical research on ACM. Historians are exploring the rich archival records that ACM has itself made available to researchers and supplementing these traditional sources with oral histories of ACM awardees, officers, and members. A great deal of ACM's activities takes place not at its New York headquarters but in the decentralized work of its active special interest groups and local chapters. The ACM History Committee encourages ACM SIGs and makes widely available (through workshops

Copyrighted material

and outreach) the full range of professional practices and standards in history. Twenty SIGs have significant web-, wiki-, or conference-based historical activity.[23]

Notable recent activities include SIGOPS's SOSP History Day (2015), SIGCSE's ongoing Computing Educators Oral History Project, and the roundup of the ACM History Committee activities.[24]

For readers searching for additional information on the "big picture" of ACM there are several places to turn to for further exploration. *Communications of the ACM* has published a handsome number of articles on ACM history, as the collected bibliography at the end of this volume amply attests. Bernadette Longo's biography, *Edmund Berkeley and the Social Responsibility of Computer Professionals*,[25] profiles one of the organization's founders and one of its most colorful and opinionated members. And there is now a full set of online profiles of the winners of the A. M. Turing Prize as well as a full-scale oral history project seeking to create professionally done interviews with all living awardees.[26] It so happens that *29* Turing laureates figure someplace in this present volume—listed here in order of receiving the award: Alan Perlis, Maurice Wilkes, Richard Hamming, Marvin Minsky, John McCarthy, Edsger Dijkstra, Charlie Bachman, Donald Knuth, Herb Simon, Allen Newell, John Backus, Edgar Codd, Dennis Ritchie, Ken Thompson, Niklaus Wirth, Ivan Sutherland, Fernando Corbató, Butler Lampson, Juris Hartmanis, Edward Feigenbaum, Douglas Engelbart, Fred Brooks, Alan Kay, Peter Naur, Frances Allen, Barbara Liskov, Shafi Goldwasser, and, most recently, Whitfield Diffie and Martin Hellman. In addition, modern computing depends implicitly on many others, such as Vinton Cerf and Robert Kahn (2004), for the fundamental internet concept as well as the three eponymous winners for the RSA algorithm (2002).

Research in history, as in other fields, sometimes generates tidy answers but just as often leads to further questions to be asked and problems to be addressed. In assembling this volume, the authors have made significant progress in assembling chapters in ACM's storied history. But of course there is much more to be done.

---

[23] See "SIG History Activities" at http://history.acm.org/content.php?do=sighistory.

[24] See "SOSP History Day," October 4, 2015 (Monterey, CA) at http://sigops.org/sosp/sosp15/history; Computing Educators Oral History Project at http://ceohp.org; and the ACM History Committee website and blog at http://history.acm.org and http://history.acm.org/content.php?do=blog.

[25] B. Longo. 2015. *Edmund Berkeley and the Social Responsibility of Computer Professionals*. Association for Computing Machinery and Morgan & Claypool, New York.

[26] See http://amturing.acm.org.

Copyrighted material

**Figure 1.5** **ACM bumper sticker proclaimed "Computing the Future" [n.d.].** (Image courtesy of the Charles Babbage Institute Archives, University of Minnesota Libraries)

ACM's leading efforts in defining "computer science" as a field of research and as a respected academic discipline clearly merits additional investigations. While this book outlines the important connections between ACM and Stanford, there remains complementary investigations of other pioneering computer science departments such as Purdue's and Carnegie Mellon's (and elsewhere) where ACM leaders were notably prominent and remain so through today.[27] ACM's special interest groups, or SIGs, have played impressive roles in cultivating the fields of programming languages, graphics, operating systems, computer-science education, and security (among many fields); numerous case studies need to be done to understand the creation, organization, operations, and consequences of the notable conference series that they've run. ACM's role in the wider world of politics, society, and economics also merits further consideration. If computing is creating the future—as the ACM once suggested with an informal bumper sticker (see Figure 1.5) and as Turing laureate Juris Hartmanis emphasized in an acclaimed 1992 National Research Council study[28]—it's a vital matter that a diverse range of professionals, educators, students, and citizens are engaged with its promise. We hope that future volumes in ACM history will take up some of these challenges and opportunities.

---

[27] R. L. Pyle. 2015. *First in the Field: Breaking Ground in Computer Science at Purdue University.* Purdue University Press, West Lafayette, IN; H. Crowther-Heyck. 2005. *Herbert A. Simon: The Bounds of Reason in Modern America.* Johns Hopkins University Press, Baltimore. Compare P. Mounier-Kuhn. 2012. Computer science in French universities: Early entrants and latecomers. *Inform. Cul. J. Hist.*, 47(4): 414–456; R. J. Carnota. 2015. The beginning of computer science in Argentina and the Calculus Institute, 1957–1970. *IEEE Ann. Hist. Comput.*, 37(4): 40–52.

[28] J. Hartmanis and H. Lin, editors. 1992. *Computing the Future: A Broader Agenda for Computer Science and Engineering.* National Academy Press, Washington, DC

Copyrighted material

2

# From Handmaiden to "Proper Intellectual Discipline": Creating a Scientific Identity for Computer Science in 1960s America

**Janet Abbate**

## 2.1 Introduction

In September 1966, ACM—a major professional society for computer scientists—published a letter to the membership from its new president, Anthony G. Oettinger. Oettinger lamented that U.S. government funding for computing was targeted to serve other disciplines, rather than computer scientists themselves. ACM members, he wrote, "have expressed deep concern over the fact that while very large amounts of money go into computation these days this does not in itself furnish money *for research and education in computer science*. They point out that most of the support for computation is now channeled to computation centers for their service activity." Having brought these concerns to the National Academy of Science and the National Science Foundation, Oettinger had found "numerous misconceptions about computer science firmly expressed within high councils of science and

Copyrighted material

government." At the top of his list of misconceptions was, "The computer is just a tool and not a proper intellectual discipline."[1]

Indeed, computer science as an academic field was still quite new in 1966—but it was growing fast. Electronic digital computers had been invented during World War II, and the postwar years saw escalating demand for computers in science, industry, and government.[2] A number of universities began offering computer science degrees in the early 1960s, and by 1963 at least 28 universities were offering bachelor's, master's, or doctoral programs in computer science.[3] Professional societies complemented these efforts to create a scientific infrastructure for computer science. In 1947, an informal group of computer experts formed the first professional society devoted to computing, the Association for Computing Machinery (ACM). As Edmund C. Berkeley, one of the chief founders of ACM, saw it, the new society would "advance the science, design, construction, and application of the new machinery for computing."[4] Other computer groups were formed within the American Institute of Electrical Engineers (1946) and Institute of Radio Engineers (1951); when these two societies merged in 1963 to form the IEEE, their computer groups were combined and eventually in 1970 they became the IEEE Computer Society. Professional societies supported scientific work by providing a forum for research results through academic conferences and peer-reviewed journals, by assisting in curriculum development, and—as we will see—by raising a collective voice for computer scientists in policy matters.[5]

---

[1] A. G. Oettinger. 1966. President's letter to the ACM membership. *Commun. ACM*, 9(12): 838, emphasis in original. Since ACM had no direct representation in NAS or NSF, Oettinger depended on the goodwill of Harvard physicist Harvey Brooks to act as an intermediary. "By unanimous vote of the [ACM] Council at its meeting of August 29th, I was instructed to forward a letter expressing these concerns to the Committee on Science and Public Policy of the National Academy of Sciences (COSPUP). Dr. Harvey Brooks, Chairman of COSPUP, very kindly initiated discussion of our problem both within the National Academy of Sciences and within the National Science Board of the National Science Foundation."

[2] For overviews see P. E. Ceruzzi. 1998. *A History of Modern Computing*. MIT Press, Cambridge, MA; M. Campbell-Kelly and W. Aspray. 1996. *Computer: A History of the Information Machine*. Basic Books, New York.

[3] W. F. Atchison and J. W. Hamblen. 1964. Status of computer sciences curricula in colleges and universities. *Commun. ACM*, 7(4): 226–227.

[4] Quoted in B. Longo. 2015. *Edmund Berkeley and the Social Responsibility of Computer Professionals*, p. 80. Association for Computing Machinery and Morgan & Claypool Publishers, New York.

[5] See chapter 5 in this volume by Dziallas and Fincher for the role of ACM and IEEE CS in curriculum development and how these efforts helped define the identity of computer science.

Copyrighted material

The National Science Foundation, the primary source of government funding for scientific research in America, did not initially recognize computer science as a scientific field on a par with other disciplines.[6] NSF had been founded in 1950 with a mandate to support "basic scientific research," and many of its staff viewed computer science—with its emphasis on building machines—as a mere handmaiden for the "real" sciences.[7] NSF operations were structured largely along disciplinary lines, with units (programs, sections, or divisions) for mathematics, astronomy, physics, chemistry, biology, and engineering sciences.[8] The organizational control wielded by the established disciplines, as well as NSF's emphasis on basic research, put the emerging field of computer science at a disadvantage. In this context, the notion of computing as a "science" and the appropriateness of NSF funding for computing researchers were both contested. Yet fairly quickly, computer scientists were able to change their status at NSF from handmaiden to "proper intellectual discipline." Computing research began to obtain dedicated funding through a series of NSF units, starting with the Office of Computing Activities in 1967 and culminating with the creation of the directorate for Computer and Information Science and Engineering (CISE) in 1986.

This chapter addresses the historical question of how computer scientists were able to gain intellectual respectability and an institutional foothold at NSF in such a short time. The answer reveals much about the politics of scientific identity and

---

[6] While computer scientists also got significant funding from military agencies, that tended to be for mission-oriented projects; NSF provided the majority of funding for basic computer science until 1982; see National Research Council. 1999. *Funding a Revolution: Government Support for Computing Research*, Fig. 3.6. National Academies Press, Washington, DC. (Available at http://wayback.archive-it.org/5902/20160210155450, http://www.nsf.gov/statistics/nsf99346/). Moreover, funding from military agencies such as DARPA did not depend on computer science being seen as a *scientific* discipline. As NSF program manager Frederick W. Weingarten pointed out, "The generals knew that computers were very important. They weren't hung up about whether it was a basic discipline or not" (F. Weingarten. 1990. Interview by William Aspray, Charles Babbage Institute, OH 212, p. 18. Available at http://purl.umn.edu/107706). Thus, in terms of shaping a disciplinary identity, computer scientists' interactions with NSF were more important than their interactions with the military. For DARPA's role in computing, see J. Abbate. 1999. *Inventing the Internet*, Chap. 1. MIT Press, Cambridge, MA; and A. L. Norberg and J. E. O'Neill. 1996. *Transforming Computer Technology: Information Processing for the Pentagon, 1962–1986*. Johns Hopkins University Press, Baltimore.

[7] *National Science Foundation Act of 1950*, PL-507, 81st Congress, 2nd Session.

[8] Although "engineering sciences" might seem like a natural home for computer science research, computer scientists were not very involved with NSF's engineering division in the 1960s and early 1970s. Aside from a brief flurry of activity in information theory in FY 1961, the Engineering division did not focus on computing topics until it started a program for computer engineering

Copyrighted material

the multiple meanings of *discipline* in the context of 1960s America. In framing this investigation I assume that "science" is a socially constructed rather than natural category, and that the term "discipline" has had multiple meanings that are historically specific. I draw on Thomas Gieryn's theory of "boundary-work," which holds that one strategy of those who wish to establish a new scientific field is to rhetorically redraw boundaries so that an area formerly considered "outside" of science now appears to be "inside." Such maneuvers are possible because definitions of science change over time and encompass a diverse and sometimes contradictory set of characteristics, allowing actors to selectively emphasize those aspects that best support their own particular struggle. As Gieryn observes, "What science becomes, the borders and territories it assumes, the landmarks that give it meaning depend on exigencies of the moment—who is struggling for credibility, what stakes are at risk in front of which audiences, at what institutional arena?"[9]

In the case of computer science, a range of cultural assumptions about both the nature of science and the nature of computing were available in public discourse of the 1960s. As I have described elsewhere, computer scientists who publicly asserted the scientific status of their field tended to draw on three distinct meanings of science (sometimes in combination).[10] The first line of argument defined science as the study of natural phenomena. While including computing machines in this category may seem counterintuitive, advocates argued that computer scientists actually study *information*, which occurs naturally in the world (in DNA, for example). This framing appears in the ACM's 1965 Preliminary Recommendations for a computer science curriculum, which offered this definition: "Computer science is concerned with *information* in much the same sense that physics is concerned with energy. . . . As physics uses energy transforming devices, computer science uses information transforming devices."[11] A second approach defined science as the derivation of ab-

---

in 1979 (see NSF Annual Reports for these years). Dian Olson Belanger's history of engineering at NSF argues that the engineers had initially tried to emulate scientists, but by the 1960s had become disillusioned with this strategy and "moved to assert that engineering deserved recognition on its own merits" See D. O. Belanger. 1998. *Enabling American Innovation: Engineering and the National Science Foundation*, p. 75. Purdue University Press, West Lafayette, IN.

[9] T. F. Gieryn. 1999. *Cultural Boundaries of Science: Credibility on the Line*, x–xi. University of Chicago, Chicago.

[10] J. Abbate. October 2013. Is computer science "science"? A half-century debate. Keynote talk, 2nd International Conference on History and Philosophy of Computing, Paris.

[11] ACM Curriculum Committee on Computer Science. 1965. An undergraduate program in computer science: Preliminary recommendations. *Commun. ACM*, 8(9): 543–552, on 544.

Copyrighted material

stract ideas from concrete phenomena. ACM president George Forsythe provides an apt example in a 1965 letter to the membership, where he posed the rhetorical question, “If the computer is just a tool, why should there be an international society devoted to it?” His answer was that the “essence” of computer science was not machinery but abstract theory. Forsythe wrote, “As with Mathematics or Statistics or Physics, so also with Computer Science: It is more efficient to abstract the essence of the subject from its applications and study the essence itself.”[12] A third strand of argument presented the experimental method as the defining characteristic of science. Herbert Simon took this position in his 1969 book *The Sciences of the Artificial*, in which he argued that computer systems had become so complex that their behavior could not always be predicted and must be discovered empirically.[13]

I see no reason to doubt that computer scientists’ beliefs about science were sincerely held or that they aligned with practitioners’ personal experiences as researchers. But statements about the scientific nature of computing could also be made to serve a more public and strategic purpose. I argue that in the institutional arena of US science policy and for an audience of both computing insiders and skeptical outsiders, computer scientists performed two forms of boundary-work, using mutually reinforcing discursive and institutional strategies. Institutionally, they pursued strategies to include representatives of computer science within bodies that made or influenced science funding decisions. Discursively, they argued in speeches and publications that computer science met what they claimed were the crucial criteria for scientific identity.

Looking at discipline formation as boundary-work reveals an outward-facing meaning of *discipline,* as a claim to status and turf; this sense is apparent in many comments by contemporary actors. But some of the debates over the scientific status of computing were conducted among computer scientists themselves and not aimed at policy makers. What was at stake in these internal discussions? I suggest that the desire for a disciplinary identity had a second meaning for the computer science community: as a search for intellectual coherence and consensus on practice. Computer science was a new field whose definition and scope varied widely between institutions. Indeed, not all computing practitioners agreed that their field was a science at all, some preferring to identify as engineers or business

---

[12] G. E. Forsythe. 1965. President’s letter to the ACM membership: Why ACM?” *Commun. ACM*, 8(3): 143–144. As Joseph November describes in this volume, Forsythe saw computer science as natural, experimental, *and* abstract, combining the three definitions of science discussed here.

[13] H. A. Simon. 1969. *The Sciences of the Artificial*, p. 21. MIT Press, Cambridge, MA.

Copyrighted material

professionals. By declaring their field to be a science, computer scientists sought both to erase the boundary separating them from the scientific establishment and to reassure themselves that their field had the potential to become—like other sciences—a "proper" intellectually coherent discipline. Gregory Good describes this aspirational meaning of *discipline* as "a combined social contract and conceptual and practical framework," a promissory note that keeps colleagues loosely aligned as they build their field through a process of "assembly."[14] In Good's model, "Scientists assemble disciplines using many elements: phenomena, methods, instruments, theories, analytical techniques, and institutional tools such as journals, government bureaus, and university positions."[15] These elements can provide a basis for community and collaboration while the discipline's conceptual framework is gradually filled in. Viewing discipline-building as assembly helps explain how the institutional development of computer science in the 1960s (professional societies, journals, university positions, and an NSF office) could precede any firm consensus on the field's content.

This chapter begins with an assessment of the status of computing at NSF at the beginning of the 1960s. I then examine examples of institutional and discursive boundary-work that ACM leaders and other computer scientists performed as a means to improve their standing and resources. Finally, I describe the successful outcome of these efforts, as embodied in the new NSF Office of Computing Activities. The evidence for my historical argument is drawn from a variety of published and archival primary sources. Contemporary policy reports by the National Academy of Science and the President's Science Advisory Committee supply important examples of how public discourse about science became linked to material resources. NSF Annual Reports provide details on program activities as well as the public-facing view of the rationale for these, while the *Organizational Development of the National Science Foundation* is invaluable for tracking year-by-year realignments that shifted with the political winds. ACM publications, notably its flagship *Communications of the ACM*, convey the views and policies of ACM leaders as well as feedback from ordinary ACM members in the form of letters. These contemporary historical sources are complemented by retrospective oral histories from the CBI collection, which provide insight into the motives and perceptions of some key participants.

---

[14] G. A. Good. 2000. The assembly of geophysics: Scientific disciplines as frameworks of consensus. *Stud. Hist. Philoso. Sci. (Part B)*, 31(3): 259–292, on 259, 261.

[15] Ibid., 259.

Copyrighted material

## 2.2 The Status of Computing at NSF in the Early 1960s

The National Science Foundation made occasional grants for computer science research in the 1950s, but most of its computing funds went to university computer centers to support research in other scientific fields. NSF's earliest efforts to fund university computer centers were overseen by mathematician and computer pioneer John von Neumann. Von Neumann had hands-on experience with university computing, having initiated and overseen the creation of a computer at Princeton's Institute for Advanced Study from 1946–1951. In February 1955 von Neumann was asked to head NSF's ad hoc Advisory Panel on University Computing Facilities, which was able to persuade the National Science Board to include computers in the funding category for large-scale scientific facilities.[16] The first five grants were made in 1956, and in 1959 NSF created a dedicated budget for computer facilities.

Kent Curtis, who took charge of the computer centers program in 1967, recalled that the program helped establish computer science, "because it provided the first experimental equipment which the computer scientists had to use in research in their own field."[17] But the emphasis on funding computer centers as a service function also reinforced the subordinate status of computing at NSF. This is reflected in the NSF annual reports for fiscal years 1959–1962, which mention computers only in terms of facilities, summer programming courses for college professors, "supplementary teaching aids," and "mechanical aids" for information handling.[18] The 1962 annual report reiterates this service role: "Computers are proving of ever-increasing value as *tools* for research and training in virtually every imaginable scientific field"; nowhere does the report mention computer science.[19] Funding for computer science *research* was channeled through NSF's Mathematical Sciences program. This unit had well-established disciplinary norms, and by mathematicians' standards, proposals for computer science projects often looked

---

[16] T. A. Keenan. 1990. Interview by W. Aspray, Charles Babbage Institute, OH 217, p. 7. Available at http://purl.umn.edu/107401; W. Aspray and B. O. Williams. 1994. Arming American scientists: NSF and the provision of scientific computing facilities for universities, 1950–1973. *IEEE Ann. Hist. Comput.*, 16(4): 61. Aspray and Williams note that von Neumann was awarded NSF's first computing grant in 1954, for a conference exploring applications of computers to meteorology.

[17] K. K. Curtis. 1987. Interview by J. Minker, pp. 3–4. Available at http://purl.umn.edu/107238.

[18] Quotations are from *Annual Report of the National Science Foundation, FY 1960*, pp. 103, 118. NSF, Washington, DC.

[19] *Annual Report of the National Science Foundation, FY 1962*, p. 60, emphasis added. NSF, Washington, DC.

Copyrighted material

less polished than competing mathematical proposals.[20] Thus, computer science was positioned at NSF as either a handmaiden to other sciences (in the facilities program) or a second-class form of mathematics (in the research program).

A series of ACM presidents bemoaned the status quo at NSF in speeches and letters to their membership. Alan J. Perlis, who helped found the computer science department at Carnegie Tech (now Carnegie Mellon), argued in 1963 that dedicated funding for computer science was "urgently needed . . . to develop computing without the constraints now attached to it as a purely derivatory activity."[21] His successor, George Forsythe of Stanford, cynically noted, "I see no practical alternative today to our *assuming the role* of one of the mathematical sciences" as a strategy to obtain federal funding.[22] But as Oettinger pointed out in 1966, it was unlikely that computer science would be fairly represented "in a body dominated by mathematicians, . . . whose outlook on computing, if at all understanding, is from the point of view of a computer *user*."[23] To change the status of computer science at NSF, ACM leaders and other concerned computer scientists would employ both discursive and organizational strategies to position themselves within the mainstream of science.

## 2.3 Organizational Boundary-work: Getting a Seat at the Table

ACM leaders strove to get their voices heard in the organizations that controlled funding decisions. One strategy was to join and become active in professional organizations at the bottom of the science policy pyramid—where membership was relatively easy to obtain—and use these channels to gain influence at higher levels. Thus, in the spring of 1961, ACM joined the Conference Board of Mathematical Sciences, an umbrella organization that included the American Mathematical Soci-

---

[20] See, e.g., W. R. Adrion, Interview by William Aspray, Charles Babbage Institute, OH 211, p. 12. Available at http://purl.umn.edu/104300; F. Weingarten. 1990. Interview by William Aspray, Charles Babbage Institute, OH 212, p. 9. Available at http://purl.umn.edu/107706. Joseph November makes a similar point in chapter 3 of this volume, where he describes Forsythe's concern that computer science's "priorities were sufficiently different from mathematics that the latter field had no mechanisms to reward or protect those who did meaningful work to develop algorithms and computational techniques" and that computer science would therefore "be viewed by mathematicians as the lowest form of mathematics."

[21] A. J. Perlis. 1963. Computation's development critical to our society. *Commun. ACM*, 6(10): 642.

[22] G. E. Forsythe. 1966. President's letter to the ACM membership. *Commun. ACM*, 9(4): 244, emphasis added.

[23] A. G. Oettinger. 1966. President's letter to the ACM membership. *Commun. ACM*, 9(12): 839, emphasis added.

Copyrighted material

ety, the Association for Symbolic Logic, the Institute of Mathematical Statistics, the Mathematical Association of America, the National Council of Teachers of Mathematics, and the Society for Industrial and Applied Mathematics.[24] ACM did not join CBMS simply because it was a "natural" intellectual home; indeed, many ACM members did not identify with the "mathematical sciences" and questioned why ACM should pay even a modest membership fee to belong.[25] But one of CBMS's functions was to provide advice to government agencies on mathematical issues, including funding needs. Urging skeptical members to continue support for CBMS in 1965, Forsythe as ACM president warned, "It is not easy for a new discipline to break into the Establishment! . . . Since research and development and education in computing are vastly expensive, there is no alternative but to maintain as good relations as possible with the only organization having the kind of funds needed—the US Government."[26] In this usage, a *discipline* functions as a political unit engaged in a boundary-drawing struggle.

Forsythe argued that ACM's investment of dues money, and its officers' time attending meetings, was beginning to pay off. The National Academy of Sciences' Division of Mathematical Sciences was in the process of forming a Committee on Support of Research in the Mathematical Sciences (COSRIMS) as part of a series of discipline-specific reports on research funding needs requested by the NAS Committee on Science and Public Policy. As Forsythe explained, "Such committees will have an important effect on the allocation of national funds (and hence effort) in the coming decade. It is extremely important to computing and to the nation that our field be adequately represented on the committees. But such representation doesn't occur automatically." He pointed out that COSRIMS members had been chosen on the advice of CBMS, and that ACM's membership in CBMS had shifted the boundary of "mathematical sciences" to the advantage of computer scientists: "It is largely due to ACM's membership [in CBMS] . . . that the Conference Board is beginning to think of 'the mathematical sciences, including computing' instead of just 'mathematics' when important decisions are made."[27] A few months later, Forsythe's successor, Oettinger, reminded ACM members of the political importance of COSRIMS as he asked the society to contribute needed funds for the COSRIMS report. NSF had provided initial support for COSRIMS, but this amount

---

[24] H. D. Huskey. 1961. From the President of ACM. *Commun. ACM*, 4(4): 1.

[25] While ACM's allegiance with CBMS would prove politically effective, many computer scientists remained uncomfortable identifying as mathematicians, and ACM eventually left CBMS in 1985.

[26] G. E. Forsythe. 1965. Official ACM. *Commun. ACM*, 8(7): 424.

[27] Ibid.

Copyrighted material

was not sufficient to complete and publish the three-volume report, and professional societies were being asked to make up the difference. Oettinger urged ACM to pay this money in order to secure long-term research funding: “I must emphasize that, at this time, COSRIMS is our only major official point of contact with the powerful policy-recommending machinery of the National Academy of Sciences-National Research Council.”[28] The ACM Council agreed to contribute $1000, and the report was published in 1968 and well received.[29]

In this example, organizational and discursive boundary-work went hand in hand: by changing their institutional location, computer scientists were able to shift the boundary of “mathematical sciences” to include computer science, and the COSRIMS report helped shift the public identity of computer science from a purely service role to a research field. The COSRIMS report powerfully supported the computer scientists’ ambitions, describing investment in computer science as a matter of national urgency. While the report noted that computer science “shares with mathematics the role of handmaiden to all of science and technology,” it emphasized that computer science should also be supported as its own research area.[30] Harvey Brooks, chair of the Committee on Science and Public Policy, warned in his letter presenting the report, “The development of computer science only as a by-product of the application of computer techniques in other fields often results in failure to develop a distinctive body of theory and technique in computer science in

---

[28] A. G. Oettinger. 1966. President’s letter to the ACM membership. *Commun. ACM*, 9(12): 839.

[29] The first two volumes of the report were policy recommendations for research and education in mathematical sciences. The third volume was a collection of essays on contemporary mathematical topics, described by a *New York Times* reviewer as “probably the best book on higher mathematics ever written” (H. Schwartz. 1969. “Review of *The Mathematical Sciences: A Collection of Essays*. Edited by the National Research Council’s Committee on Support of Research in the Mathematical Sciences (Cosrims) with the Collaboration of George A. W. Boehm. Cambridge, MIT Press,” *New York Times*, April 20, 1969, p. 39). A review of the first two volumes in *Science* observed more cynically that the report “seeks to build a case that will encourage federal agencies to lay on the dollars with a generous hand,” but also noted that “preparation of the report provided an unusual opportunity for people from all branches of pure and applied mathematics to learn more of one another’s problems and opportunities” and quoted Brooks’s call to support computer science “in its own right” (L. J. Carter. 1968. More funds urged for science’s “leading wedge”. Review of *The Mathematical Sciences: A Collection of Essays*, Cosrims and G. A. W. Boehm, editors. *Science* New Series 162(3856): 883, 885).

[30] Committee on Support of Research in the Mathematical Sciences COSRIMS. 1968. The mathematical sciences: A report, p. 94, quoting Newell. National Academy of Sciences, Washington, DC.

Copyrighted material

its own right."[31] The report recommended that "at the national level special priority be given to support of the expansion of research and graduate study in computer science," citing a "critical shortage of research leaders," and concluded that "every effort should be made to support as many good proposals as possible for research in computer science."[32] ACM leaders' strategy for making themselves heard in the top forums for science policy advice had succeeded.

ACM did not confine itself to joining established policy groups such as CBMS; it also created its own institutional mechanisms for monitoring and influencing government science policy and funding. For example, in November 1966 Oettinger persuaded the normally frugal ACM Executive Committee to hire a Washington correspondent, James P. Titus, to gather intelligence on federal activities and report to the membership in a monthly column in *Communications of the ACM*.[33] The topics covered a range of issues, from proposed computer privacy regulations to guidelines for software patents, but Titus frequently highlighted opportunities for, and obstacles to, government support for computer science.[34]

Another initiative was begun by ACM president Alan J. Perlis, who in April 1964 convened the whimsically named "Commission of Thoughtful Persons" to consider the society's future. The commission, which was evenly split between academic and industry practitioners, was "unanimous in recommending that the ACM adopt the policy that as much government support as is consonant with legal activities be obtained."[35] One strategy the group suggested was to offer government bodies free expert advice on computing issues, a service that would allow computer scientists to cultivate government connections while conveying the concerns of ACM members on policy issues. At the ACM Council meeting the following August, past-president Oettinger reported that he was forming such a group, called the Committee on Government Relations, which would be "a source of information about the professional

---

[31] Ibid., Brooks cover letter, Feb. 2, 1968, pp. iii–iv.

[32] Committee on Support of Research in the Mathematical Sciences COSRIMS. 1968. The mathematical sciences: A report, pp. 17, 205–206. National Academy of Sciences, Washington, DC.

[33] A. G. Oettinger. 1966. President's letter to the ACM membership. *Commun. ACM*, 9(12): 712.

[34] See J. P. Titus. 1967. ARPA: A visible means of support (Washington Commentary). *Commun. ACM*, 10(8): 519–520; and J. P. Titus. 1967. The patchwork nature of government computing support (Washington Commentary). *Commun. ACM*, 10(9): 589–592.

[35] G. E. Forsythe. 1964. President's letter to the ACM membership. *Commun. ACM*, 7(8): 508. Besides Perlis the initial members included Paul Armer (RAND), John W. Carr III (University of Pennsylvania), ACM Executive Director Don Madden (IBM), and Walter Ramshaw (United Aircraft). The second meeting also included Dick Hamming of Bell Labs and Forsythe, the new ACM president.

Copyrighted material

viewpoint of ACM members for various Government agencies."[36] The committee's other three members were equally distinguished and well-placed: ACM President Alan Perlis; Paul Armer, Head of the Computer Science Department at the RAND Corporation; and J. C. R. Licklider, who was concluding a stint as director of ARPA's Information Processing Techniques Office to become a consultant for IBM. The group remained active for several years, and in December 1966 Oettinger reported that as a result of its efforts, "several agencies of the U.S. government have now expressed an interest in direct communication with the computing field." The group was enlarged and renamed the Government Advisory Committee, with a mission "to insure direct and effective participation by the computer science and engineering community in discussions that may affect its future." The committee would not only answer government inquiries but also take the initiative to "speak out freely, forcefully, and intelligently on issues affecting computing and build a bridge between the government and the computer field at large."[37]

As these examples illustrate, ACM's organizational boundary-work created multiple platforms for influencing policy and reconstructing the public image of computer science. What was said in those public forums is the subject of the next section.

## 2.4 Discursive Boundary-Work: Establishing a Scientific Identity

Computer scientists knew that they had to be recognized *as* scientists in order to be eligible for NSF funding and participation in U.S. science advisory bodies. For many researchers, science also had a normative intellectual status to which they aspired. Asserting their membership in a *scientific* discipline functioned as a claim to intellectual rigor and prestige and provided meaning to their personal goals and experiences as researchers. But since computer science did not fit the canonical model of science—having neither a codified body of theory nor a domain of the natural world associated with it—its advocates were forced to explain and justify their claims to scientific identity. Throughout the 1960s, therefore, computer scientists took every opportunity to assert, both among themselves and to their scientific peers, that they were truly scientists. This discourse reveals both computer scientists' tendency to equate *discipline* with *science* and a striking variety in their criteria for what makes a science.

---

[36] Ibid., 633–636.

[37] A. G. Oettinger. 1966. President's letter to the ACM membership. *Commun. ACM*, 9(12): 839. The new committee was headed by former ACM President Harry D. Huskey.

Copyrighted material

The effort to establish scientific credibility began with the name of their field, which was a subject of ongoing debate. As noted above, NSF had classified computing research under "applied mathematics" rather than assigning it an independent scientific identity. The term "computer sciences" was introduced in 1959 by Louis Fein, a consultant who wrote a widely read report on the state of computing education in U.S. universities.[38] While Fein never defined "science," his concept of "discipline," which was modeled on mathematics, drew on the contemporary understanding of science as a creative, theoretical, problem-driven enterprise. Fein's criteria for a discipline included: "Workers in the field do nonroutine intellectual work," "The field has sometimes been axiomatized," and "The field is open, i.e., problems are self-regenerating."[39] In 1963, Saul Gorn of the University of Pennsylvania used a similarly expansive label in an article promoting "The Computer and Information Sciences: A New Basic Discipline." While Gorn did not define what constituted a scientific discipline, his implied criteria included a unique subject domain—"the synthesis and analysis of mechanical languages and their processors"—and a theoretical dimension, which for him included "probabilistic information theory" as well as "a theory of mechanistic information and its processing."[40] Gorn's article was addressed to an audience of mathematicians, appearing in the *Society for Industrial and Applied Mathematics Review*, and seems aimed at persuading the scientific establishment that computer science was, in fact, a "basic science."

Other early leaders, such as operations research specialist Philip M. Morse, adopted the singular "computer science." Morse had founded MIT's Computation Center in 1956 with a donated IBM 704 and had persuaded NSF to support the facility as a shared computer center for forty colleges in the region.[41] He chaired the organizing committee for a 1960 conference of computer center directors, also funded by NSF, and proclaimed in his report on the conference that "Computer science is a new scientific field" and "Computer science is a discipline in its own right."

---

[38] L. Fein. 1959. The role of the university in computers, data processing, and related fields. *Commun. ACM*, 2(9): 11. For more on Fein see Joseph November's chapter 3 in this volume.

[39] Ibid.

[40] S. Gorn. 1963. The computer and information sciences: A new basic discipline. *SIAM Rev.*, 5(2): 150.

[41] D. Walden and T. Van Vleck, editors. 2011. *The Compatible Time Sharing System (1961–1973): Fiftieth Anniversary Commemorative Overview*. IEEE Computer Society, pp. 1, 6 and 4.21; F. J. Corbató. 1989. Interview by Arthur Norberg, Charles Babbage Institute, OH 162, p. 5. Available at http://purl.umn.edu/107230.

Copyrighted material

Morse invoked intellectual sophistication as the hallmark of a scientific discipline: "The problems associated with exploiting fully the potentialities of present and projected computers are difficult and intellectually challenging."[42] George Forsythe, who spearheaded the creation of a Division of Computer Science within the Stanford mathematics department in 1961, also favored the singular term, giving a lecture that year that described "a coherent body of technique, which I call computer science."[43]

ACM leaders also attempted to change the name of their society to sound more scientific. ACM stands for "Association for Computing Machinery," a phrase that irked many theoretical computer scientists because it invoked the computer as artifact rather than research field. In 1964, Perlis' Commission of Thoughtful Persons recommended, "A new name should be found which does not contain direct reference to equipment."[44] After some discussion, in 1965 ACM held a vote of its members on a proposal to change the society's name to "Association for Computing and Information Sciences."[45] A substantial 63% of the votes favored the name change, but this fell just short of the two-thirds required, and the issue was never again brought to a formal vote.[46]

Perhaps the most explicit defense of computer science directed toward the skepticism of mainstream scientists was a 1967 letter to the editor of *Science* by three leading researchers: Allen Newell, Alan Perlis, and Herbert Simon. These three had successfully established a department of computer science at the Carnegie Institute of Technology in 1965, but evidently they still did not feel accepted by their scientific peers. Their letter begins bluntly, "Professors of computer science are often asked: 'Is there such a thing as computer science, and if there is, what is it?"' The first answer they offer is cheekily brief: "Wherever there are phenomena, there can be a science to describe and explain those phenomena. . . . Ergo, computer science is the study of computers."[47] But sensing that this "simple answer"

---

[42] P. M. Morse. 1960. Report on a conference of university computing center directors (June 2–4 1960). *Commun. ACM*, 3(10): 520–521.

[43] Forsythe's lecture, "Educational implications of the computer revolution," is quoted in D. E. Knuth. 1972. George Forsythe and the development of computer science. *Commun. ACM*, 15(8): 722.

[44] A. J. Perlis. 1964. Report of the commission of thoughtful persons to the ACM council, 24 April 1964. *Commun. ACM*, 7(8): 508.

[45] G. E. Forsythe. 1965. Official ACM. *Commun. ACM*, 8(7): 424.

[46] G. E. Forsythe. 1965. "President's letter to the ACM membership," *Commun. ACM*, 8(9).

[47] A. Newell, A. Perlis, and H. A. Simon. 1967. Computer science. *Science*, 157(3795): 1373–1374.

Copyrighted material

might not sway skeptics, they go on to argue for the uniqueness and complexity of their field of study. Addressing the hypothetical objection that the computer is only a tool—"Computers, like thermometers, are instruments, not phenomena"—the authors respond, "The computer is such a novel and complex instrument that its behavior is subsumed under no other science. . . . Hence, the computer is not just an instrument but a phenomenon as well, requiring description and explanation." They conclude by emphasizing that computer scientists share a scientific ethos characterized by "passion" for research and "confidence that intelligent, persistent curiosity will yield interesting and perhaps useful knowledge."[48] As these examples show, advocates of computer science ranged widely in their notions of the essence of science and selectively emphasized whichever features best fit their own practice.

In addition to arguing directly for their field's scientific worthiness, ACM leaders pursued a discursive strategy of generating and disseminating information to back up their claims. Quantitative data on topics such as "scientific manpower" provided powerful ammunition for science policy struggles in 1960s America.[49] By creating facts (statements about reality that were accepted as true and objective) on such topics, computer scientists could shape the context in which federal policy makers would consider their requests for funding. ACM President Oettinger was forthright about the political function of quantitative studies, arguing in 1966, "Our [funding] case and that of the universities would be much stronger if reliable data about the present state of the computer industry and its future material and manpower needs were available."[50] He proposed that ACM conduct a survey to create such data, and ACM's Education Committee responded by preparing a report on the needs of education in computer science. ACM also helped fund a survey of industry demand for computer personnel.[51] These data helped articulate a national need for computer science funding by making visible a numerical gap between the projected supply and demand.

The most visible and influential data-based interventions on behalf of computer science were two publications known as the Rosser and Pierce Reports. The Rosser Report was begun by the NRC in 1962. It was nominally requested by NSF Director Alan T. Waterman, but the instigator was Philip Morse of MIT, who was seeking increased funding from NSF for a bigger computer to support MIT's work on time

---

[48] Ibid., 1374.

[49] J. Lucena. 2005. *Defending the Nation: U.S. Policymaking to Create Scientists and Engineers from Sputnik to the 'War against Terrorism.'* University Press of America, Lanham, MD.

[50] A. G. Oettinger. 1966. On ACM's responsibility. *Commun. ACM*, 9(4): 246.

[51] A. G. Oettinger. 1966. President's letter to the ACM membership. *Commun. ACM*, 9(12): 839.

Copyrighted material

sharing. Arthur Grad, who oversaw NSF's computer facilities grants, advised Morse that his request was too large and that he would need to make a policy case for it.[52] Morse had been elected to the NAS in 1955, and he used his influence there to request an NRC report on computing needs—demonstrating once again how institutional access could enable discursive boundary-work. The resulting report, published in 1966, was titled *Digital Computer Needs in Universities and Colleges,* but it became known as the Rosser Report after lead author J. Barkley Rosser, a theoretical computer scientist at the University of Wisconsin.[53]

The Rosser Report proclaimed, "Computer science is rapidly assuming the position of an established discipline" and argued that computer centers were needed both to provide services for other sciences and to support computer science itself.[54] Numerous statistics were offered to justify these arguments: the number of university computing centers had increased tenfold from 1957 to 1964; "campus expenditures for computing were doubling every two years"; "35,000 computer staff positions were being created each year in the United States."[55] The report recommended doubling federal support for campus computing, to support both computer science research and training for industry computer jobs.[56] Reaction to the report was mixed, and it did not immediately result in a new funding vehicle

---

[52] Grad came to the mathematical sciences section of NSF in 1959 and administered the computer facilities grants. He recalled the genesis of the Rosser Report as follows: "That all started with Phil Morse at MIT. They needed a bigger computer. They estimated they would need about ten million dollars. And I told them, well, there wasn't much I could do about it since my entire budget was only five. And I suggested to him that probably the best thing he could do was to have a National Academy study done pointing out the need for more money for computers. So the Academy duly appointed the committee to make those studies. . . . But it all started from Phil Morse's need for a big computer" (A. Grad. 1990. Interview by William Aspray, Charles Babbage Institute, OH 216, p. 10. Available at http://purl.umn.edu/107338). Thomas A. Keenan recalled that "the National Academy of Sciences decided to (mainly at P. Morse's instigation) to set up a committee to study the uses of computers in colleges and universities" and that Morse had asked Rosser and Keenan to serve on the committee. Keenan later came to NSF as a program officer for OCA in 1969.

[53] J. B. Rosser. Chair, National Research Council Committee on Uses of Computers. 1966. Digital computer needs in universities and colleges: A report. National Academy of Sciences/National Research Council, Washington, DC.

[54] Ibid., 123, 16.

[55] Quoted in W. Aspray and B. O. Williams. 1994. Arming American scientists: NSF and the provision of scientific computing facilities for universities, 1950–1973. *IEEE Ann. Hist. Comput.*, 16(4): 64.

[56] Rosser, Digital computer needs in universities and colleges, 1–2.

Copyrighted material

for computer science research, but it did make computer science more visible to policy makers and paved the way for a more influential study.[57]

The Rosser Report may have been weakened by a perception that it was solely concerned with serving the computing community, rather than the broader public interest.[58] But contemporary political culture offered computer science advocates an alternative discursive framing. Lyndon Johnson's 1964 State of the Union speech had urged the creation of a "Great Society" that would eliminate poverty, racial injustice, and urban problems, and the first item on Johnson's national agenda was "a program in education to ensure every American child the fullest development of his mind and skills."[59] Among those who heeded Johnson's call was John R. Pierce, an electrical engineer and research director at Bell Telephone Laboratories who was serving on the President's Science Advisory Committee. In 1966 Pierce recruited a panel of experts from universities and industry research labs to investigate national needs and opportunities in educational computing. The committee's findings were published in February 1967 as *Computers in Higher Education*, commonly known as the Pierce Report.

The Pierce Report embedded computer researchers' aspirations to be accepted as scientists in a new narrative that positioned funding for computer science as a path to national wellbeing. Its first pages boldly declared, "Adequate support of computing as a part of education is essential for a rapid and full realization of the social and economic benefits of computing."[60] The report recommended that all universities should provide educational computing and that the federal government should share the cost. This plea for funding was woven into a tale of enlightened progress that framed access to computers as almost a civil right:

> Happily, at some fortunate and forward looking colleges and universities the educational use of computers is widespread and effective. But this does not apply to

---

[57] Aspray and Williams discount the influence of the Rosser Report (Arming American scientists, 66). Gupta, however, writes that the Rosser Report "had considerable impact on funding for computers in higher education" (G. K. Gupta. 2007. Computer science curriculum developments in the 1960s. *IEEE Ann. Hist. Comput.*, 29(2): 52).

[58] Rosser's main co-author on the report, Thomas A. Keenan of the University of Rochester, felt the report's focus on funding may have been counter-productive: "I think the Bureau of the Budget at the time thought it sounded . . . self-serving" (T. A. Keenan. 1990. Interview by W. Aspray, Charles Babbage Institute, OH 217, p. 4. Available at http://purl.umn.edu/107401).

[59] L. B. Johnson. 1964. State of the Union addresses by Lyndon B. Johnson. In J. Manis, editor. *An Electronic Classics Series Publication*, p. 17. Pennsylvania State University, Hazleton, PA.

[60] J. R. Pierce. 1967. *Computers in Higher Education: Report of the President's Science Advisory Committee*, p. 4. US GPO, Washington, DC.

Copyrighted material

> the majority, where computing facilities are often absent or inadequate. . . . Can this deficit be remedied, so that no American need have second-rate education in this respect?[61]

Having made a case for funding *educational* computing, the report deployed this as a rationale for supporting computer *science*. After all, trained faculty would be needed to teach computing to all of these new students:

> The whole success of educational computing . . . depends on expanded education and research in computer sciences. This education requires a good faculty and access to very good computing facilities for both course work and research. We recommend that the Federal Government expand its support of both research and education in computer sciences.[62]

This recommendation, with its repeated emphasis on "research" and "science," went well beyond the basic teacher training that might have sufficed to support computer literacy courses in schools. The report also broadened the importance of academic computer science beyond its contribution to education—the ostensible focus of the study—by arguing, "In order to provide the computer experts who . . . are so vital to our national defense, to increasing our productivity, and to improving our standard of living, we must have excellent computer science departments at a number of schools."[63] At the same time, the report tried to distance computer science from its traditional, low-status service role, stating, "The computer sciences faculty should not be burdened with the administration of a computer center."[64] By invoking "research" and intellectual "excellence," the report discursively placed computer experts among the company of autonomous scientists, rather than mere technical or educational assistants. Computers were a vital policy tool, but could only fulfill that promise as the center of a new scientific discipline.

## 2.5 Success: The Creation of NSF's Office of Computing Activities

The Pierce Report's persuasive rhetoric—and its authors' access to the President—brought results.[65] Immediately after its publication, President Johnson told the nation in a February 1968 speech that "One educational resource holds exciting

---

[61] Ibid., 2.

[62] Ibid., 5.

[63] Ibid., 22.

[64] Ibid., 41.

[65] Milt Rose, who was responsible for the computing programs in the NSF Mathematical Sciences Division in the 1960s, argued that it was the Pierce Report and Pierce's contacts with the

Copyrighted material

promise for America's classrooms: the electronic computer."[66] He then directed the National Science Foundation to work with the Office of Education to establish a new program for computers in education. How to implement this directive was not clear, since neither agency had a plan for educational computing, but the initiative quickly defaulted to NSF as the agency with more computer expertise.[67] NSF Director Leland Haworth established an Office of Computing Activities in July 1967 and appointed Milton E. Rose, who had been leading NSF's Mathematical Sciences section, as its first head. Commenting on the newly created OCA, Johnson predicted with satisfaction, "The day is not far when these exciting new machines will be contributing to the education of our people."[68] But OCA was not solely focused on applying computers to education: following the Pierce Report's logic to its conclusion, OCA was also given a mandate for "providing Federal leadership" in the research side of computing.[69] This fact went unremarked by the politicians, but would prove to be a turning point for computer science at NSF.

Computer science's newly raised status came with additional funding: OCA's initial budget was $22 million, a 73% increase from the $12.7 million allocated for computer activities in education and research in the previous year.[70] Perhaps as

---

president's science adviser and other high-level government officials that provided the political influence to open a computer office and embark on a major computer education initiative in the foundation (W. Aspray and B. O. Williams. 1994. Arming American scientists: NSF and the provision of scientific computing facilities for universities, 1950–1973. *IEEE Ann. Hist. Comput.*, 16(4): 68, note 15).

[66] L. B. Johnson. 1967. Special Message to Congress: "Education and Health in America," February 28, 1967, in *Lyndon B. Johnson: 1967: Containing the Public Messages, Speeches, and Statements of the President*, p. 251. Office of the Federal Register, Washington, DC.

[67] NSF staff involved in the startup of OCA recall that the motivation for placing the educational computing activity within NSF, rather than Education, was partly a matter of expertise, but also economic. Rose recalled in an interview, "The Department of Education was considered as the proper agency for this work on computers in education, but the Bureau of the Budget and presidential adviser Joseph Califano believed that the DOE would not have the technical command of the subject to use prudently and effectively the substantial funding" (quoted in Aspray and Williams, Arming American scientists, 62). Kent Curtis, who joined NSF in the summer of 1967, believed that NSF was given responsibility for the education program because the government was under financial pressure due to the Vietnam war, and having NSF repurpose its existing computer science program was seen as the cheapest option (K. Curtis. 1987. Interview by J. Minker, pp. 22–23).

[68] L. B. Johnson. 1968. Message to the Congress transmitting annual report of the National Science Foundation, March 20. In *Lyndon B. Johnson: 1968–1969: Containing the Public Messages, Speeches, and Statements of the President*. Office of the Federal Register, Washington, DC.

Copyrighted material

significant was the new location of computer science within the overall NSF organization: rather than being under one of the research directorates, OCA reported directly to the NSF Director. This positioning may have reflected the lingering view that computer science was primarily a form of scientific infrastructure, rather than a discipline; but it also fulfilled the hopes of ACM activists by bringing computer science out from under the shadow of mathematics, where its status as a research field had always been in question.[71] It also kept computer science out of the Engineering Division, whose Advisory Committee had been lobbying since 1965 to be put in charge of computing activities.[72] Thus, OCA's organizational location gave the nascent computer science field some breathing room to develop independently of its two parent disciplines.

OCA's internal organization reflected the multiple, still-evolving identity of computer science within NSF. It had three sections that corresponded to its three functions of supporting computers as scientific infrastructure; computers as a tool for education; and computer science as a discipline.[73] These multiple identities sometimes caused tension, reflecting the uneasy position computer science still occupied between handmaiden and autonomous science; but once an independent office for computing had been established, its managers could, and did, look for opportunities to shift the balance of activities toward research.

---

69. National Science Foundation. 1984. Organizational development of the National Science Foundation (NSF Handbook Number 1), p. 71. NSF, Washington, DC.

70. National Science Foundation. 1968. *Annual Report of the National Science Foundation, FY1968*, Table 2, pp. 8–9. NSF, Washington, DC.

71. Weingarten recalled that OCA was put under the Director "because the program was infrastructural" (F. Weingarten. 1990. Interview by William Aspray, Charles Babbage Institute, OH 212, p. 18. Available at http://purl.umn.edu/107706).

72. Belanger notes, "In 1965 the NSF Advisory Committee for Engineering had strongly recommended that the Engineering Division 'have an important part, or even the major role, in the broad area of computers'." She implies that the placement of the Office of Computing Activities under the NSF Director, and its emphasis on education rather than engineering, was a disappointment for NSF's engineers (D. O. Belanger. 1998. *Enabling American Innovation: Engineering and the National Science Foundation*, p. 171. Purdue University Press, West Lafayette, IN.)

73. Most of OCA's initial activities and managers were simply transferred from the Mathematical Sciences section. The Institutional Computing Services section took over the role of funding universities to purchase computers as a tool for scientists. Rose recruited Kent Curtis from Berkeley to run this section, and Glenn Ingram moved over from Mathematical Sciences to serve as Curtis's deputy. Arthur Melmed was brought from Mathematical Sciences to head the Special Projects Section, also known as Computer Innovations in Education, a new program created to fulfill OCA's educational mandate. OCA's third section, Computer Science Education, Research &

Copyrighted material

Having its own office at NSF also gave computer science greater visibility and legitimacy in the larger scientific community. For example, the National Academy of Science's 1968 report on *The Mathematical Sciences* interpreted the existence of "a separate Office of Computing Activities" at NSF as evidence that the government now viewed computer science as "an independent entity."[74] Computer science also became more visible in NSF's official publications. NSF's annual reports had barely mentioned computer science until 1968, when the establishment of OCA was discussed over several pages.[75] The 1969 report went further and proclaimed "the emergence of computer science as an academic discipline."[76] Once again, institutional and discursive boundary-work went hand in hand.

## 2.6 Conclusion

Through a variety of rhetorical and organizational strategies, computer science advocates were able, within the space of a decade, to claim scientific respectability for their field and gain control of their own funding stream within NSF. This was a remarkable coup for such a young field. ACM leaders advanced this cause in multiple ways: by working to get representation for computer scientists on scientific advisory bodies; by generating data to create supporting facts and narratives for their cause; and by making the case to their fellow scientists for the scientific legitimacy of their field.

The story of ACM and NSF also provides a window on the meanings of *science* and *discipline* in 1960s America. Because computer scientists were forced to articulate and defend their scientific identity, we are able to witness the variety—and glaring lack of consensus—among contemporary definitions of what makes a science. Computer professionals could argue that they were scientists by virtue of having a unique subject domain within the natural world, a high level of intellectual challenge, a body of theory or technique, or even a passion for research. We also see

---

Training, which focused on computer science as its own discipline, was led by Fred Weingarten. See National Science Foundation. 1984. Organizational development of the National Science Foundation (NSF Handbook Number 1), p. 71. NSF, Washington, DC.

[74] Committee on Support of Research in the Mathematical Sciences (COSRIMS). 1968. "The mathematical sciences: A report," p. 95.

[75] National Science Foundation. 1968. *Annual Report of the National Science Foundation, FY 1968*, p. 196. Only two prior annual reports had mentioned computer science as a research area: the 1964 report had an extended discussion of NSF-funded time sharing research (pp. 25–26), and the 1963 report made a passing reference to computer science research in "artificial intelligence, pattern recognition, etc." (p. 12).

[76] Ibid., FY 1969, p. 94.

Copyrighted material

that the notion of *discipline* can have multiple functions. In the pursuit of power, the scientific discipline as a political actor (represented in this case by ACM) can perform boundary-work to meet external challenges. In the pursuit of meaning, assembling a disciplinary identity can build a sense of value and shared purpose among practitioners of a new field.

For computer scientists in the early 1960s, recognition as a discipline could be seen as a criterion for membership in a powerful body, a measure of intellectual respectability, or a sign of internal agreement on goals and methods. The creation of the NSF Office of Computing Activities represented their success in achieving the first two forms of discipline by the late 1960s. Whether they would achieve the third remained an open question.

## 2.7 Acknowledgments

This work was supported through a 2012 Fellowship in ACM History. I am indebted to Bill Aspray for sharing research materials on the history of NSF's computing efforts and to the Charles Babbage Institute for access to its archival and oral history collections.

Copyrighted material

3

# George Forsythe and the Creation of Computer Science As We Know It

Joseph November

## 3.1 The Man Who Would Remove the "M" from the ACM

If George E. Forsythe (1917–1972) had only managed to persuade a couple hundred more ACM members to see things his way, you would be reading this chapter in a book published by the Association for Computing and Information Sciences instead of the Association for Computing *Machinery*. In 1965, while serving as president of the ACM, Forsythe had called on the organization to change its name to reflect the importance of "the art and science of representing and processing information" to the society. Ultimately, 63% of the roughly 6,000 ACM members who voted on the issue favored changing the organization's name, but the measure did not quite meet the two-thirds majority necessary to pass. "Personally, I am very sorry," wrote Forsythe, "This is the first proposed change that the membership has ever voted down."[1]

In the decades since 1965, the ACM continues to retain its original name, but Forsythe's work to reorient the organization towards promoting a "science of computing" left a powerful legacy, both for the ACM and for the broader field of computer science. Examining how Forsythe arrived at his vision for computer science and how his vision became so influential can give us much insight into why the field of computer science took on its distinct form—and the ACM's role in that formation process.

---

[1] G. E. Forsythe. 1965. President's letter. *Commun. ACM*, 8(9): 541.

Copyrighted material

If you are not a computer scientist, you probably have not heard of George Forsythe; but within that field, especially the parts of it connected to the influential Stanford Computer Science Department, which he founded in 1965, he is a towering historical figure. Stanford computer scientist Donald Knuth, the creator of TeX and the author of *The Art of Computer Programming*, summarized Forsythe's status: "It is generally agreed that he, more than any other man, is responsible for the rapid development of computer science in the world's colleges and universities."[2] Another Stanford computer scientist, John Herriot, shared such sentiment, arguing, "that the fact that nationally as well as at this university, computer science came to be recognized as an area of study that was of importance in itself, was due to the efforts and the vision of George Forsythe more than any other person. . . . "[3]

Indeed, Forsythe's name adorns some of the most crisply articulate and widely cited early writings about the nature and aims of academic computer science, many of which were published in *Communications of the ACM* when (and immediately after) he served as president of the organization from 1964–1966. And in his honor, the first home of Stanford University's Computer Science Department, Forsythe Hall, was named for him. The building still bears his name even decades after the department he founded outgrew it.

More than 40 years after Forsythe died, tales of his exemplary brilliance, generosity, and vision abound. His place in academic genealogies and bibliographies is, if anything, becoming more prominent as those whom he trained (and those with whom he worked) achieve intellectual and institutional success.[4] Decades on, strong feelings remain about Forsythe's efforts to establish computer science as an academic discipline, and he is often cast as a martyr-like figure whose vision of what a computer science department ought to do was at once widely emulated but also left unfulfilled when Forsythe died before he could fully implement it. Although many of the accounts of Forsythe's life show him as industrious and academically productive, few delve deeply enough into either his published or private writings to

---

[2] D. E. Knuth. 1972. George Forsythe and the development of computer science. *Commun. ACM*, 15(8): 721–727.

[3] Oral history interview with J. Herriot, Charles Babbage Institute, OH 21. Available at http://purl.umn.edu/107356.

[4] See N. H. F. Beebe. 2012. A Bibliography of publications of George Elmer Forsythe. Available at ftp://131.254.254.45/pub/netlib/bibnet/authors/f/forsythe-george-elmer.pdf. Cleve Moler's 1997 "Tree of Forsythe Students," shows the institutional placements of Forsythe's graduate students. Available at http://infolab.stanford.edu/pub/voy/museum/forsythetree.html.

Copyrighted material

learn about the world out of which computer science emerged and how Forsythe's often frustrated efforts to overcome the challenges presented by that world grew into many of the specific characteristics of computer science today.

Initially, my research centered on the questions of how, specifically, Forsythe built and promoted his vision for computer science and why that vision was apparently so well received. Ironically, the vast amount of available archival and other less traditional documentary material have muddied the issue to the point that a neat synthesis is impossible in the space of a book chapter. My hope had been that, by drawing from the collected documents related to Forsythe's work, I would be able to lift the shroud of mythology surrounding the man, a mythology that has grown out of narratives centered on his unexpected death in 1972 at the age of 55 and at the height of his career. However, in the process of trying to moving beyond pronunciations like "one might almost regard him as the Martin Luther of the Computer Reformation!"[5] I discovered there was another shroud, one formed from the presence of *too much* documentary material.

As cultural anthropologist Diana E. Forsythe (1947–1997), the daughter of George and Alexandra (Sandra) Forsythe, explained, her father "documented his every professional act."[6] Much of this writing has found its way into the Stanford University Archives, and particularly (but not exclusively) into the George and Alexandra Forsythe Papers (1936–1979), which includes thousands of letters, memos, notes, drafts, and diaries (as well as formally published material) and which spans 40 linear feet. She added that the experience of going through the collection was "painful," and that the "letters and memos in his files confront me again and again with reminders of the plans and dreams curtailed by his sudden death."[7]

Beyond showing what George Forsythe hoped computer science could be, his notes include candid reflections on the many institutional, personal, and intellectual challenges faced by early promoters of computer science. Given the volume and complexity of Forsythe's writing, a book-length scholarly biography would be necessary to enable readers to most productively make use of his perspective on the

---

[5] Knuth, George Forsythe and the development of computer science, 721–727.

[6] D. E. Forsythe. 2001. George and Sandra's daughter, Robert's girl: Doing ethnographic research in my parents' field. In D. J. Hess, editor. *Studying Those Who Study Us: An Anthropologist in the World of Artificial Intelligence*, pp. 183–196, on 191. Stanford University Press, Stanford.

[7] Ibid., 191.

Copyrighted material

formation of computer science.[8] The good news is that by glancing at selections of the rich material written by—or about—Forsythe, we can nevertheless gain access to some otherwise unreachable foundations of computer science.

## 3.2 Forsythe Before CS: From Mathematics to Meteorology to Computing

When Forsythe began his career at Stanford in 1957 there were very few people using general-purpose electronic digital computers, and there was no such thing as a computer science department.[9] Forsythe was at the time a formally trained mathematician (by Jacob Tamarkin at Brown University) specializing in numerical analysis who had a strong interest in electronic computers. Although he had initially been on track to pursue a career in mathematics, his course changed during World War II when he joined the US Army Air Force (USAAF), where he worked to develop mathematical solutions to meteorological problems. There he encountered the work of British mathematician and meteorologist Lewis Fry Richardson, who had predicted that someday machines capable of computing differential equations could be used to forecast weather. Besides seeing computing machines as being useful for predicting the weather, Forsythe began to believe by the late 1940s that computers could broadly transform science and industry.[10]

During the 1950s, Forsythe worked as an applied mathematician for the Institute for Numerical Analysis at the National Bureau of Standards. The Institute's computer, the Standards Western Automatic Computer (SWAC), interested him and his mathematician wife, Sandra, enough that they followed its construction closely and learned how to program it. By the mid-1950s, the Forsythes were publishing algorithms they had developed to get SWAC to efficiently work with algebraic equations. This work captured the attention of Stanford mathematician John G. Herriot, a longtime friend and fellow Tamarkin student, who recruited him to Stanford's Mathematics Department to help train graduate students to use com-

---

[8] Speaking about the Forsythe papers, "Knuth suggested that Forsythe's notes could form the core of an intellectual history of computer science through 1972." See P. Davis. 1998. Remembering George Forsythe. *SIAM News*. Available at http://www.siam.org/news/news.php?id=805.

[9] Technically, Forsythe began his *second* career at Stanford in 1957; he had briefly worked there as a mathematics instructor in 1941, before he joined the USAAF as a meteorology researcher and instructor.

[10] J. November. 2007. George Elmer Forsythe. In *The New Dictionary of Scientific Biography*. Charles Scribner's Sons, New York.

Copyrighted material

puters in their research.[11] By 1960, Forsythe had co-authored a textbook, titled *Finite-Difference Methods for Partial Differential Equations*, which contained much discussion of computer use, but it cast computers as *tools* to be used in service of mathematics rather than objects worthy of study in their own right.[12]

In 1961, as Forsythe sought to create ever-more efficient algorithms for carrying out numerical analysis work on computers, he came to the conclusion that specialized training in computer operations would be necessary to provide "the general-purpose mental tools" mathematicians needed to break new theoretical ground. With Herriot's help, he founded the Division of Computer Science within the Mathematics Department. The curriculum he established there included not only traditional mathematical training in numerical analysis and automata theory, but also courses in programming, data processing, game theory, information theory, information retrieval, recursive function theory, and computer linguistics.[13]

## 3.3 Hard Lessons on the Road to Computer Science

Stanford University's Computer Science Department has been remarkably productive. Its faculty and graduates have consistently been leaders of the field, and it is widely celebrated as a model of how a good department ought to work. However, the department's continued existence as a separate department (first in the School of Humanities and Sciences then in the School of Engineering)—five decades after Forsythe established it in 1965—is a sign that some of his most ambitious plans have gone unrealized. To understand how such a successful department's longevity contradicts Forsythe's long-term hopes for computer science, one must become familiar with the environment in which Forsythe was working in the 1960s and with his intellectual agenda as well as the challenges he faced. In short, the entity known as a "computer science department" was but a step—albeit a necessary one—toward Forsythe's plans to make working with and studying electronic digital computers truly productive. Seeing why that is the case requires a trip back to when the notion of investing heavily in computing was not popular at Stanford.

---

[11] Ibid.

[12] G. Forsythe and W. Wasow. 1960. *Finite-Difference Methods for Partial Differential Equations*. Wiley, New York.

[13] G. E. Forsythe. June 25, 1965. Stanford University's program in computer science. Technical Report CS26, Computer Science Department, Stanford University. Available at http://i.stanford.edu/pub/cstr/reports/cs/tr/65/26/CS-TR-65-26.pdf).

Copyrighted material

Forsythe's first effort, carried out in 1962, to generate broad institutional support—and raise enough funds to acquire an IBM 7090—for the Division of Computer Science ended in failure, and its lessons would shape his many successes in the years to come. What Forsythe had hoped to do was to expand Stanford's Computation Center, which was run by the Division of Computer Science, by promoting it as an accessible place where scientists could go to find ways to employ computers in their research. Forsythe believed that the scientists who would benefit the most from using computers were biologists, and he canvassed the Stanford campus in an effort to convince them to use computers. He had been "firmly convinced that computing in life science research will ultimately be an absolutely essential ingredient for the rapid growth of fruitful investigations."[14] Further, even though his areas of expertise, numerical analysis and meteorology, were far removed from the concerns of biologists, he believed he could create a productive rapport with biologists because his father, Warren Ellsworth Forsythe, had been a prominent medical research professor at the University of Michigan.[15]

At first the effort went well, with Forsythe garnering commitments from diverse biology and medical researchers, including pharmacologist Keith Killiam, neurologists Kao Liang Chow and Frank Morrell, psychologist Karl Pribram, and biostatistician Lincoln Moses.[16] In the coming decades, all of these scientists would become known as pioneering computer users in their respective fields. In the early 1960s, though, their enthusiasm alone was not enough to impress the entities from which Forsythe sought support, namely his own department, Stanford's administration, and the National Institutes of Health (NIH).

On Stanford's campus, Forsythe's effort to gain funding for the Computation Center could not overcome the skepticism about computing that had arisen in reaction to a 1956 study conducted by Louis Fein, a non-academic computing consultant, on the prospects of building a computer science department or school at the university. In short, Fein recommended establishing "computer-related science" as a "supra-discipline," one that he called "Synnoetics," under the aegis of which would be united "Information Theory . . . Automation . . . Cybernetics . . . Operations Research . . . Theory of Games . . . Data Processing . . . Artificial Intel-

---

[14] G. E. Forsythe. December 4, 1962. Memorandum. "NIH." Stanford University, ACME Collection, SC 236 Box 1 Folder 2.

[15] "Warren Ellsworth Forsythe," Faculty History Project, University of Michigan. Available at http://www.lib.umich.edu/faculty-history/faculty/warren-ellsworth-forsythe.

[16] J. November. 2012. *Biomedical Computing*: *Digitizing Life in the United States*, p. 230. Johns Hopkins, Baltimore.

Copyrighted material

ligence . . . Management Science . . . Bionics. . . . "[17] Fein recalled that his report was viewed by faculty and administrators as too ambitious. John Herriot, for instance, according to Fein, violently dismissed it: "[Herriot] would get up and shout, 'What you want is pie in the sky and you can't have pie in the sky!' And that was the reception I got."[18] Fein also recalled that even the newly hired George Forsythe's attitude towards the report was "equivocal; he could go for it or not go for it."[19] From Stanford Provost Fred Terman's perspective, "the faculty were always coming to him with one hot project after another and this was just another hot project as far as he was concerned. He couldn't see that this was any better than any other hot projects circulating in the air."[20] Although Terman is so often cast as a great promoter of computer science, there was a time back in the early 1960s when it was clear that he had yet to be convinced of its exceptional promise.

Within the mathematics department, Forsythe's colleagues, including Herriot (who had personally recruited him), distanced themselves from efforts to expand the Computation Center, on the grounds that the activity was at once too ambitious and too pedestrian. The department's chair, David Gilbarg, as Fein remembered in 1979, "thought computing was like plumbing. We don't have plumbers studying within the university and what does computing have to do with intellect? I am exaggerating of course. He was murder on it and he may still be to this day."[21] Forsythe agonized over this unexpected criticism and his colleagues' apparent fears about how getting involved in computing could lead to losses of status and autonomy. Renowned technology writer Pamela McCorduck, who in the 1970s examined Forsythe's private writing on the matter, succinctly described the situation: "You really see that laid out in Forsythe's papers—the tearing of hair over what the relationships between the department and the center should be—I mean it is just a non-issue nowadays but then it seemed to be very difficult."[22] These kinds of arguments against academic computing were not lost on Forsythe and he later addressed them publically and robustly.

---

[17] Fein reproduced much of his 1956 report in L. Fein. June 1961. The computer-related sciences (Synnoetics) at a university in the year 1975. *Am. Sci.*, 49(2): 149–168.

[18] Oral history interview with Louis Fein. Charles Babbage Institute, OH 15, p. 11. Available at http://purl.umn.edu/107284.

[19] Ibid.

[20] Oral history interview with Albert H. Bowker. Charles Babbage Institute, OH 6. Available at http://purl.umn.edu/107140.

[21] Oral history interview with Louis Fein, 11.

[22] Pamela McCorduck in Louis Fein oral history interview, 12.

Copyrighted material

More damaging in the short term to Forsythe than being lumped together with Fein was the negative reaction from the federal agency from which he had sought external funding, the NIH. In his 1962 grant proposal to the NIH, Forsythe claimed that after the facility had been tested by the first wave of researchers it would reach enough new users that it could establish "the foundation for a future general medical data processing center embracing all aspects of medical problems." Forsythe further argued that the facility would "establish an intellectual bridge between the computation facility and the medical center, bringing into focus the problems of quantification peculiar to biological systems and also serve to build a reservoir of talent who would be familiar with modern computational facilities on the one hand and the kinds of problems facing medicine on the other." Toward these ends he had requested $1 million to be spent over five years.[23]

Over the last six months of 1962, NIH reviewers sent Forsythe mixed signals. Some had been swayed, but most were "unwilling to game so much money in equipment and personnel on an 'unproved' approach to the general problem of processing medical data." One sticking point was Forsythe's plan for the Center to carry out mass conversions of analog data into digital data. Up to that point, no such system ever had been successfully implemented in a biomedical environment. Further, in other contexts similar (and more costly) efforts had proved unproductive, such as the large-volume analog-to-digital conversion projects undertaken by McDonnell Aircraft Corporation and the (NIH-funded) UCLA biomedical computation center.[24] Sensing that a rejection was looming, Forsythe scrambled to submit an amendment to his proposal, which he called a "counter-proposal," and which included provisions to train Stanford life scientists in enough mathematics to prepare their analog data for conversion to digital. He also offered to use some of his funding to give the Stanford scientists access to existing analog-to-digital conversion facilities, such as those McDonnell and UCLA.[25]

In November 1962, the NIH formally rejected all forms of Forsythe's proposal. For Forsythe, the "complete refusal of an extremely well thought-of plan," was dispiriting, especially due to the tremendous effort he and others in the Division of Computer Science had put into the project: "I don't believe any proposal or report

---

[23] G. E. Forsythe and K. Killam. Medical Data Processing, Stanford University Archive, ACME Collection (SC 236), Box 1, Folder 2, p. 5.

[24] Ibid.

[25] G. E. Forsythe. November 29, 1962. Memorandum. "NIH Grant." Stanford University, ACME Collection, SC 236, Box 1 Folder 2. In this grant, Forsythe made no distinction between corporate and academic computer resources.

Copyrighted material

we have ever made was so much thought about, or involved coordination with so many people."[26] Initially, Forsythe had been "bitter about the matter," but as he pushed the NIH review committee for details, though, the agency's justification for turning down his grant proved revelatory.[27]

One basis for rejecting the grant, NIH reviewers noted, was that for all the intellectual merit of Forsythe's vision for biomedical researchers to use computers, there were few means to publish much of the work that would be done. Mathematics and engineering journals, let alone biology and medical research journals, did not have provisions for publishing computing techniques and algorithms, regardless of how useful they might prove. This would be problematic when it came to any of the NIH's future decisions related to continuing Forsythe's funding or that of researchers who had committed to computing, because the agency tended to measure success of grants in terms of the quantity and prestige of publications they generated.[28]

The other, more surprising, criticism coming from the NIH was that Forsythe had not done enough to promote computing as a productive intellectual activity to scientists. Forsythe had gathered a diverse group of researchers to use the Computation Center, but what about the larger population that had turned him down? Forsythe's initial reaction had been that he and other Division of Computer Science faculty and staff had already been proselytizing the good news of computing: "All of our staff members, the Director and Associate Director included, act both formally and informally as computing missionaries around the campus. Much of this missionary work is accomplished through personal contact. . . . "[29] However, as Thomas J. Kennedy, Jr., one of the members of the NIH committee that had rejected Forsythe's proposal, saw it, Forsythe had made a good intellectual case for computing but had failed to provide scientists with the resources to overcome the "'cultural' problems associated with introducing computer technology into the main stream of life science research."[30]

---

[26] Ibid.

[27] R. Wade Cole correspondence to Thomas J. Kennedy, Jr. (NIH), December 5, 1962, Stanford University, ACME Collection, SC 236, Box 1 Folder 2.

[28] G. E. Forsythe. April 30, 1966. Memorandum. "One More Pass at NIH." Stanford University, ACME Collection, SC 236, Box 1 Folder 2.

[29] Cole correspondence to Kennedy.

[30] Thomas J. Kennedy, Jr. correspondence to R. Wade Cole. December 18, 1962, Stanford University, ACME Collection, SC 236, Box 1 Folder 2.

Copyrighted material

After spending the first half of 1963 futilely attempting to find ways to get the NIH review committee to change its mind and attempting to get Stanford itself to pay for the bulk of the Computation Center, Forsythe gave up. In 1965, he turned over his correspondence and notes related to the failed grant to the next person who would try to establish a biomedical computing center at Stanford, geneticist (and Nobel Laureate) Joshua Lederberg.[31] By then, Forsythe had a much clearer vision of how to get universities to take computer science seriously, and he had acquired the means to widely share that vision.

## 3.4 Building a Home for Computer Science at Stanford

Following the setback with Computation Center, the Mathematics Department continued to hire mathematicians interested in working with computers. As the Division of Computer Science grew, Forsythe also began to take a more extroverted approach to establishing the importance of computing. On campus, he attempted to raise awareness of computers' potential by giving lively presentations to other departments and to university administrators. He also invested much time and energy into conversing with people one-on-one about computing, even engaging students at football games by having the Computation Center print up material used for a massive "card stunt" carried out by tens of thousands of cheering fans.[32] From Herriot's perspective, this activity was crucial in gathering enough support for Stanford to continue to invest in computing.

> Well, he gave talks. He talked to the administration and he convinced them about things that should be going on. But I suppose that more than anything, giving talks at various meetings and just talking to people, making the ideas become familiar to everybody . . . George was very good at his mission, explaining things to people and he had a lot of energy. We owe a lot to George.[33]

Forsythe's cheerleading for computing played particularly well to the audience of Stanford administrators. Although Provost Fred Terman and Dean Albert Bowker had balked at the 1956 Fein Report and although they did little to help Forsythe in

---

[31] For this reason, the papers related to Forsythe's attempt to establish a computing center for Stanford biomedical researchers were deposited by Lederberg into the *ACME* (Advanced Computer for Medical Research) Collection at Stanford's archive rather than the Forsythe papers.

[32] Oral history interview with Alexandra Forsythe. Charles Babbage Institute, OH 17, p. 16. Available at http://purl.umn.edu/107291.

[33] Oral history interview with John Herriot. Charles Babbage Institute, OH 21.

Copyrighted material

his effort to secure NIH funding, they both maintained a strong interest in continuing to make Stanford into an important center of activity related to computing. By 1963, Terman's plan (crafted jointly with Stanford's former President John Sterling) to reorient the university towards interdisciplinary and federally-supported research was well underway.[34] Already, Terman had overruled faculty concerns when he reoriented Stanford's medical school away from training and towards research. After dismissing, en masse, faculty who prioritized clinical training, Terman oversaw the hiring of top-notch interdisciplinary scientists, such as biochemist Arthur Kornberg and geneticist Joshua Lederberg, whose work could be applied to medical problems.[35] Under Terman's watch, the physical sciences and engineering saw such restructuring too, with faculty encouraged to collaborate across disciplines and also to raise money for their research by applying for federal grants or by finding ways to bring their work to the local university-cultivated marketplace—fostering Terman's reputation as "Father of Silicon Valley."[36] Throughout this effort, Terman increasingly saw computerization as a means to the end of transforming Stanford into a research powerhouse. Forsythe's hiring had been a consequence of Terman's emphasis on computing, as had been the hiring of other computing experts, most of whom had enough mathematics background to be housed without much controversy in the Mathematics Department.

Terman's desire to establish Stanford as a major center for computing began to create tension in the Mathematics Department when he sought to add to the Division of Computer Science faculty who were computing experts but who had little to do with professional mathematics. The tensions came to a head when Forsythe (backed strongly by Terman) attempted to recruit John McCarthy to Stanford.[37] Although McCarthy held a Ph.D. in mathematics, and although he was a champion of using mathematics to improve what would become computer science, the MIT computing expert was interested primarily in developing programming languages and exploring areas like timesharing and artificial intelligence. Other potential

---

[34] R. S. Lowen. 1997. *Creating the Cold War University: The Transformation of Stanford*, pp. 120–146, 157–177. University of California Press, Berkeley.

[35] E. J. Vettel. 2004. The protean nature of Stanford University's biological sciences 1946–1972. *Hist. Stud. Phys. Biol. Sci.*, 35(1): 95–113, on 96.

[36] C. S. Gillmor. 2014. *Fred Terman at Stanford: Building a Discipline, a University, and Silicon Valley*, p. 349. Stanford University Press, Stanford, CA.

[37] Oral history interview with Alexandra Forsythe. Charles Babbage Institute, OH 17.

Copyrighted material

hires, such as Edward Feigenbaum, who was trained at Carnegie Institute of Technology (now Carnegie Mellon University) by Herbert Simon and Allen Newell in areas that emphasized economics and psychology, had even less connection to mathematics.

In order to bring in McCarthy, Feigenbaum, and other computer experts whose presence Terman saw as crucial to bolster Stanford's computing, he formally established the Computer Science Department within the School of Humanities and Sciences in January 1965 and named Forsythe as department chair.[38] Some mathematicians, like Herriot, switched over to the new department. Like other Stanford departments, Computer Science had the authority to grant Ph.D. and M.S. degrees, but unlike traditional departments it did not have much in the way of undergraduate teaching obligations—the focus was on research and the training of graduate students.[39]

Terman had taken a bold step in creating the Computer Science Department, but he was able to convince Stanford faculty, administrators, trustees, and himself of the appropriateness of this move by grounding it in the justifications Forsythe had provided to him for establishing computer science as an academic discipline. Although Stanford was not the first university to found a computer science department—Carnegie, Cornell, Illinois, Michigan, North Carolina, Notre Dame, Pennsylvania, Penn State, Purdue, Stanford, Texas, Toronto, and Wisconsin all had (or were about to have) them—Forsythe's vision of computer science was articulated in a way that made its message resonate with a wide audience in the academic world and the (often overlapping) early world of computing.[40]

## 3.5 Forsythe and the Challenge of Defining "Computer Science"

When Forsythe was named department chair, he had already been elected president of the ACM from 1964–1966. As president (and later as an active ex-president) he used the *Communications of the ACM* as a pulpit to broadcast his vision of

---

[38] G. Wiederhold, Stanford Computer Science Department timeline. Available at http://www-cs.stanford.edu/about/department-timeline (accessed February 2016).

[39] G. E. Forsythe. 1967. A university's educational program in computer science. *Commun. ACM*, 10(1): 6. Much of Forsythe's language on graduate training found its way into another influential ACM report: W. F. Atchison, et al. March 1968. Curriculum 68: Recommendations for academic programs in computer science. *Commun. ACM*, 11(3): 151–197. See also chapter 1 in this volume by Misa and chapter 5 by Dziallas and Fincher.

[40] G. E. Forsythe. 1967. A university's educational program in computer science. *Commun. ACM*, 10(1): 4.

Copyrighted material

academic computing. (For the evolution of the ACM's definition of "computer science," particularly as articulated by the organization's presidents, see Janet Abbate's chapter 2 in this volume.) Much of what Forsythe said and wrote to Terman (and that Terman and others in Stanford's leadership found so convincing) wound up in a more accessible, polished form in *Communications* in the mid-late 1960s.

Tellingly, other widely shared visions of computer science written around the same time did not have nearly the resonance as those of Forsythe, despite sometimes appearing in publications with much larger readerships than *Communications.* Defining "computer science" in a letter published in *Science* in 1967, shortly after Forsythe published his own views in *Communications*, Allen Newell, Alan Perlis, and Herbert Simon (all renowned computing figures by then) emphasized computers themselves as objects of study. They wrote: "Phenomena breed sciences. There are computers. Ergo, computer science is the study of computers."[41] They dismissed notions that computers could be thought of as mere instruments ("like thermometers") or that the study of these devices could be "subsumed under any existing area of science." Making the case for centering computer science on the study of machinery, they assailed the notion that work with programs and algorithms should instead be the central activity of computer science. They even went so far as to issue a veiled criticism of Forsythe's popular 1965 attempt to rename the ACM as the Association for Computing and Information Sciences, writing: "Showing deeper insight than they are sometimes credited with, the founders of the chief professional organization for computer science named it the Association for Computing Machinery."[42] Forsythe's own writing, though limited to the much smaller readership of *Communications*, took a more nuanced, inclusive approach to the issue of which phenomena computer scientists might study, and unlike Newell, Perlis, and Simon's letter, discussed the pragmatic aspects of accessing those phenomena.

Besides being practically grounded and widely encompassing, Forsythe's vision eschewed much of the popular rhetoric that came up in discussions of computer science. For instance, his goals for computer science were far more modest that what *Time* magazine wrote in its 1965 cover story, "The Cybernated Generation,"

---

[41] A. Newell, A. J. Perlis, and H. A. Simon. 1967. Computer science. *Science*, 157(3795): 1373–1374. Forsythe would later describe this definition as "perhaps the tersest answer" to the question of "what is computer science?" See G. E. Forsythe. 1968. What to do till the computer scientist comes? *Am. Math. Month.*, 75(5): 454–462.

[42] Newell, et al., Computer science, 1373–1374. Perlis was president of the ACM 1962–1964 immediately before Forsythe.

Copyrighted material

about Louis Fein's vision for computer science serving as the means to free other scientists from the shackles of laborious calculating and organizational work:

> 'What the hell are we making these machines for,' says Dr. Louis Fein, a California computer consultant, 'if not to free people?' Many scientists hope that in time the computer will allow man to return to the Hellenic concept of leisure, in which the Greeks had time to cultivate their minds and improve their environment while slaves did all the labor. The slaves, in modern Hellenism, would be the computers.[43]

Fein, unlike Forsythe, also waxed eloquent about the potential of computers as means to bring more rigor to their users' thinking: "We are quite familiar with the role of computers as problem solvers, as calculators and as simulators, emulators, and imitators. But computers play their most significant role as a Socratic goad to analysis and problem formulation."[44] Such insights may elicit a knowing nod in agreement today, but in 1965 very few people, including perhaps the majority of decision-makers at universities, had had the bruising experience of attempting to transform their ideas into a functioning computer program.

These other visions may have reached much larger audiences than Forsythe's writing in *Communications*, but what Forsythe had to say about the scope and aims of computer sciences seems to have stuck. Fein's vision of "Synnoetics" never gained much traction, and for all the intellectual accomplishments of computer scientists at Carnegie (Perlis, Simon, Newell), their institutional models were not widely emulated. There was something appealing about Forsythe's approach to computer science at Stanford, an approach honed by carefully evaluating his many setbacks in earlier years. A close look at what Forsythe actually wrote to the ACM membership gives insight into that appeal.

Forsythe's best-known manifesto for computer science was published in January 1967 in *Communications of the ACM* at the conclusion of his two-year term as the organization's president. It was titled "A University's Educational Program in Com-

---

[43] "The Cybernated Generation." 1965. *Time*, 85(14): 84-91. Available at http://content.time.com/time/magazine/article/0,9171,941042,00.html. Alluding to Greek philosophy to explain computing is a running theme of the article and of Fein's own published work.

[44] L. Fein. 1963. Computer-oriented peace research. *AFIPS '63 (Fall) Proceedings of the November 12–14, 1963*, pp. 631–639, on 632. Tragically, considering his role in the origins of "computer science," Louis Fein's papers were discarded before they could be archived; see A. Leonard. June 10, 1999. Can history survive Silicon Valley? *Salon*. Available at http://www.salon.com/1999/06/10/stanford/.

Copyrighted material

puter Science," and was based on several Stanford technical reports.[45] Forsythe's article answered the question, "What is Computer Science?" in ways intended to raise awareness of the field's great potential and of the feasibility of many of the steps to reach that potential. Becoming familiar with what Forsythe wrote is essential for understanding why his vision was so appealing to so many at Stanford and beyond, and to understanding the challenges faced by computer science as the discipline emerged.

Forsythe's direct answer to "What is Computer Science?" would be uncontroversial today: "I consider computer science, in general, to be the art and science of representing and processing information and, in particular, processing information with the logical engines called automatic digital computers." He elaborated on this, again in a way that would raise few qualms in the 21st century:

> Computer science deals with such related problems as designing automatic digital computers and systems, the design and description of suitable languages for representing both processors and algorithms, the design and analysis of methods of representing information by abstract symbols, and of the complex processes for manipulating these symbols.[46]

In the mid-1960s, when Forsythe was writing the above, such answers were highly provocative, and needed to be defended with extraordinary precision and diplomacy. One issue was that many scientists, mathematicians, and engineers found the "science" component of the term "computer science" highly problematic and even offensive. Thus, Forsythe devoted a large part of his article to defining computer science as a science, albeit one that involved activities not undertaken in other scientific fields. The problem of legitimizing computer science as a science may seem quaint today, but in the 1960s amid the science spending boom, it was in the interest of academic computing specialists to have their activity viewed as *science*.

To establish computer science, and to establish how it could remain a science while distinct from other forms of science, Forsythe made a lengthy comparison of

---

[45] G. E. Forsythe. 1967. A university's educational program in computer science. *Commun. ACM*, 10(1): 3–11. The article is based on G. E. Forsythe. May 18, 1966. A university's educational program in computer science. Technical Report CS39, Stanford University, Computer Science Department. That report is based, in turn, on G. E. Forsythe. June 25, 1965. Stanford University's program in computer science. Technical Report CS26, Computer Science Department. And that report derives from memos, correspondence, and talks written by Forsythe in 1963 and 1964.

[46] G. E. Forsythe. 1967. A university's educational program in computer science. *Commun. ACM*, 10(1): 3.

Copyrighted material

it to physics. “The ultimate purpose of physics,” he argued, “is the intellectual one of understanding the physical world, and the role of experiment is the secondary one of assisting in that understanding.” In computer science the task was different, not in kind but rather in emphasis:

> Similarly, we may expect that someday the agreed ultimate purpose of computer science will be to understand the behavior of information and the laws which govern its processing. At the present time, however, many computer scientists consider the design of optimal processes and processors of information as the primary purpose of their subject, and relegate to theory the secondary role of suggesting techniques, methods of analysis, and proofs of optimality. Whether one is primarily interested in understanding or in design, the conclusion is that, if experimental work can win half the laurels in physics, then good experimental work in computer science must be rated very high indeed.

What Forsythe meant by “laurels in physics,” were Nobel Prizes and recognition that one’s activities opened up productive new lines of scientific inquiry, and he pointed to Donald Glazer’s recent Nobel “solely for the design of a bubble chamber, since this opened up a new field of physics.”[47] Theoretical work, he maintained, would have to wait, although only a little while: “As long as computers continue to changing drastically every three or four years, there is scarcely a chance to sit down and contemplate the creation of a theory. In this respect our subject is reminiscent of early engineering, and also of mathematical analysis in the time after Newton. I wish to emphasize my belief that this is a passing stage of computer science.”[48]

If one could argue that technique in computer science was the same as it was in physics, there was also the matter of the object of study. Physics seemingly studied natural phenomena, whereas computer science studied artificial ones. Forsythe’s response was that computer science’s object of study was not a set of machines but rather “information—in both natural and artificial systems.” This area of activity covered “information representation—as in the genetic code or in codes for efficient message transmission, and the study of information-processing devices and techniques, such as computers and their programming systems.”[49]

---

[47] G. E. Forsythe. 1967. A university’s educational program in computer science. *Commun. ACM*, 10(1): 4.

[48] G. E. Forsythe. 1968. What to do till the computer scientist comes? *Am. Math. Month.*, 75(5): 456.

[49] G. E. Forsythe. 1967. A university’s educational program in computer science. *Commun. ACM*, 10(1): 3–11. Forsythe’s 1966 technical report goes into more detail about the nature of information.

Copyrighted material

## 3.6 Conclusion

George Forsythe did not invent the term "computer science"—that claim may well belong to Louis Fein.[71] And he was not the founder of the first computer science department; there were a half-dozen established before 1965.[72] What he did do, however, was to define computer science in a way that a wide audience could see both its potential and its feasibility. Today, many of his recommendations, such as creating mechanisms to encourage and reward the development of algorithms, seem to be the epitome of obvious sensibility, but there was a time when the correctness of such propositions was far from self-evident. In carefully and respectfully addressing the many seemingly quite valid arguments against investing in computer science as a discipline, Forsythe provided readers of *Communications* the language as well as the intellectual and institutional frameworks they needed to secure a place for computer science.

Forsythe's writings about the formative years of academic computer science suggest that perseverance was an essential component of his success. Many of his contemporaries also saw the merit of establishing computer science, but few so doggedly persisted in the face of well-reasoned skepticism and cruel setbacks. His work also shows the value of his refusal to take the importance of computing for granted. In our world, saturated as it is by computing, it is easy to lose sight of the effort that was required to create a space for people to think systematically about how to work productively with information.

---

[71] A. Leonard. June 10, 1999. "Can history survive Silicon Valley?" *Salon*. Available at http://www.salon.com/1999/06/10/stanford/.

[72] G. E. Forsythe. 1967. A university's educational program in computer science. *Commun. ACM*, 10(1): 4.

Copyrighted material

# 4

# Solving a Career Equation: The First Doctoral Women in Computer Science

### Irina Nikivincze

## 4.1 Introduction

During the rapid growth of computer science as a discipline in the 1960s and the 1970s, women were entering the fields of science, engineering, and mathematics in increasing numbers. In the late 1960s, about 11% of all bachelor's degrees in computer science were awarded to women, but only a handful of women pursued doctoral-level training.[1] A survey of 60 Ph.D.-granting computer science departments in the U.S. reported that the percentage of doctorates received by women had increased sharply in the 1970s, reaching nearly 10% by 1976. By earning advanced degrees in a technical field, these women were challenging gender stereotypes and literally "inventing [their] careers."[2] This chapter examines the career patterns of the first women to receive doctoral degrees in computer science in the U.S. between 1970 and 1976. These pioneers comprise a distinguished group of researchers who followed their interests, entered a new field, and persisted in computer science in both academia and industry while remaining largely invisible to the public.

---

[1] In the 1960s, women earned less than 3% of all doctoral degrees in computer science. See Figure I.1 in J. McGrath Cohoon and W. Aspray, editors. 2006. *Women and Information Technology: Research on Underrepresentation*, p. x. MIT Press, Cambridge, MA.

[2] O. E. Taulbee and S. D. Conte. 1977. Production and employment of Ph.D.'s in computer science—1976. *Commun. ACM*, 20(6): 370–372; J. Abbate. 2012. *Recoding Gender: Women's Changing Participation in Computing*, p. 4. MIT Press, Cambridge, MA.

Copyrighted material

Association for Computing Machinery History Committee, and three oral histories collected by Janet Abbate. The interviews explore various stages of women's professional work: graduate training, career paths, work, recognition, gender issues, and strategies for success. Conclusions drawn from the interviews are supplemented by articles from *Communications of the ACM*, *Datamation,* and *Computerworld*.

The chapter begins by laying a historical foundation of the social, political, and economic factors influencing computer science and exploring gender dynamics in science and computing in the 1970s. It continues by describing the data and the research strategies used in building a cohort of women computer scientists and then reporting on the findings of an aggregated analysis of their careers in what can be called a "collective biography" of the cohort. The chapter ends with a summary of findings and the conclusion.

## 4.2 Historical Context

Before computing could develop into "an independent new science," it had to be accepted as a legitimate discipline. The development of computer science was a long process muddled by an uncertain identity of the new area of inquiry.[6] The work associated with early computers was performed by employees with a range of educational backgrounds in mathematics, physics, and engineering. A number of well-known female computer scientists such as Grace Hopper, Mina Rees, and Evelyn Boyd Granville were trained as mathematicians, others such as Thelma Estrin and Martha Sloan as engineers, and others such as Gwen Bell as geographers. The appearance of training and degrees defined and launched a lengthy process of solidifying the professional identity of computer professionals.

Computing first found its place in academic departments of engineering and mathematics and in the 1960s made its claims as an independent science. However, the earliest computer departments in the U.S., according to Edsger Dijkstra, "were no more than ill-considered cocktails of presumably computer-related topics that happened to be available on campus."[7] Although Purdue University established the first department of computer science in the U.S. in 1962, it was not until 1965 that preliminary curricula published by the Association for Computing Machinery

---

[6] J. Hartmanis. 1994. On computational complexity and the nature of computer science. *Commun. ACM*, 37(10): 37–43, quote on 41; S. V. Pollack, editor. 1982. *Studies in Computer Science*. Mathematical Association of America, Washington, DC.

[7] E. W. Dijkstra. 1986. EWD952: Science fiction and science reality in computing, 1986. Available at http://www.cs.utexas.edu/users/EWD/ewd09xx/EWD952.PDF.

Copyrighted material

ties had anti-nepotism laws that prohibited the hiring of spouses, creating a major problem for dual-career couples.[12]

The result of the tremendous expansion of universities and laboratories from the late 1950s to early 1970s—the "golden age" for science in America—was in fact a "very dark age" for academic women. Few women were faculty members at the 20 leading research universities. Although their numbers were higher in departments of home economics, genetics, anatomy, psychology, physiology, and bacteriology, women constituted no more than five percent of all faculty in the remaining 16 science fields. Even having one woman (a practice commonly referred to as "tokenism") "would have been quite an advance from the norm at prestigious universities at the time." Instead of academia, women scientists tended to opt for positions in industry, nonprofit organizations, or the federal government, or for entry-level jobs in technical fields. Even in these positions, however, women faced "the start of a career-long struggle with a succession of gender issues."[13]

In the second half of the 20th century, science remained an institution with "immense inequality in career attainment." Women often occupied the margins of the scientific community when it came to visibility, institutional location, and status. As newcomers, many stated that they were treated differently. Prominent scientists such as Evelyn Fox Keller, Barbara McClintock, Salome Waelsch, Andrea Dupree, and Sandra Panem often considered themselves to be outsiders and even "deviants" in the world of science.[14] By venturing into science, these women encountered estrangement, ambivalence toward their work, and invisibility. The persistence of such experiences prompted Margaret Rossiter in 1993 to name the phenomenon of invisibility and the recurring denial of credit for achievements of women the "Matilda effect." Unfortunately, this situation was not much different for women in academic computing.[15]

---

[12] M. W. Rossiter. 1995. *Women Scientists in America: Before Affirmative Action, 1940–1972*, p. 122. Johns Hopkins University Press, Baltimore.

[13] Ibid., 123, 129, 142.

[14] J. S. Long and M. F. Fox. 1995. Scientific careers: Universalism and particularism. *Ann. Rev. Sociol.*, 21:45–71, quote on 45; E. F. Keller, 1985. *Reflections on Gender and Science*. Yale University Press, New Haven, CT; H. Zuckerman, J. R. Cole, and J. T. Bruer, editors. 1992. *The Outer Circle: Women in the Scientific Community*. Yale University Press, New Haven, CT.

[15] Rossiter documented the under-recognition and invisibility of women in science and in history in general and suggested that this phenomenon be called the "Matilda effect" in memory of Matilda J. Cage, a late 19th century American suffragist who became aware of and denounced a widespread practice of the denial of credit to women. See M. W. Rossiter. 1993. The ~~Matthew~~

Copyrighted material