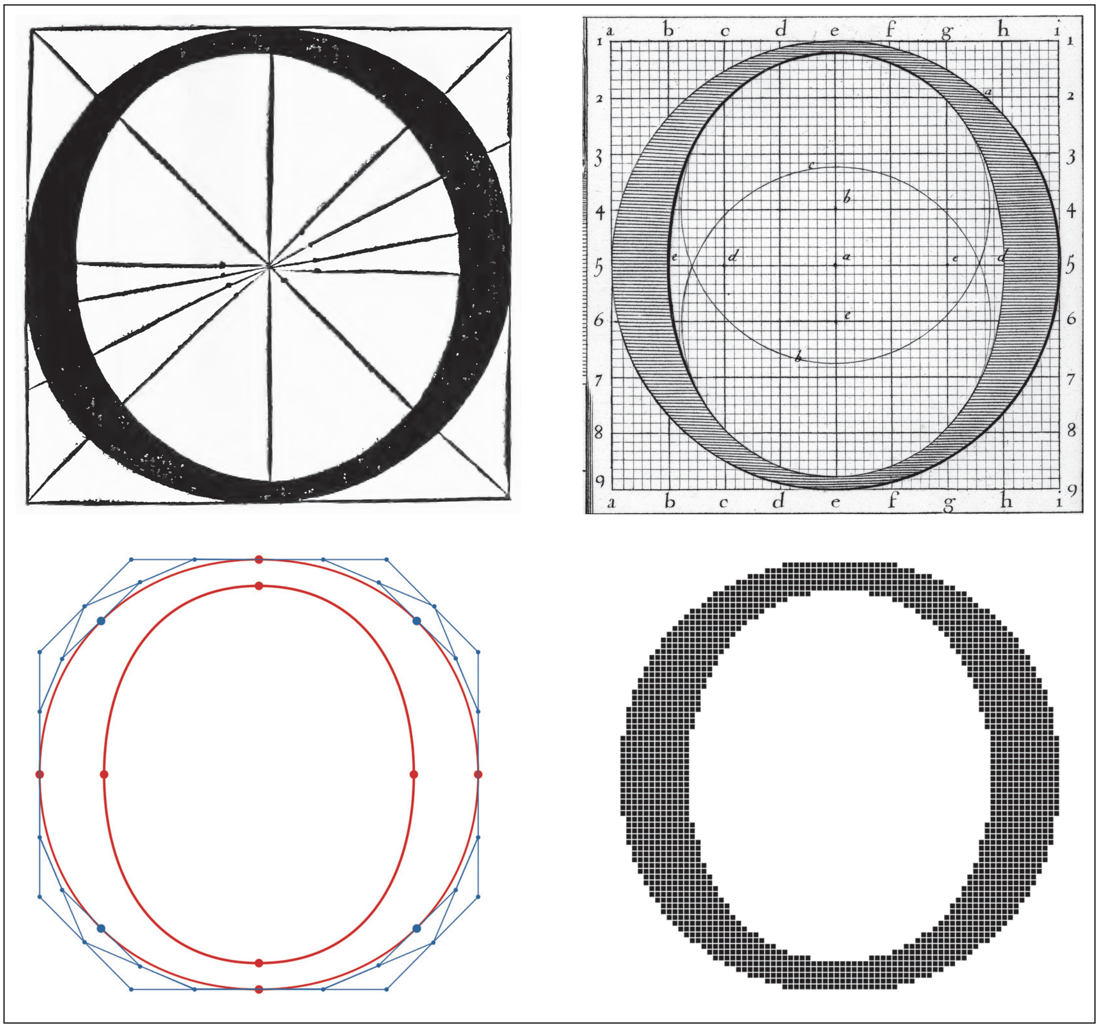
IEEE
Annals
of the History of Computing

Volume 42 Number 1 JANUARY-MARCH 2020



History of Desktop Publishing: Font Technology and Marketing

IEEE

IEEE COMPUTER SOCIETY

www.computer.org/annals

COMPSAC is the IEEE Computer Society Signature Conference on Computers, Software and Applications. It is a major international forum for academia, industry, and government to discuss research results and advancements, emerging challenges, and future trends in computer and software technologies and applications. The theme of COMPSAC 2020 is "Driving Intelligent Transformation of the Digital World".

Staying relevant in a constantly evolving digital landscape is a challenge faced by researchers, developers, and producers in virtually every industry and area of study. Once limited to software-enabled devices, the ubiquity of digitally-enabled systems makes this challenge a universal issue. Furthermore, as relevance fuels change, many influencers will offer solutions that benefit their own priorities. Fortunately, history has shown that the building blocks of digital change are forged by those conducting foundational research and development of digital systems and human interactions. Artificial Intelligence is not new, but is much more utilized in everyday computing now that data and processing resources are more economically viable, hence widely available. The opportunity to drive the use of this powerful tool in transforming the digital world is yours. Will your results help define the path ahead, or will you relegate those decisions to those with different priorities for utilizing intelligence in digital systems? COMPSAC has been and continues to be a highly respected venue for the dissemination of key research on computer and software systems and applications, and has influenced fundamental developments in these fields for over 40 years. COMPSAC 2020 is your opportunity to add your mark to this ongoing journey, and we highly encourage your submission!

COMPSAC 2020, organized as a tightly integrated union of symposia, will focus on technical aspects of issues relevant to intelligent transformation of the digital world. The technical program will include keynote addresses, research papers, industrial case studies, fast abstracts, a doctoral symposium, poster sessions, and workshops and tutorials on emerging and important topics related to the conference theme. Highlights of the conference will include plenary and specialized panels that will address the technical challenges facing researchers and practitioners who are driving fundamental changes in intelligent systems and applications. Panels will also address cultural and societal challenges for a society whose members must continue to learn to live, work, and play in the environments the technologies produce.

Authors are invited to submit original, unpublished research work, as well as industrial practice reports. Simultaneous submission to other publication venues is not permitted. All submissions must adhere to IEEE Publishing Policies, and will be vetted through the IEEE CrossCheck portal. Further info is available at www.compsac.org. Conference authors and authors of previously published papers should visit **https://ieeecompsac.computer.org/2020/jc-cj/** to view special publishing opportunities available at COMPSAC.

Organizers
Standing Committee Chair: Sorel Reisman (California State University, USA)
Steering Committee Chair: Sheikh Iqbal Ahamed (Marquette University, USA)
General Chairs: Mohammad Zulkernine (Queen's University, Canada), Edmundo Tovar (Universidad Politécnica de Madrid, Spain), Hironori Kasahara (Waseda University, Japan)
Program Chairs in Chief: W. K. Chan (City University, Hong Kong), Bill Claycomb (Carnegie Mellon University, USA), Hiroki Takakura (National Institute of Informatics, Japan)
Workshop Chairs: Ji-Jiang Yang (Tsinghua University, USA), Yuuichi Teranishi (National Institute of Information and Communications Technology, Japan), Dave Towey (University of Nottingham Ningbo China, China), Sergio Segura (University of Seville, Spain)
Local Chairs: Sergio Martin (UNED, Spain), Manuel Castro (UNED, Spain)

Important Dates
Main conference papers due: Extended to 13 February 2020
Main conference paper notification: 3 April 2020
Journal/Conference (JC) submissions due: 10 April 2020
Journal/Conference (JC) notifications: 30 April 2020
Workshop papers due: 9 April 2020
Workshop papers notification: 1 May 2020
Camera-ready and registration due: 15 May 2020

IEEE COMPUTER SOCIETY

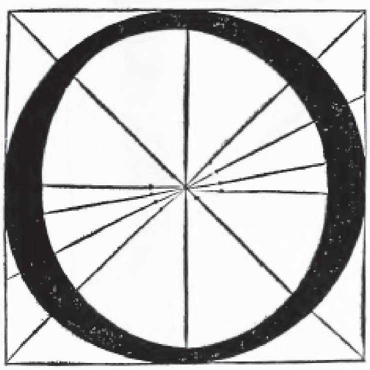
Photo: King Felipe III in Major Square, Madrid
Photo credit: Iria Castro - Photographer (Instagram @iriacastrophoto)

Contents

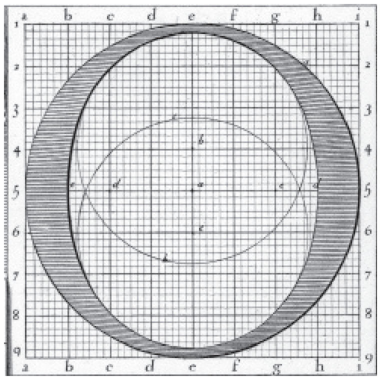
January-March 2020
Volume 42 Number 1

Published by the IEEE Computer Society

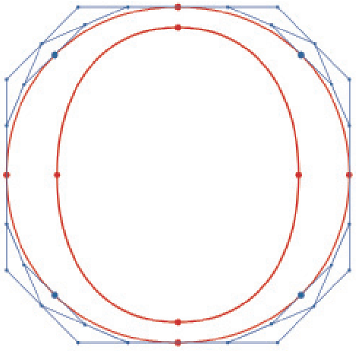
ISSN: 1058-6180



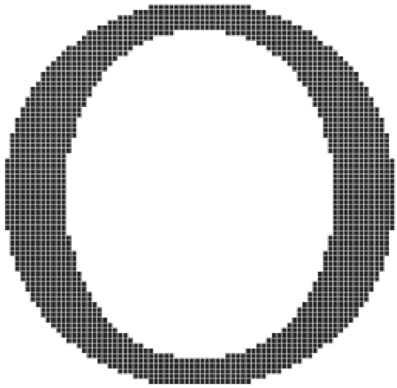
Capital ‘O’ designed with circular arcs and radiuses by Luca Pacioli, woodcut from *Divina Proportione*, Venice: Paganinus, 1509



‘O’ in circular arcs in a Cartesian coordinate grid, from Romain du Roi, Sébastien Truchet, engraving by Louis Simonneau, c. 1695.



Capital ‘O’ with outer contour of Bézier curves showing first stage of de Casteljau’s recursive subdivision algorithm, diagrammed by Kris Holmes



Capital ‘O’ bitmap, equivalent to 14 point type at 600 pixels per inch, raster designed by Kris Holmes

IEEE
Annals
of the History of Computing

IEEE Annals of the History of Computing

Published by the IEEE Computer Society | www.computer.org/annals

IEEE Annals of the History of Computing (ISSN 1058-6180) is published quarterly by the IEEE Computer Society, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-1314, USA; phone +1 714 821-8380. IEEE Computer Society Headquarters, 1730 Massachusetts Ave. NW, Washington, D.C. 20036-4928, USA; IEEE Headquarters, 345 East 47th St., New York, NY 10017, USA. Subscribe to *IEEE Annals* by visiting www.computer.org/annals. **Postmaster:** Send undelivered copies to *IEEE Annals of the History of Computing*, IEEE, Membership Processing Dept., 445 Hoes Lane, Piscataway, NJ 08855, USA. Periodicals postage paid at New York, NY, and at additional mailing offices. Canadian GST Registration No. 125634188. Canada Post Publications Mail Agreement Number 0487880. Return undeliverable Canadian addresses to P.O. Box 122, Niagara Falls, ON L2E 658. Printed in the United States.

IEEE COMPUTER SOCIETY

PURPOSE: The IEEE Computer Society is the world's largest association of computing professionals and is the leading provider of technical information in the field.

MEMBERSHIP: Members receive the monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.

COMPUTER SOCIETY WEBSITE: www.computer.org

OMBUDSMAN: Direct unresolved complaints to ombudsman@computer.org.

CHAPTERS: Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.

AVAILABLE INFORMATION: To check membership status, report an address change, or obtain more information on any of the following, email Customer Service at help@computer.org or call +1 714 821 8380 (international) or our toll-free number, +1 800 272 6657 (US):

- Membership applications
- Publications catalog
- Draft standards and order forms
- Technical committee list
- Technical committee application
- Chapter start-up procedures
- Student scholarship information
- Volunteer leaders/staff directory
- IEEE senior member grade application (requires 10 years practice and significant performance in five of those 10)

PUBLICATIONS AND ACTIVITIES

***Computer*:** The flagship publication of the IEEE Computer Society, *Computer* publishes peer-reviewed technical content that covers all aspects of computer science, computer engineering, technology, and applications.

Periodicals: The society publishes 12 magazines and 18 journals. Refer to membership application or request information as noted above.

Conference Proceedings & Books: Conference Publishing Services publishes more than 275 titles every year.

Standards Working Groups: More than 150 groups produce IEEE standards used throughout the world.

Technical Committees: TCs provide professional interaction in more than 30 technical areas and directly influence computer engineering conferences and publications.

Conferences/Education: The society holds about 200 conferences each year and sponsors many educational activities, including computing science accreditation.

Certifications: The society offers three software developer credentials. For more information, visit www.computer.org/certification.

BOARD OF GOVERNORS MEETING

28 – 29 May: McLean, Virginia

COMPUTER SOCIETY OFFICES

Washington, D.C.: 2001 L St., Ste. 700, Washington, D.C. 20036-4928; **Phone:** +1 202 371 0101; **Fax:** +1 202 728 9614; **Email:** help@computer.org
Los Alamitos: 10662 Los Vaqueros Cir., Los Alamitos, CA 90720; **Phone:** +1 714 821 8380; **Email:** help@computer.org

MEMBERSHIP & PUBLICATION ORDERS

Phone: +1 800 678 4333; Fax: +1 714 821 4641; Email: help@computer.org

IEEE

revised 14 January 2020

From the Editor's Desk

Gerardo Con Diaz
University of California, Davis

Welcome to a new year at *IEEE Annals!* Our year opens with the final issue of our desktop publishing series. We will then publish a fascinating collection of articles and department contributions on topics ranging from human–computer interaction to digital capitalism. I look forward to sharing these wonderful works with you and invite you to reach out to me or any of our editors if you are interested in publishing your work with us.

I am delighted to start the year with the announcement of a new award created by the IEEE Computer Society: the Best Paper Award for the *IEEE Annals of the History of Computing*. The inaugural winners of this award are Stephanie Dick and Daniel Volmar, for the paper, "DLL Hell: Software Dependencies, Failure, and the Maintenance of Microsoft Windows" (40:4 October-December 2018). Our search for the winners of the next award starts with this issue. If you would like to nominate any articles you read over the course of the coming year, please reach out to our Associate Editor-in-Chief, David Hemmendinger (hemmendd@union.edu).

Gerardo Con Diaz is the Editor-in-Chief of *IEEE Annals* and an Associate Professor of Science and Technology Studies, University of California, Davis, CA, USA. Contact him at condiaz@ucdavis.edu.

Digital Object Identifier 10.1109/MAHC.2020.2974386
Date of current version 6 March 2020.

Guest Editors' Introduction

Desktop Publishing, Part 3

Font Technology and Marketing

Burton Grad
Software Industry SIG, Computer History Museum

David Hemmendinger
Union College

THIS IS THE third special issue of *Annals* devoted to the history of desktop publishing. The first issue appeared in the fall of 2018, and was about the developments in the 1960s and 1970s of computer-driven printing and the technologies that led to the commercial growth of the desktop publishing industry. The second issue, in the fall of 2019, continued the story by describing how those technologies, developed in the 1970s particularly at Xerox PARC, became the foundation for the growth of major desktop-publishing software companies, with articles by founders of Adobe, Aldus, Quark, Frame Technology, and Ventura.

The prediction that computers would end the era of print publication goes as far back as Vannevar Bush's 1945 Memex. The first two *Annals* special issues on desktop publishing have documented what happened instead, as computing became central to publishing. They were primarily about the technology and the companies that created the digital publishing industry and made it so popular. We continue that documentation in this issue, and also have two articles on the history of font design, which is central to all publishing, both traditional "hot-metal" and digital.

All three special issues stem from a Desktop Publishing Pioneer meeting held in May 22–23, 2017, sponsored by the Software Industry SIG and the Software History Center, at the Computer History Museum in Mountain View, California, Transcripts of the workshops conducted during the meeting and of oral history interviews of some participants are at: www.computerhistory.org/collections/oralhistories (search for "desktop publishing"). Further supplemental material for the special issues is at history.computer.org/annals/dtp, which also has links to the meeting transcripts and interviews.

The first *Annals* issue on desktop publishing had an interview with font designer Charles Bigelow, and this issue contains his two-part historical account of the development and use of digital font technology by companies struggling for market dominance ("Font Wars") in the evolving desktop publishing world. He provides a *tour de force* analysis of the elements involved in creating fonts and why this was such a vital element in enabling desktop publishing to produce high quality publications. It also has an interview with Liz Bond Crews, who was part of the initial Adobe effort to market PostScript. She presents a broad view of the critical importance of the use of standard fonts and describes how vital the availability of high-quality fonts was in the selection of hardware platforms and software.

The Interleaf company was not represented at the meeting, but we now have an Anecdote by Mark Dionne and David Walden about Interleaf's pioneering work in high-end desktop publishing, and an interview of Larry Bohn, vice president for production planning and marketing at Interleaf. There are yet other companies that had a role in the development of desktop publishing, including Imagen, which made innovative laser printers, and Bitstream, which contributed to font technology. Although those companies were not included, we hope that these special issues may be a basis for further historical work.

ACKNOWLEDGMENTS

We thank P. McJones and D. Walden for conducting and editing the two interviews and also thank the reviewers for their help with the articles.

Digital Object Identifier 10.1109/MAHC.2020.2971285
Date of current version 6 March 2020.

The Font Wars, Part 1

Charles Bigelow
Cary Graphic Arts Collection,
Rochester Institute of Technology

***Abstract*—The Font Wars were a decades-long competition in the computer industry for dominance in font technology, viewed as a key success factor for personal computing platforms. The Font Wars spurred innovative scientific research into the small, nearly subliminal forms of the printed letters on which modern civilization was based, yet which had received little scrutiny outside the printing trades. More than a business episode, the Font Wars were above all a manifestation and translation of ideas—some modern, some ancient, some theoretical, some practical—into computer software and hardware. At the heart of the Font Wars was a fundamental question: What is the best way to turn traditional printed letter forms into digital fonts for computer screens and printers? Answers to this question were researched, implemented, and launched into the marketplace, where their intense competition transformed the 500 year tradition of printing and publishing, placing the electronic literacy on the screens of billions of digital displays, computers, tablets, and smart phones around the world.**

FONT WARS I: RASTERS AND OUTLINES

The font wars broke out in public on September 19, 1989 at the Seybold Desktop Publishing Conference in San Francisco, where Bill Gates of Microsoft announced a surprise partnership with rival Apple in a new font technology called "Royal." Gates claimed that Royal was superior to the PostScript font technology of Adobe, the leader in desktop publishing technology. When John Warnock of Adobe later took the podium, he called the Royal announcement "the biggest bunch of garbage mumbo-jumbo" he had ever heard, declaring, "What those people are selling you is snake oil."[1–3]

Those dueling proclamations at the Seybold Conference launched digital font technology into prominence in the war for dominance in personal computing. When competing personal computer platforms, including Apple, Microsoft, NeXT, and Sun Microsystems, adopted graphical user interfaces, font technology emerged as a key success factor.

Computer font technology had begun in the typesetting industry in the 1960s, but readers had barely noticed. Newspapers, magazines, and books typeset with digital machines looked much

Digital Object Identifier 10.1109/MAHC.2020.2971202
Date of current version 6 March 2020.

the same as when set with analog machines. The goal of digital typesetting was to keep the look of existing fonts and print while making typographic prepress production faster and cheaper.

From the 1960s onward, rapid progress in computing hardware and computer graphics stimulated new font technology concepts, many of which were little known outside research labs and professional journals. To the audience at the 1989 Seybold Conference, the flurry of font technology buzzwords may well have sounded like the mumbo, jumbo, and snake oil that Warnock famously decried. Vectors versus bitmaps, Bézier cubics versus quadratic B-splines, declarative "hints" versus procedural "instructions." What were those things? Where did they come from?

By the time of the Font Wars, typography was five centuries old and a foundation of modern civilization, yet there had been little scientific study of the shapes of the printed letters that had been the building blocks of the Renaissance, the Enlightenment, and the Industrial Revolution.[4] In the last three decades of the 20th century, computer scientists and engineers sought to answer the fundamental question: What is the best way to render traditional printed forms of letters and characters into digital fonts for computer screens, typesetters, and printers? [note 1][*]

The answers to that question were ideas, which in the Font Wars were not only technical methods but also tactical weapons in battles over hegemony in personal computing. The power of technological ideas in economic growth was analyzed by Paul Romer in a macroeconomic model for which he received a Nobel Prize in 2018.[5]

Three Kinds of Ideas

The main ideas involved in the Font Wars were of three kinds: Mosaics, Outlines, and Structures. These ideas were "nonrival" in Romer's term. They spread and were shared. The most ancient ideas became so deeply embedded in literacy that they seem obvious, not really ideas at all. In classical Greek culture, however, the concept of the hero who receives lasting fame for feats in battle was applied also to originators and explicators of ideas. Epic heroes like Achilles and Odysseus gained lasting fame, but so did intellectual heroes like Plato, Euclid, and Archimedes. The Greek tradition was institutionalized in early modern times with the founding in 1665 of the *Philosophical Transactions of the Royal Society*, which published and recognized authorship and priority of scientific discovery, as scientific and scholarly journals continue to do. [note 2]

Mosaics: The word "mosaic" comes from a Latin term for an image composed of small tiles, and may derive from the Greek word for "Muse." Today, mosaic displays are on billions of smartphones and computer screens, where tiny, illuminated "picture elements" (abbreviated "pixels" or "pels") are the equivalents of tiles. A subcategory of mosaic in which pixels are arrayed in parallel rows is a "raster," a word derived from the Latin word for "rake." Single-bit (black or white) pixel-based images are termed "bitmaps," and fonts for raster displays and printers are called "bitmap fonts." Screens with color and gray level pixels may also be called bitmap displays.

Outlines: The major font technologies of the Font Wars used outlines to represent letters and characters by mathematical equations of straight lines and curves. For display or printing on a raster device, letter outlines are first traced on a raster grid and then converted into patterns of pixels, a process commonly called "rasterization." Outline fonts are economical ways of storing and transforming character shapes, but mosaic output is usually the final stage of display or printing of texts.

Structures: Over centuries and millennia of cultural evolutions, typographic characters acquired structural regularities, including alignments and symmetries that readers intuitively feel are important for legibility and beauty. At the relatively low screen and printer resolutions during the Font Wars, the rasterization of outline characters often disrupted structural regularities. To satisfy reader expectations and perceptions, inventors of digital font technologies invented ways to impose structural regularity on rasterization of outline characters.

MOSIACS AND RASTERS

Mosaics

Ancient mosaics and writing: The idea of arranging graphical symbols in rectilinear arrays

[*]References of the form [note n] are to extended notes at https://history.computer.org/annals/dtp/fw.

was invented around 5000 years ago in the early cities of ancient Mesopotamia, where the Sumerians invented a writing system of wedge-shaped marks impressed into clay tablets. Today the writing is called "cuneiform" ("wedge-shaped").[6] Through centuries of refinement and modification, cuneiform writing became organized into parallel rows arranged into columns, as in the *Annals of Sennacherib,* a history of Assyrian king Sennacherib's wars (see Figure 1).

Cuneiform was the dominant information technology of ancient Mesopotamia for thousands of years, used by Sumerian, Babylonian, Elamite, Assyrian, Hittite, and Persian cultures. It was eventually replaced by alphabetic writing, which likewise was arrayed in rows and columns, as is typography today.

Sumerians decorated building walls with mosaic patterns, but a connection between their decorative mosaics and the quasi-mosaic structure of their writing system remains unclear. Sumerian ideas nevertheless remain common today. Centuries after inventing writing, the Sumerians also composed the first known lyrics in praise of beer,[7] a literary genre widely heard and seen today in commercials displayed on mosaic raster television screens.

Classical and medieval mosaic texts: Ancient Greeks and Romans adorned building floors with pictorial mosaics of mythological, natural, or historical scenes, sometimes with mosaic texts. A mosaic of a barking dog captioned with the words "CAVE CANEM" ("Beware of the Dog") was excavated at the entrance to the "House of The Tragic Poet" in Pompeii, covered by the eruption of Mount Vesuvius in 79 CE (see Figure 2). Some Medieval European mosaics depicting Christian saints were captioned with names and texts, as in the Palatine Chapel in Palermo, Sicily, its mosaics constructed in the 12th century CE. In Islamic architecture, mosaics of intricate mathematical sophistication occasionally incorporated texts of prayers and Koranic phrases, as in The Dome of the Rock in Jerusalem, constructed around 692 CE.

19th century pointillism: Mosaic images were "in the air" in 19th century Europe. Georges Seurat, Paul Signac, and other "pointillist" and "divisionist" painters used small daubs of paint to compose scenes that merged into recognizable images when viewed at a distance. A popular form of color printing called "chromolithography" used tiny dots hand-stippled onto lithographic stones and reproduced. Photographic images were projected through screens to be rendered into arrays of dots of different sizes for "halftone" printing.

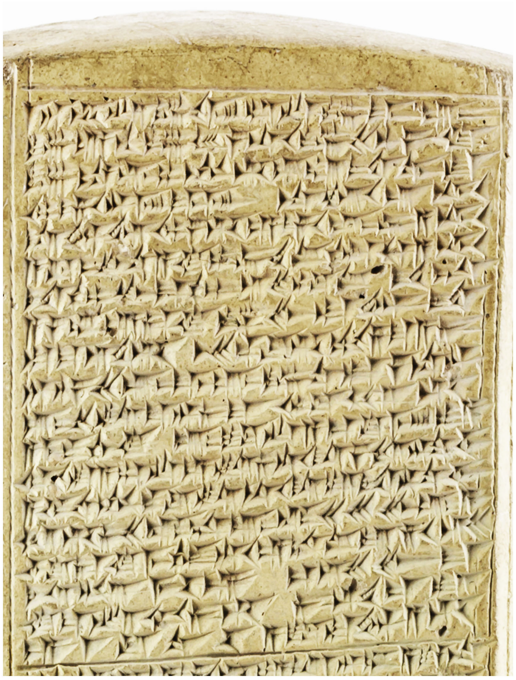


Figure 1. Portion of cuneiform prism: *Annals of Sennacherib* Courtesy of the Oriental Institute of the University of Chicago.

Earliest digital book: The first book of digital typography was an 1878 edition of *Les Laboureurs*, an extract of a poem by Alphonse de Lamartine. The book's pages of rasterized type were encoded on thousands of Jacquard punched cards and woven in gray and white silk on a Jacquard loom by a lacemaking firm in Lyon, France (see Figure 3). The jewel-like precision of the book type has a digital resolution comparable to laser printer resolutions of a century later. [note 3]

Early *20th century mosaic text:* Several electrical methods of mosaic imaging were invented in the early 20th century, among them the "Bartlane" transmission system first used in 1921 to transmit news photos by transatlantic cable. The "Hellschreiber" device patented by Rudolf Hell in 1929 transmitted low-resolution digital text over telephone lines. Several other 20th century inventions of digital text are described and illustrated by Jacques André in a history of 20th century typography.[8]

Figure 2. Mosaic from Pompeii. (https://commons.wikimedia.org/wiki/File:Dom_dramaturga.jpg).

Engineers were not the only ones to recognize the potential of digital type. In 1964, Beatrice Warde, one of the most gracefully articulate writers on typography of all time, glimpsed the future of typography in an animated digital display in Times Square. The display was a mosaic of 4000 light bulbs that transformed moving digital text into moving pictorial images. It was created by Canada's innovative animator, Norman McLaren. Warde wrote of it: "The creative dramatic mind makes itself at home in the still-new Electric Age, seizing on possibilities which have never twinkled anywhere in the Gutenberg Galaxy."[9] [note 4]

Raster Typesetters, Printers, and Displays

The first electronic typesetting machine with raster fonts was the Hell Digiset, invented around 1964–1965 by the German firm of Rudolf Hell AK. The first operational version was commercially

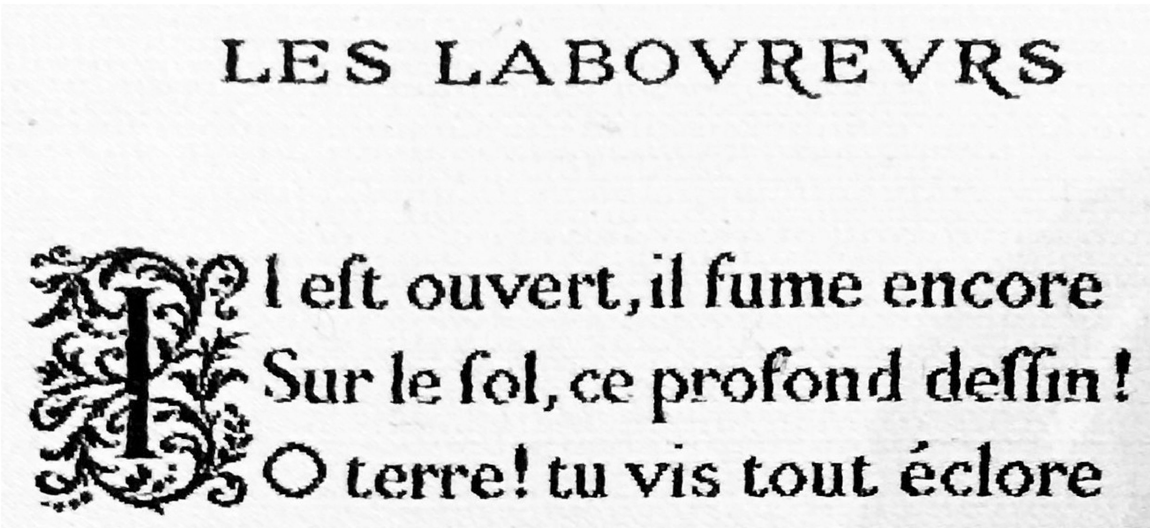


Figure 3. Jacquard-woven text and ornament from *Les Laboureurs* (image magnified 3x). Courtesy of the Cary Graphic Arts Collection of Rochester Institute of Technology.

installed in 1966 (some sources say 1967). Like a television, the Digiset used a scanning electron beam controlled by magnetic fields, with precise on/off control, to draw letters on the screen of a phosphor-coated cathode-ray tube (CRT). Designed as bitmaps, the letters were painted on the CRT phosphor screen as a series of parallel vertical stripes and optically transferred to photosensitive film or paper. The Digiset 50T1 could compose type at around 600 characters per second—much faster than hot-metal or phototypesetting machines. In the U.S., RCA combined Digiset hardware with RCA software and sold the unit as the VideoComp, beginning in 1968. The principles of the VideoComp were described in an article in *Scientific American* in 1969.[10] [note 5]

Raster laser printers: In 1969, Gary Starkweather at a Xerox facility in Webster, New York, invented the first raster-based laser printer. It used a spinning mirror to deflect a laser beam to "write" pixel-based text and images on an electrostatic copier drum that transferred and fused the toner image to paper, as in a Xerox copier. Starkweather soon moved to Xerox PARC, where he developed a series of experimental laser printers.[11]

In the early 1970s, Xerox developed an electrostatic printer with a raster resolution of 192 dots per inch—the XGP (Xerox Graphics Printer), and seeded them to computer science groups at California Institute of Technology, Carnegie Mellon, MIT, Stanford, and University of Toronto, where faculty and students developed hardware and software interfaces to computer systems. In 1977, Xerox launched a commercial laser printer, the 9700, which could print and collate two pages per second, with images containing raster fonts and digitized graphics. Its resolution of 300 dots per inch enabled high-speed printing with acceptable type quality. The 9700 was popular for high-speed printing of data-driven documents by banks, credit card companies, and other firms and organizations. Xerox management initially predicted they would sell around 600 of the machines, but instead sold 10,000. On the 40th anniversary of the introduction of the 9700, Xerox said the printer generated up to one billion dollars of revenue per year for Xerox.[12]

Mass-market laser printers: Xerox did not initially pursue the low-cost market in laser printers, but the Japanese Canon company developed a few laser printers that sold modestly, until 1983, when Canon introduced the LBP-CX, a low-cost laser diode printer engine that used dry toner. The base machine was called a "marking" engine or printer engine because it could print raster dots on paper, but needed a computer to control where the dots are placed. With LBP-CX print engines from Canon and a proprietary hardware and software controller, Hewlett Packard launched the LaserJet in 1984. Apple followed with the LaserWriter in 1985 with PostScript software and a custom hardware controller. The LBP-CX-based printers became popular because they were quieter and faster than dot-matrix and daisy-wheel printers, produced higher quality text output, and yet were small enough to sit on a desktop.

Raster displays: Raster text was displayed on computer monitors in the early 1970s at Bell Telephone Laboratories. [note 6] A few years later at the Xerox Palo Alto Research Center (PARC), raster fonts were used on the bitmap display screens of the experimental Alto personal computers as well as in laser printers.[12]

WYSIWYG and bitmap fonts: Bitmap fonts on display screens paired with laser printers introduced "What You See is What You Get" (WYSIWYG) typography, in which page layouts on computer screens simulated the look of printed pages, albeit at lower resolutions.[12] Around 1974, scientists at PARC combined the Bravo interactive text editor and the Alto personal computer with laser printers to produce a working WYSIWYG system that became popular in the PARC laboratory.

In 1981, after research and development to integrate personal computer technology with a human–computer interface, Xerox launched the Xerox 8010 Information System, familiarly known as the "Star" workstation. The 8010 "Star" used bitmap display and WYSIWYG typography. The Alto and the 8010 Star had nominal screen resolutions of 72 pixels per inch (ppi), matching the standard printers' point dimension.[13] WYSIWYG systems soon became popular because of their visually intuitive interfaces. In 1984, the Interleaf corporation in Boston launched a commercial WYSIWYG document system with bitmap displays, and Apple released the Macintosh

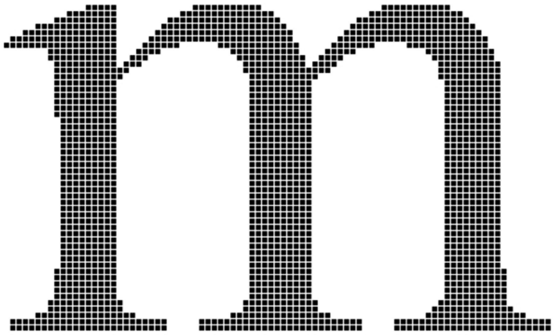


Figure 4. Bitmap of Times New Roman "m." Courtesy of Beat Stamm.

personal computer. Bitmap screen fonts were initially coarse compared to traditional fonts, but as resolutions of display screens and printers increased, familiar typefaces like Times New Roman became recognizable in raster form (see Figure 4).

Bitmap font size and storage: Bitmap font data are proportional to image area. Doubling the linear size of a bitmap font quadruples the data. For example, at a printer resolution of 300 dots per inch, a 12 point bitmap letter of average width might contain 1250 pixels, but at 48 point, it would contain 20 000 pixels. Thus, large size raster fonts required high storage capacity in an era of expensive random access memory and hard disks.

Run-length encoding: To compress raster font data, digital typesetting machines used run-length encoding, a lossless technique that stores the starting pixels of each scan run and its run length in pixels. Run-length encoding is more economical than bitmap encoding because it stores only pixels along contours, not internal pixels; thus, the contours scale linearly by size instead of by the square of linear size. Peter Karow, developer of the Ikarus technology for digitizing and processing fonts, stated that to store a 100 x 100 bitmap character, 1200 bytes may be needed, but to store a run-length encoding of the same raster character, only 450 bytes may be needed.[14,15]

In practice, run-length encoding is not optimal compression for fonts because, being raster-based, they are not easily scalable. Different font sizes and resolution require different font masters. Some digital typesetting machines used lenses, along with spacing and run length adjustments of the scanning CRT beam, to produce intermediate sizes between masters, but most machines still needed four to five master sizes for a full range of sizes for each typeface.

Raster Font Bottlenecks

As digital typesetting spread from telephone directories and newspapers to trade and book publishing, typesetting machine manufacturers had to produce more fonts. The early method of making digital raster fonts used electro-optical scanning of analog characters followed by editing the resulting bitmaps to remove stray pixels, improve contours, regularize alignments and features, and adjust spacing. In the late 1970s and early 1980s, raster font editing was a production bottleneck. A pioneer of outline font technology, Peter Karow, compared the relative efficiency of electro-optical scanning to careful hand digitization of outline fonts.[14]

OUTLINES

Data Compression Versus Processing Speed

Outline font technology compresses font data because only the perimeters of characters need to be stored. Letter outlines expressed mathematically as vectors and curves require less data than raster letters or run-length encoded letters, except at very small raster sizes. Another advantage of mathematically expressed outline letters is that they are easily modified by geometric transformations, such as slanting, stretching, rotating, mirroring, and other typographical variations.

The tradeoff between raster versus outline font data is that rasterization of an outline requires greater processing time, and the greater the data compression, the greater the processing time to rasterize it. Bitmap fonts are ready made and require the least processing in the machine. To convert outline letters to rasters, they must first be traced on an output pixel grid and the pixels along the contours marked as "in" or "out" of the letter. Next, the pixels of the interiors of characters must be filled in, to make the solid areas of the bitmaps that will ultimately be displayed or printed (see Figure 5).

The conflicting goals of high data compression, fast processing, and aesthetic quality spurred font technology innovation. Never before had so much

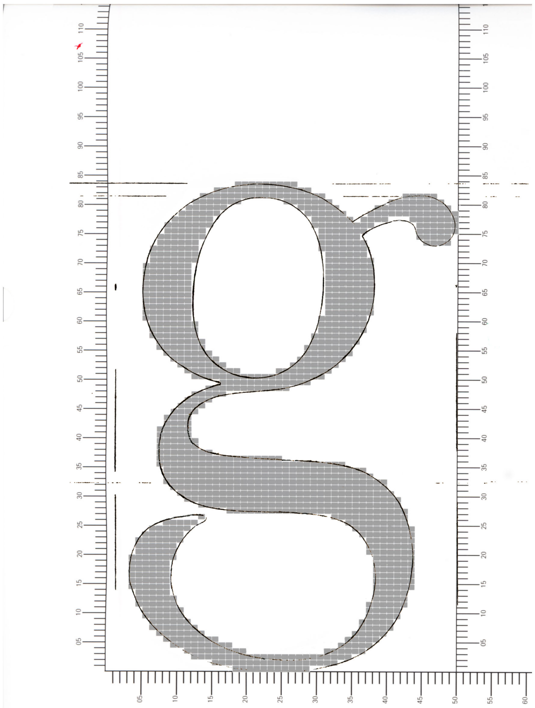


Figure 5. Outline and bitmap of Baskerville digital "g." Courtesy of Kris Holmes.

mathematical and computer research been so intently focused on such tiny cultural artifacts. Five centuries before the Font Wars, Renaissance scholars, artists, and mathematicians used Euclidean geometry to construct outlines of Roman capital letters, to imitate inscriptional letters chiseled in stone more than a thousand years earlier. Those intellectual and artistic exercises did not appreciably affect the making of type until the late 20th century, when electronic computation and the science of algorithms raised the idea of geometric letter outlines from concept to implementation, from theory to practice.

Big and Small Shapes

Early applications of computer graphics were for design and manufacturing of cars, airplanes, and ships—big, expensive, complicated, attention-getting objects compared to printing types. Although literate society depended on printing type for centuries, the details of type shapes were presumably ignored by most readers, except for generic distinctions, such as roman versus italic or roman versus blackletter. During the Font Wars, however, usage of the term "font" increased by a factor of five (according to Google Books Ngram).

A. R. Forrest, in his 1968 doctoral thesis, "Mathematical curves and surfaces for computer aided design," remarked on computer-graphical representations of vehicles and their components: "Cars, ships and aircraft provide obvious examples for surface methods, but it must be remembered that other shapes such as castings, forgings, turnings, etc., are used throughout industry."[16]

Fonts could be added to Forrest's list of "other shapes." When type forms were researched in computer graphics in the 1970s and 1980s, many of the inventions employed mathematics known from antiquity as well as mathematics published in the 19th and 20th centuries, including papers by Forrest.

Polygons

In early computer graphics, outline font formats were often based on polygons—closed contours made of connected straight line segments. The study of polygons goes back to ancient mathematics.

Polygons in Mesopotamia: The purpose of Babylonian mathematics was often utilitarian, for use in surveying, canal digging, and construction, but some cuneiform mathematical tablets present "supra-utilitarian" problems, perhaps for mathematical investigation and instruction rather than practical usage. Some mathematical tablets from the Old Babylonian period, roughly 2000 BCE to 1600 BCE, show triangles and, it is surmised, hexagons inscribed in circles, presumably for estimating the value of pi (π) and calculating circumferences and areas. At least one Old Babylonian tablet is a list of Pythagorean triples—the lengths of the three sides of right triangles which follow the Pythagorean theorem—showing that the Babylonian mathematicians understood the Pythagorean principle more than a thousand years before Pythagoras, though there are no extant Babylonian proofs of the theorem. [note 7]

Polygons in Greek mathematics: Traces of Babylonian mathematical problems appear much later in Greek mathematics. The routes of transmission are unknown, but a few ancient Greek historians claimed that Pythagoras as well as Thales had studied in Babylon and/or Egypt. Centuries after those early Greek mathematicians, Euclid in Book IV of *The Elements* demonstrated constructions of regular polygons inscribed in, or circumscribed about,

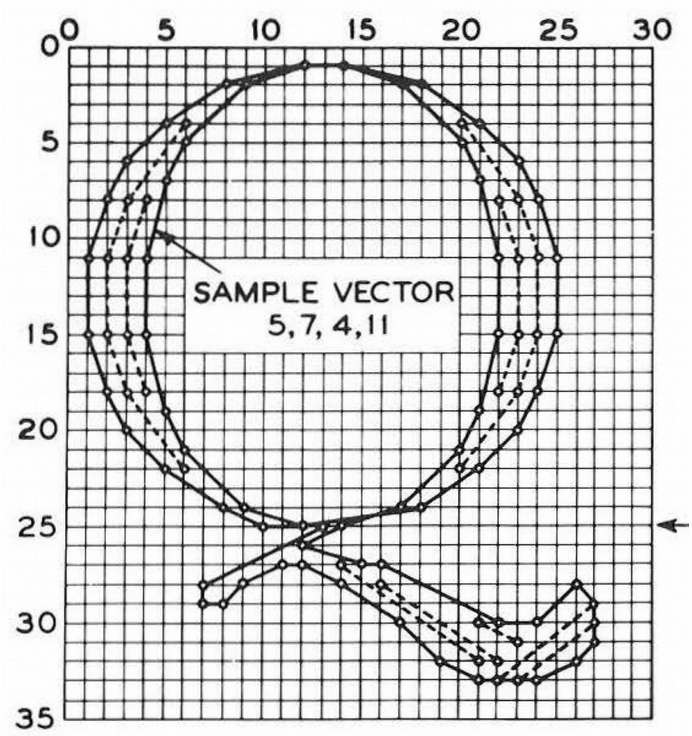


Figure 6. Polygonal outline "Q." Mathews *et al.* Courtesy of *Visible Language*.

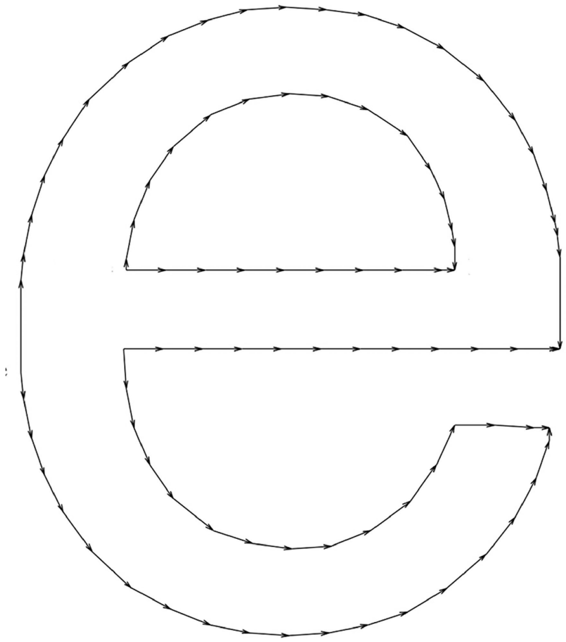


Figure 7. Linotron 202 polygonal "e." Courtesy of Brian Kernighan.

circles. A century later, Archimedes used polygonal constructions of increasing order—greater numbers of sides—to approximate the value of pi, using what is now called the method of exhaustion. Several Greek inscriptional letters were polygonal by design, their shapes composed of vertical, horizontal, and/or diagonal strokes, such as Alpha, Epsilon, Kappa, and Mu, but these were structures, not approximations to arcs. [note 7]

Polygonal outline letters for plotters and CRTs: In 1962–1963, Jack E. Bresenham at IBM devised an efficient, integer-based algorithm to draw lines on digital plotters.[17] In 1967, A. V. Hershey at the U.S. Naval Weapons Laboratory published a technical report, "Calligraphy for Computers," containing coordinates and images of a large repertoire of polygonal outline characters for CRT vector displays and plotters.[18] Also in 1967, Mathews *et al.* at Bell Laboratories developed vector-drawn polygonal fonts on a CRT display (see Figure 6).[19] [note 8]

Polygonal outlines in digital typesetting: The first commercially successful digital CRT typesetter with polygonal fonts was the Linotron 202, launched in 1978. The 202 stored letter outlines with polygonal approximations to curves and rasterized them "on-the-fly" as a CRT beam painted the shapes of letters in a series of parallel vertical lines on a phosphor-coated screen. The glowing lines merged into letter shapes to expose photographic paper or film. Like most digital typesetters of the era, the 202 scanned letter shapes vertically and advanced film or paper mechanically to produce lines of type in "galley" format (columns of text before make-up into pages). The 202 polygonal fonts were scalable from 4.5 point to at least 72 points in size.

The 202 was first marketed for typesetting newspapers, where the flat sides and angular vertices of the 202's polygonal approximations to curves were largely imperceptible at text sizes—around 8 to 12 points—but were somewhat detectable at larger headline sizes in higher quality publications (see Figure 7).

In 1979, the Computing Science Research group at Bell Laboratories bought a Linotron 202, but dissatisfied with the machine's buggy software and poor reliability, Joe Condon, Brian Kernighan, and Ken Thompson reverse-engineered its software and wrote a new operating system and software tools. They decoded the 202's secret font format, wrote programs to build and install user-defined fonts, to manipulate and upload scanned graphics, and to draw lines and circular arcs for diagrams. They interfaced their new 202 system software to the Labs' Unix computers and word processing programs, expanding the typesetter's utility far beyond commercial typesetting. But, when Linotype learned of those software advances, they asked Bell Labs to suppress publication, fearing font piracy if the 202's font format became public. As a notable example of exclusion of ideas, the 1980 technical report on the project by Condon *et al.* was not published until 33 years later.[20] [note 9]

CURVES

Mathematical curve contours use fewer points to define shapes and, thus, compress font data more than polygonal contours, while also eliminating objectionable faceting. As outlines, curve-based contours are scalable to different sizes before rasterization, obviating the need for different sizes of raster fonts.

Font formats have been based on quadratic (second degree) and cubic (third degree) curves. Formats based on segments of quadratic curves include circular arcs and parabolic arcs, as well as arcs of ellipses and hyperbolas. Formats based on segments of cubic curves usually are cubic polynomials, including Hermite cubic curves, Bézier cubic curves, and cubic B-splines.

Historically, circular arcs are the earliest mathematical curves used to describe letter contours and are also among the earliest used for computer fonts.

Circles

Circles in antiquity: Around 2000 BCE, ancient Babylonian mathematicians calculated approximations to pi (π) to measure circular circumferences and areas, and also devised the familiar sexagesimal division of the circle into the 360 degrees. More than 1600 years later, Greek mathematicians, notably Euclid, Apollonius of Perga, and Archimedes, rigorously investigated the geometry of circles, but without applying their findings to letters. Greek inscriptions were cut freehand, not calculated, although there is evidence that something like a compass was used to inscribe circular letters in an early Athenian inscription. [note 7]

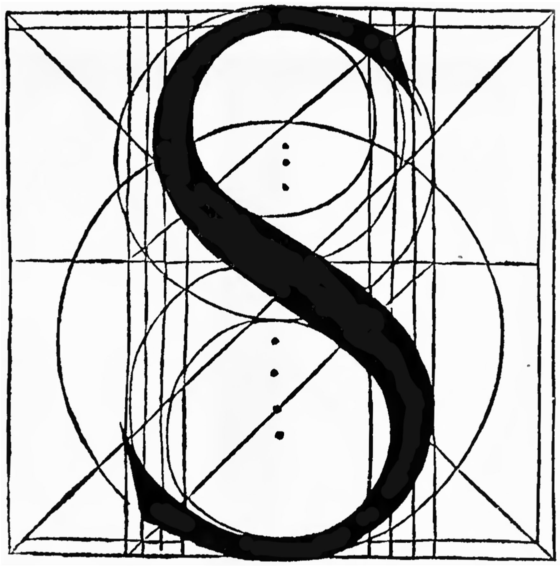


Figure 8. "S" constructed from circles by Luca Pacioli, *Divina Proportione*, Paganius Paganinus 1509.

Renaissance letter construction with circles: Vitruvius, a Roman architect and engineer of the first century BCE, wrote *De Architectura,* a set of ten books on architecture and proportion. After their rediscovery in the Italian Renaissance, his books inspired not only architecture but also geometrical constructions of Roman capital letters, including those by mathematician Luca Pacioli in his book *Divina Proportione* of 1509 (see Figure 8).

Later artists, including Albrecht Dürer, devised additional geometrical letter constructions of Roman capitals. In the 20th century, however, Edward Catich made rubbings, tracings, and meticulous revivals of the classical Trajan inscription carved in 113 CE in Rome, and demonstrated that Roman inscriptional letters had first been painted freehand with a broad brush and then chiseled freehand into stone. Catich's demonstrations cast doubt on the hypothesis of geometrical letter construction by ancient Roman stone carvers. [note 10]

Joseph Moxon & Circles: Inspired by Vitruvius on proportion and Dürer on letter construction, Joseph Moxon, an English maker of maps and globes, published in 1676 a book on geometric constructions of letters: *Regulae Trium Ordinum Literarum Typographicarum: or the Rules of the Three Orders of Print Letters: viz. The {Roman, Italick, English} Capitals and Small. Shewing how they*

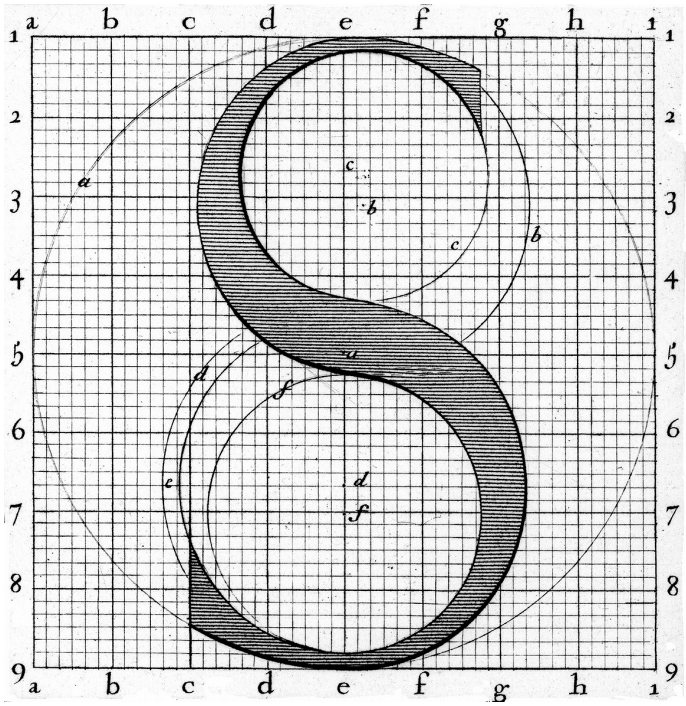


Figure 9. "Romain du Roi" "S" constructed from circles by Sébastien Truchet, circa 1695. Image: A. Jammes, 1961.

are compounded of Geometrick Figures, and mostly made by Rule and Compass. Useful for Writing Masters, Painters, Carvers, Masons, and others that are Lovers of Curiosity. Moxon measured and described proportions of typefaces cut by Christoffel van Dijck of Holland. [note 11]

Romain du Roi—Circles in cartesian coordinates: Around 1675, Louis XIV's polymath minister Jean-Baptiste Colbert initiated a program for the French Academy to describe the Arts and Crafts of France. This early instance of government funded industrial research led to the formation of a committee of savants headed by Abbot Jean-Paul Bignon, which in 1693 began to analyze the printing trades and to design a new set of fonts for the Royal printing house. The Bignon committee analyzed the shapes of printing types, devised the first typographic point system, and designed large geometric models of letters for a new printing type, the "Romain du Roi," in honor of Louis XIV. The letters were geometrically drawn with circular arcs and straight lines on high-resolution Cartesian grids and then engraved by Louis Simonneau on copper plates, approximately seven centimeters square (see Figure 9). Philippe Grandjean then hand-cut the steel master punches used to make the type. The Romain du Roi family of types included lowercase as well as capitals, and italic styles as well as roman, and influenced later type styles of the 18th century (see Figure 9). [note 12]

Digital Circles

In the 1960s, Jack E. Bresenham followed his line plotting algorithm with an integer-based algorithm for rasterizing circular arcs, not published until 1977.[21] In 1976, Berthold Horn also published a method of generating circles on rasters.[22] A later contribution to precision raster arc drawing came from Douglas McIlroy.[23] By the early 1970s, computer controlled plotters from the Aristo firm in Germany were able to draw circular arc-based contours and could plot font outlines produced by the Ikarus program.

Circular arcs 1—Ikarus: In late 1972 at the URW technology consulting firm in Hamburg, Peter Karow, a physicist who had worked on computer-aided design in ship building, began development of the "Ikarus" software system for digitizing, storing, and processing letter forms.[24] Karow used Hermite form cubic curves for input, manipulation, and storage of letter outlines, but because typesetting machines and computer plotters of that era could not process cubic curves, Karow programmed a conversion from Ikarus cubic to circular arc contours for computer-controlled plotters, particularly those from Aristo.

Circular arcs 2—Rockwell (MGD) Metro-set: In May 1973, Gregory W. Evans and Robert L. Caswell at the Rockwell Corporation invented a circular arc font format for a digital typesetter, the MGD Metro-set. The Metro-set was soon withdrawn from the typesetter market, but a patent by Evans and Caswell, also known as the Rockwell patent, was profitable through litigation against digital typesetter manufacturers, with repercussions in the Font Wars.[25] [note 13]

Circular arcs 3—Camex/Bitstream: In 1981, Camex, a Boston company making interactive workstations for newspaper advertising layout, converted six of its workstations into digital font production tools for Bitstream, a newly formed digital font company founded by former Linotype designers and managers.[26] The Camex-Bitstream "Letter Input Processor" (also called "Letter IP" or "LIP") used a cubic curve format

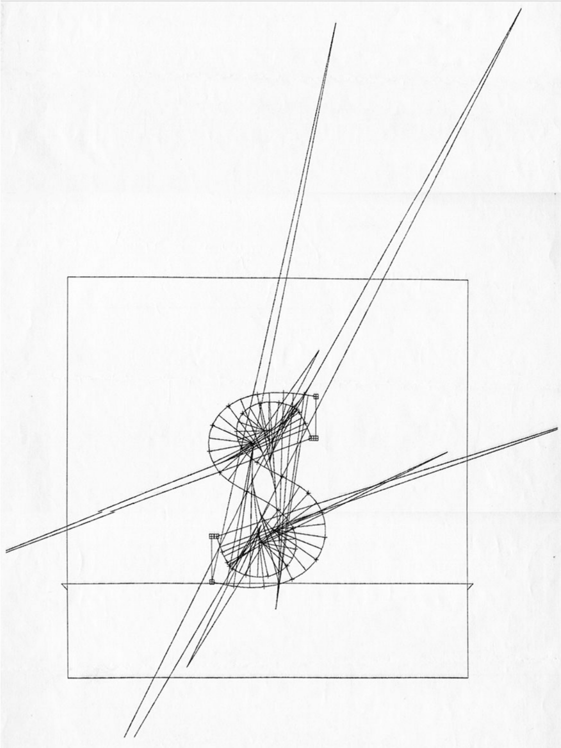


Figure 10. Arc-and-vector "s" with radiuses.

for input and storage of letter outlines but a circular arc and vector format as an output and distribution format (see Figure 10). The Letter IP enabled a designer to quickly input points of curves and line segments from an enlarged letter outline, using a digitizer tablet and "puck," while viewing input progress on a CRT screen.[26,27] The Letter IP enabled Bitstream to become a competitor to established companies like Linotype and Monotype, which were also entering the digital font licensing business. [note 14] Bitstream later developed outline-to-bitmap font rasterization software for license to OEM (Original Equipment Manufacturer) firms.

Conic Curves

Conic curves in antiquity: Conic curves are those produced by the intersection of a plane with a cone. Depending on the orientation of the plane relative to the cone, the resulting conic curve may be a circle or an ellipse, both of which are closed curves, or a parabola or a hyperbola, which are open curves.

In the Hellenistic period, roughly 320 to 30 BCE, the study of conic curves flourished among Greek mathematicians, especially Euclid, Archimedes, and Apollonius of Perga. Apollonius, called the "great geometer," wrote eight books on the study of conics, introducing the terms "ellipse," "parabola," and "hyperbola" still used today. [note 15]

Conics in the 17th century: René Descartes in *La Géometrie* (1637) and Pierre de Fermat, in manuscripts not published until after his death, analyzed conic curves algebraically. Neither mathematician applied his mathematical discoveries to outlines of letters. Decades later, the Bignon Committee used circular arcs, not algebraic conics, in plotting the shapes of the Romain du Roi on Cartesian grids. Regardless of whether letter outlines were circular or general conic arcs, there was no way to render them in type technology of the time. [note 12]

Conics in the 20th century: In 1940, Roy Liming, the head of Loft Mathematics at North American Aviation, calculated conic curves to "loft" (plot) the outlines of the fuselage of the renowned P-51 Mustang fighter plane of WWII. "Lofting" was the term for full scale drafting of shapes for aircraft surfaces and cross sections, often done in large lofts above aircraft factory floors. Liming's approach influenced aircraft design and lofting through the following decade.

In 1967, M. L. V. Pitteway devised algorithms for drawing conic curve segments on digital plotters.[28] In 1968, A. R. Forrest at Cambridge University treated conic curves in part of his Ph.D. thesis on mathematics for curves and surfaces in computer graphics.[16] In 1971, Amalie J. Frank at Bell Laboratories encoded raster characters in patches defined by parabolic curves. In 1983, Theodosios Pavlidis at Bell Laboratories noted the solid historical background of conic curves and their advantages in curve fitting over cubic splines, but did not apply conic curves to characters.

Conic curves for fonts 1—Pratt: Conic curves entered the Font Wars in 1984, when Vaughan Pratt talked informally about conic curves for fonts at the Stanford "Metafont for Lunch Bunch," a weekly research lunch organized by Donald Knuth to discuss his Metafont system of parametric font design. Pratt described algorithms for rasterizing conics, suggested a solution to the problem of continuity of curvature where conic curve sections join, and a solution to the problem of an inflection point where a curve changes direction, as in a letter "S." He showed that conic curves were significantly faster to compute than cubic curves, and in a nod to history, acknowledged Apollonius of Perga on conics. Pratt later presented his research in a paper, "Techniques for conic splines," at the 1985 SIGGRAPH conference.[29] He implemented his conic splines in graphical software at Sun Microsystems.

Conic curves for fonts 2—Imagen: Pratt's 1984 Stanford talk was heard by visiting engineers from the Imagen corporation, a laser printer manufacturer founded in 1980 by Les Earnest, administrator of the Stanford Artificial Intelligence Laboratory, and Luis Trabb-Pardo, associate director of Knuth's TeX project. In the fall of 1984, Michael Sheridan, director of typography at Imagen suggested to Peter Karow a cooperative development of "hinted" outline font technology, but Karow demurred. Imagen then adopted Pratt's conic curves for outline font technology in Imagen's Document Description Language ("DDL"), briefly a competitor to Adobe PostScript. Imagen's conic-based font format remained a trade secret.

Conic Curves for fonts 3—Folio & Sun Microsystems: In 1986, Jacobo Valdés and Eduardo Martinez, who had been principal designers of hardware and software for Imagen's DDL document description language, left Imagen to start a new company, Folio, where they were joined by Michael Sheridan. In 1987–1988, Folio developed a font format based on conic curves as propounded by Pratt, and coupled the conic outlines with programs to adjust letter shapes for improved regularity when rasterized on digital grids. The technology was called "F3" (Folio Font Format). In 1988, both Microsoft and Sun Microsystems separately discussed acquiring Folio and F3 font technology. Microsoft wanted font technology for OS/2. Sun wanted font technology for the "Sparcintosh," a prototype personal computer combining Sun's SPARC processor and Unix-based Solaris operating system in a mass-market personal computer. Sun acquired Folio in 1988 but later halted the Sparcintosh project. F3 was then combined with Sun's NeWS workstation window system to make the "NeWSprint" printing system, and a "Sparcprinter" was released in 1990. Sun published the F3 Font Format specifications in 1991,[29] but Folio's plans to broadly license the F3 technology and promote it as a Unix standard were later quashed by Sun executives.

Splines

In aircraft lofting and ship design, "splines" were thin, flexible slats of metal or wood that could be laid out on a large drafting table and constrained at selected points by weighted "ducks" or "rats," to define smooth curves in drafting aircraft forms or ship hulls. The constrained splines formed smoothly continuous curves that designers could trace and transfer to other objects or media.

B-Splines: In 1946, Isaac Jacob Schoenberg adopted the term "spline" for mathematical functions that approximate the smoothing behavior of physical splines. These functions are composed of sections of polynomial curves, generally quadratic or cubic, that are joined smoothly. In computer graphics, mathematical splines need relatively few data points to describe smooth, free-form curves that can be precisely reproduced, so computer-calculated splines replaced physical splines in drafting. From the 1960s onward, the computational and interactive design features of splines became popular in computer graphics.[31] Schoenberg's term for "Basis spline" was later shortened to B-spline in the literature, and the generalized term "spline" began to be used for many kinds of shape-defining mathematical curves, not all of which are "splines" in the sense Schoenberg used the term. [note 16]

I. J. Schoenberg, erudite innovator of mathematical splines, died in 1990, shortly before Apple's 1991 launch of TrueType technology with quadratic B-spline contours in the fonts of Macintosh System 7. Microsoft launched TrueType with fonts in 1992 in Windows 3.1. Had Schoenberg known that thousands of B-spline-based fonts would be used on billions of smartphones and digital devices, he might have added them as a new chapter of his 1983 book, *Mathematical Time Exposures*.

Quadratic B-splines—Apple: Apparently inspired by the spline-based DRAW program on the Xerox Alto, Apple developed interactive spline-based drawing software for its first Alto-like computers: LisaDraw on the Lisa computer in 1983 and MacDraw on the Macintosh computer in 1984. Both Apple Draw applications were programmed by Mark Cutter using quadratic B-splines, which according to Casselman are equivalent to quadratic Bézier curves, both forming parabolic arcs.[42]

In 1987, Apple's vice president for technology, Jean-Louis Gassée, initiated a project to develop Apple font technology for screen and printer. Gassée objected to the high royalties Apple was paying Adobe for the PostScript language and Adobe Type 1 fonts, and objected to Adobe's tight control of both technologies. As if following General Electric CEO Jack Welch's maxim, "Control your own destiny or someone else will," Gassée wanted to develop font technology in-house at Apple, and in the summer of 1987, Apple hired away Imagen's lead engineer for font technology, Sampo Kaasila. That move, however, prompted Imagen to warn Apple against misappropriation of Imagen trade secret font technology. That may have been a factor in Apple's choice of quadratic B-splines for font technology, instead of conics, which Imagen had used. Faster rasterization of quadratic curves presumably also influenced Apple's choice. Apple's quadratic B-splines took around two-thirds of the time to rasterize as did Adobe's cubic Bézier curves, but usually required a greater number of defining points on and off the curves (see Figure 11).

After the Seybold 1989 Font Wars announcements, proponents of Adobe fonts criticized TrueType because its quadratic B-spline outlines needed more curve-defining points than Bézier curves, resulting in bigger file sizes. Another criticism was that quadratic B-splines do not provide continuity of curvature at arc joins, allegedly impairing visual smoothness, whereas PostScript fonts provided continuity of curvature and smoothness of shapes.

For the initial release of TrueType fonts in Windows 3.1, Microsoft strove to demonstrate that its TrueType fonts had the same or greater visual and technical quality than Adobe's industry-leading PostScript fonts. Microsoft licensed high-resolution Ikarus outline font data from Monotype and hired a team of experienced type designers, font technicians, and typographic consultants to assist Monotype in font production and quality assurance. Microsoft later sought further improvements in TrueType technology, so a team of engineers led by Eliyezer Kohen rewrote Microsoft's TrueType rasterizer to improve reliability and efficiency.

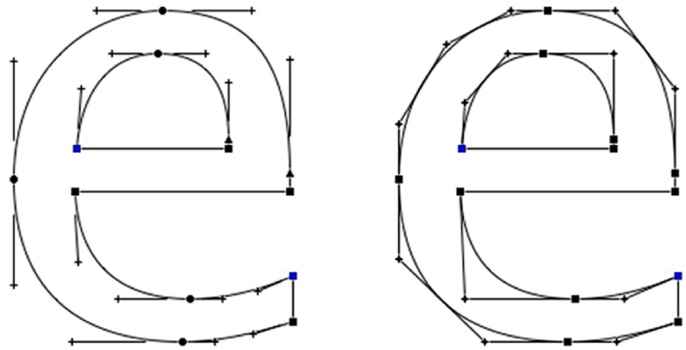


Figure 11. "e" in Bézier cubic curves with off-curve "handles" and quadratic splines with polygonal hull. Courtesy of Kris Holmes.

Cubic Curves and Splines

Ancient Greek mathematicians analyzed geometric problems equivalent to cubic equations but did not discover general solutions. Around 1100 CE, the Persian mathematician Omar Khayyam (of Rubaiyat fame), familiar with Greek texts on conics that had been translated into Arabic, demonstrated methods of solving cubic equations by intersections of conic curves. In the Italian Renaissance, mathematics flourished as Tartaglia, Cardano, and Ferrari (not the car) developed general methods for solving cubic equations. They did not, however, apply cubic curves to letter forms.

In the 20th century lead-up to the Font Wars, cubic curves used for font outlines were based on mathematical theories of interpolation and approximation, published in the late 19th and first half of the 20th century. Among various kinds of cubic curves used for fonts in the 1960s to 1990s, there were Hermite cubic curves, natural cubic splines, cubic B-splines, and Bézier cubic curves. Several such curves were discussed by Forrest[16,36] and by authors in *Computer Aided Geometric Design*,[31] including Barnhill and Riesenfeld,[32] although the curves were not applied to fonts. [note 17]

Philippe Coueignoux and, independently, Peter Karow, appear to have been the earliest to develop fonts based on cubic curves.

Cubic splines—Coueignoux at MIT: Philippe Coueignoux developed cubic spline fonts in his 1973 M.S. thesis in Electrical Engineering at MIT: "Compression of type faces by contour coding."[33] In his research, begun early in 1972, Coueignoux based letter outlines on cubic spline

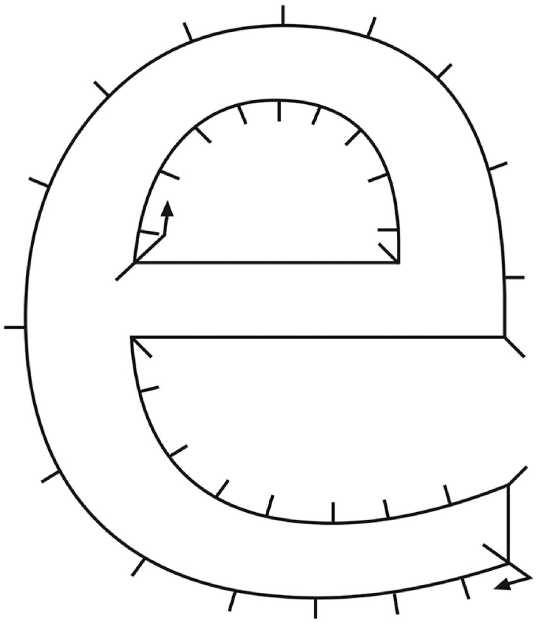


Figure 12. “e” in Ikarus cubic spline format. Courtesy of Kris Holmes.

methods published by De Boor and Rice in 1968.[34] Coueignoux’s cubic spline fonts efficiently compressed font data for storage, rasterization, and output in digital typesetting, but were not initially adopted by the typesetting industry because the computers of digital typesetters in 1973 were not powerful enough to rasterize cubic spline fonts at speeds competitive with raster-based typesetters. After receiving his Ph.D. at MIT, Coueignoux taught in France, where in 1978, his student Marc Hourdequin devised a fast rasterization technique for circular arc outline characters. In 1981, another Coueignoux student, Marc Bloch, designed a digital typesetter with fast rasterization of circular arc outline characters. Neither research method was commercialized.

Hermite cubic curves—Karow at URW: In late 1972, Peter Karow began encoding character outlines with spline curves, eventually adopting spline algorithms published in 1973 by Helmuth Späth, based on an 1877 paper on interpolation theory by Charles Hermite. For precision, Ikarus used an extremely high font resolution grid of 15,000 x 15,000 units per “em”—the virtual square within which most characters in a font are positioned. Ikarus outlines can be converted to other curve formats and can be rasterized at discrete resolutions for digital typesetting. Hermite cubic curves enable geometric transformations, for which Karow developed a repertoire of programs for rotating, mirroring, narrowing, stretching, slanting, interpolating, and contouring character outlines (see Figure 12).[14,15]

Using Ikarus, the URW firm digitized a “library” of outline fonts, reaching around 2000 fonts by the mid-1980s. Technology firms including Adobe, Apple, Microsoft, and Sun had the technical expertise to invent or acquire font technologies but initially did not have typographic expertise to create high-quality digital outline fonts from scratch, so they often purchased outline font data from URW, delivered in Ikarus formats, usually circular arc and vector.

The URW firm also provided font technology expertise, assisting Apple in developing software programs to convert from Ikarus format to Apple TrueType. Apple sent engineers to Hamburg to confer on the process and programming, and URW built TrueType conversion capability into Ikarus, including in an Ikarus version for Macintosh.

Cubic curves—Baudelaire at Xerox PARC: At Xerox PARC in 1974, Patrick Baudelaire used natural cubic splines in “Fred,” an interactive font development program on the Xerox Alto personal workstation.[35] A Fred user manipulated splines to define character outlines; from those, a second tool called PREPRESS rasterized bitmaps that a user could then bit-edit by hand, to improve the aesthetic quality, and make bitmap fonts for the Alto screen or for laser printers. Baudelaire also used splines in his “Draw” program for users to make and manipulate curves in drawings on the Alto screen, inspiring later “draw” programs on other workstations and computers. Xerox later used Fred and modifications in producing bitmap fonts for commercial laser printers.

Bernstein-Bézier cubic curves: The curves of contours in Adobe PostScript fonts were based on mathematics developed over the course of a century. In 1885, Karl Weierstrass proved an important theorem in approximation theory. In 1912, Sergei Bernstein published a “constructive” proof of Weierstrass’s theorem.

Bernstein’s approximation method was seldom used in pure mathematics but became widespread in computer graphics through the work of Paul de Faget de Casteljau, Pierre Bézier, A. R. Forrest, and others. In 1959, de Casteljau, a mathematician at the Citroën company, used Bernstein approximation to describe curves and surfaces for automobile design. Fenders, but not fonts. Citroën did not permit de Casteljau to publish his research, so his method was not known until years later, and it seems unclear how he came to adopt the mathematics of Bernstein's 1912 paper. De Casteljau devised an elegant, recursive algorithm to calculate Bernstein-based curves, and after the algorithm leaked out and became popular in computer graphics research, it was named de Casteljau's Algorithm.[36]

In the early 1960s at the Renault company, Pierre Bézier independently developed a method of describing curves and surfaces for computer-aided design and manufacturing. Again, fenders, not fonts. Bézier did publish his research, and in 1972, A. R. Forrest[37] showed that Bézier's curves could be expressed as Bernstein's polynomials. Forrest's publication helped popularize Bézier curves, as they came to be called. In the 1970s and 1980s, Forrest visited, consulted, and lectured at several centers of computer graphics research, including General Motors, University of Utah, Evans & Sutherland Computer Corporation, and Xerox PARC. [note 18]

Bézier Cubic Curves—Xerox PARC and Adobe PostScript: At the 1982 SIGGRAPH conference, John Warnock and Douglas Wyatt of Xerox PARC described a device-independent imaging model in which characters of fonts were treated as "shapes" and the contours of shapes were described with Bézier curves.[38] After Warnock left PARC and co-founded Adobe Systems in 1982, Bézier curves became the general method of describing shapes of characters in Adobe's PostScript graphics language and fonts. [note 19]

Warnock later wrote that Adobe chose Bézier curves for PostScript because, "They are simple to describe, easy to render on a computer, and flexible enough, when put together in combination, to describe the most complex shapes in the graphic arts industry."[39]

Sergei Natanovich Bernstein died in 1968, never knowing that applications of his 1912 paper on interpolation would eventually be used by billions of readers and writers of messages and text shot through the Internet and viewed on smartphones around the world. Presumably, he would have been astonished but pleased.

Bézier Cubic Curves—Knuth and Metafont at Stanford: In the late 1970s, Donald Knuth at Stanford began work on Metafont, a computer language for generating digital type. Knuth initially programmed Metafont using a scribal handwriting metaphor in which virtual pens traced spline-defined paths of letter strokes on a digital grid. A cyber-scribe wrote a computer program specifying the positional coordinates of spline paths and the parameters of a virtual pen that traced the paths of the strokes that made a character, leaving a trail of pixels in a grid on a virtual plane. This version came to be called Metafont 79 ("MF79") to distinguish it from the later version, Metafont 84 ("MF84").

By changing the positional coordinates of the spline paths, the Metafont 79 programmer changed the skeletal structure, width, and proportions of a letter or characters. By altering pen parameters—such as size, shape, and orientation—the programmer changed the weights and thick-thin modulations of letter strokes. Specification of the scale and metrics of the virtual grid determined the resolution of the eventual bitmap font.

After some experimentation with curves and discussions with the "Metafont for Lunch Bunch," Knuth revised Metafont to use Bernstein-Bézier cubic curves to describe pen paths and to trace outline contours of shapes. These, and other changes, produced MF84. In The METAFONTbook (describing MF84), Knuth credited the mathematics of Metafont curves to "Sergeĭ N. Bernshteĭn" (a transliteration from Cyrillic orthography instead of the more common "Bernstein"). He noted that the curves were also called Bézier curves, and showed the recursive algorithm of de Casteljau for drawing such curves.[40] [note 20]

Knuth published the Metafont source code[41], and Casselman[42] wrote of it: "For anyone who wants to see what attention to detail in first class work really amounts to, this is the best resource available."

Mathematical Ideas and Moore's Law

In digital typography, the mathematics of font formats progressed from equations of zero degree (points, as bitmaps), to one degree (lines, as polygons), two degrees (as quadratic curves and splines), three degrees (as cubic curves and splines).

By 1973, most of the mathematical ideas for font formats had been manifested in digital typography, either in research or in commerce. There were bitmaps (Hell Digiset); polygons (Seaco); circular arcs (Rockwell Metro-set); cubic B-splines (Coueignoux); cubic Hermite curves (Karow-Ikarus). It would be another dozen years, however, before cubic curve font formats could be processed on affordable personal laser printers.

Following Moore's Law, progress in semiconductors radically reduced the costs of digital font computation. In 1973, a 600 dots-per inch digital typesetting system cost more than a million dollars (in 1985 dollars), weighed two tons, and filled a large room. By 1985, computer memory prices had fallen to 1/400th of their 1973 cost, and microprocessor speeds had increased to at least a hundred times faster than in 1973. The Apple LaserWriter launched in 1985 could rasterize Bézier cubic outline fonts, cache the bitmap characters, and write a full page of text and image to page buffer memory in a 300 dots-per-inch printer sitting on a desktop and costing less than $7000. Without steep drops in cost of computer memory and processors, desktop publishing would not have become possible. [note 21]

Visual and Mathematical Aesthetics

Technical factors like font file size, speed of rasterization, graphical transformability, numerical stability, and efficiency of font production dominated discussions of font technology, but another factor, less often expressed, was mathematical aesthetics.

In curve-based outline formats, contours are usually described by several curve segments joined together. The outer contour of a capital "O," for example, is usually composed of at least four curve segments, one for each quadrant. Depending on the shape of the "O" and the nature of the curves, more segments may be used. (See lower-left "O" of this issue's cover illustration.)

Quadratic font formats, such as TrueType and F3, join segments of conic arcs together to make more complex curves. Where two arcs join, continuity of slope (tangency) is usually imposed in order to make a visually smooth transition, that is, without pimples or dimples, no cusps or dents. Cubic font formats, such as PostScript, join cubic curve segments to have continuity of curvature at joins, in addition to continuity of tangency. This means that the curve segments have the same radius of curvature at the join. Continuity of curvature is important for smoothness of three-dimensional (3-D) surfaces, like automobile bodies, but for 2-D letter shapes, continuity of tangency seems to be sufficient for smoothness as perceived by readers. Pratt[29] described a method of achieving continuity of curvature with conics. [note 22]

In the frenzied competitions of the Font Wars, font technology developers, type designers, and typography experts trusted their eyes but did not conduct scientific vision tests of whether visual smoothness needed mathematical continuity of curvature.

Without evidence from vision science but with a plethora of typographical opinion, mathematical taste emerged as an influential factor in font formats. Continuity of curvature is an elegant mathematical quality for letter contours, even if readers do not notice or appreciate it. Other characteristics involving mathematical taste included the elegance of de Casteljau's recursive algorithm for rendering Bernstein curves, the transformation of ancient Apollonian conics into modern algebraic algorithms, the flexibility of Schoenberg's B-splines. Warnock and Knuth chose cubic Bézier curves, Pratt chose conics, and Apple chose quadratic B-splines. Apart from technical claims, preferences for one form of outline curve over another echo the words of G. H. Hardy.

"A mathematician, like a painter or a poet, is a maker of patterns. If his patterns are more permanent than theirs, it is because they are made with ideas. The mathematician's pattern, like the painter's or the poet's, must be beautiful; the ideas, like the colours or the words, must fit together in a harmonious way."[43]

In the long view of history, the ideas manifested in the Font Wars were not a big bunch of mumbo–jumbo after all. They were fundamental concepts discovered and explored over decades, centuries, or millennia. Although these ideas exhibited potential utility for typography, the technologies of their times were not able to realize them. It was not until the mathematics of shapes became powered by electronic computing, enabling algorithmic expression and visual rendering of the small but significant forms of letters and characters—themselves shaped over centuries and millennia by hand and eye—that those ideas changed the course of literate history.

ACKNOWLEDGEMENTS

I wish to thank the readers who generously commented on multiple drafts: Karl Berry, Kris Holmes, Peter Renz, David Walden, and an anonymous reviewer. I am responsible for any remaining errors.

REFERENCES

1. M. Marshall, "Microsoft/Apple deal leaves industry reeling," *InfoWorld*, vol. 1, p. 101, Sep. 1989.
2. L. Flynn and M. Marshall, "Users caught in font war cross fire," *InfoWorld*, vol. 1, p. 101, Sep. 1989.
3. P. Lewis, "The fallout from the font wars," *New York Times*, Oct. 1, 1989, sec. 3, p. 13.
4. E. Eisenstein, *The Printing Press as an Agent of Change*. Cambridge, U.K.: Cambridge Univ. Press, 1980.
5. P. M. Romer, "Endogenous technological change," *J. Political Economy*, vol. 98, no. 5, Part 2, pp. S71–S102, 1990.
6. *Visible Language: Inventions of Writing in the Middle East and Beyond*, C. Woods, E. Teeter, G. Emberling, Eds. Chicago, IL, USA: Oriental Inst. Museum Publications, No. 32, The Oriental Institute of the University of Chicago, 2015, pp. 33–50.
7. M. Civil, "A hymn to the beer goddess and a drinking song," in *Studies Presented to A. Leo Oppenheim*, R. D. Biggs and J. A. Brinkman, Eds. Chicago, IL, USA: Univ. Chicago Press, 1964, pp. 67–89.
8. J. André, "Histoire technique des fontes numériques," in *Histoire de L'Écriture Typographique,—le XXième Siècle, Tome II/II*, J. André, Ed. France, Gap: Atelier Perrousseaux, 2016, pp. 114–143. [Review in English: C. Bigelow, "Review and summaries: The history of typographic writing—the 20th century, vol. 2 (ch. 1–5), Jacques André Ed.," *TUGboat* vol. 38, no. 2, pp. 274–279, 2017, [Online]. Available: https://tug.org/TUGboat/tb38-2/tb119bigelow.pdf]
9. B. Warde, "Twinkle in Andromeda," in *Alphabet 1964*, R. S. Hutchings, Ed. Birmingham, U.K.: Kynoch Press, 1964, pp. 78–83.
10. G. O. Walter, "Typesetting," *Sci. Amer.* vol. 220, no. 50, pp. 60–69, May 1969.
11. "Oral history of Gary Starkweather," interviewed by Gardner Hendrie, Computer History Museum, 2010, CHM Reference number X5848.2010.
12. R. F. Sproull, "The Xerox Alto publishing platform," *IEEE Ann. History Comput.*, vol. 40, no. 3, pp. 38–54, Jul./Sep. 2018.
13. J. Johnson *et al.*, "The Xerox Star: A retrospective," *Computer*, vol. 22, no. 9, pp. 11–26, 1989.
14. P. Karow, *Font Technology: Description and Tools*, Berlin, Germany: Springer-Verlag, 1994.
15. P. Karow, *Digital Typefaces: Description and Formats*. Berlin, Germany: Springer-Verlag, 1994.
16. A. R. Forrest, "Curves and surfaces for computer-aided design," Ph.D. dissertation, Univ. Cambridge, Cambridge, U.K., 1968.
17. J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Syst. J.*, vol. 4, no. 1, pp. 25–30, 1965.
18. A. V. Hershey, "Calligraphy for computers," U.S. Naval Weapons Laboratory, Dahlgren, VA, USA, Rep. No. NWL-TR-2101, Aug. 1967.
19. M. V. Mathews, C. Lochbaum, and J. A. Moss, "Three fonts of computer-drawn letters," *J. Typograph. Res.*, vol. 1, no. 4, pp. 345–356, 1967.
20. J. Condon, B. Kernighan, and K. Thompson, "Experience with the Mergenthaler Linotron 202 phototypesetter, or, how we spent our summer vacation," Bell Laboratories, Computing Science Technical Memorandum. January 6, 1980. Reconstructed and published, 2013. [Online]. Available: https://www.cs.princeton.edu/~bwk/202/
21. J. E. Bresenham, "A linear algorithm for incremental digital display of circular arcs," *Commun. ACM*, vol. 20, no. 2, pp. 100–106, 1977.
22. B. K. P. Horn, "Circle generators for display devices," *Comput. Graph. Image Process.*, vol. 5, no. 1, pp. 280–288, 1976.
23. M. D. McIlroy, "Best approximate circles on integer grids," *ACM Trans. Graph.*, vol. 2, no. 4, 237–263, 1983.
24. P. Karow, *Digital Typography & Artificial Intelligence*, Adobe/Dutch Type Library, 2013.
25. W. Schreiber, "Outline coding of typographical characters," Printing Industries of America, 1987. [Abstract at: https://www.printing.org/taga-abstracts/outline-coding-of-typographical-characters. Accessed Jun. 6, 2019]
26. M. Carter, "The Digital Typefoundry," *Visible Lang.*, vol. 50, no. 2, pp. 26–37, 2016.

27. J. Flowers, "Digital type manufacture: An interactive approach," *IEEE Comput.*, vol. 17, no. 5, pp. 40–48, May 1984.
28. M. L. V. Pitteway, "Algorithm for drawing ellipses or hyperbolae with a digital plotter," *Comput. J.*, vol. 10, no. 3, pp. 282–289, 1967.
29. V. Pratt, "Techniques for conic splines," *ACM SIGGRAPH Comput. Graph.*, vol. 19, no. 3, pp. 151–160, Jul. 1985.
30. Sun Microsystems, *F3 Font Format Specification*, Sun Microsystems, Santa Clara, CA, USA, 1991.
31. R. Barnhill and R. Riesenfeld, Eds. *Computer Aided Geometric Design*. New York, NY, USA: Academic, 1974.
32. R. Barnhill and R. Riesenfeld, Eds. "B-spline curves and surfaces," in *Computer Aided Geometric Design*. New York, NY, USA: Academic, 1974.
33. P. J. M. Coueignoux, "Compression of type faces by contour coding," M.S. Thesis, Massachusetts Inst. Technol., Cambridge, MA, USA, Jan. 1973.
34. C. De Boor and J. R. Rice, "Least squares cubic spline approximation, II-Variable knots," Comput. Sci. Dept., Purdue Univ., West Lafayette, IN, USA, Tech. Rep. CSD TR 21, 1968.
35. P. Baudelaire, "The Xerox Alto font design system," *Visible Lang.*, vol. 50, no. 2, pp. 12–25, 2016.
36. W. Boehm and A. Müller, "On de Casteljau's algorithm," *Comput. Aided Geometric Des.*, vol. 16, no. 7, pp. 587–605, 1999.
37. A. R. Forrest, "Interactive interpolation and approximation by Bézier polynomials," *Comput. J.*, vol. 15, no. 1, pp. 71–79, 1972.
38. J. Warnock and D. K. Wyatt, "A device independent graphics imaging model for use with raster devices," *ACM SIGGRAPH Comput. Graph.*, vol. 16, no. 3, pp. 313–319, 1982.
39. J. Warnock, "Simple ideas that changed printing and publishing," *Proc. Amer. Philosophical Soc.*, vol. 156, no. 4, pp. 363–378, 2012.
40. D. E. Knuth, *The METAFONTbook.* Reading, MA: Addison-Wesley, 1986.
41. D. E. Knuth, *Computers and Typesetting, Volume D: METAFONT: The Program.* Reading, MA, USA: Addison-Wesley, 1986.
42. B. Casselman, "From Bézier to Bernstein: Feature Column from American Mathematical Society," 2008. Accessed: Jun. 2019. [Online]. Available: http://www.ams.org/publicoutreach/feature-column/fcarc-bezier
43. G. H. Hardy, *A Mathematician's Apology.* Cambridge, U.K.: Cambridge Univ. Press, 1940.

Charles Bigelow is a Scholar in Residence at the Cary Graphic Arts Collection and former Cary Distinguished Professor at Rochester Institute of Technology, Rochester, NY, USA. He is a MacArthur Foundation Prize Fellow and was a professor of digital typography at Stanford University from 1982 to 1995. With Kris Holmes, he designed the Lucida fonts distributed by Adobe, Apple, Bell Labs, Microsoft, and Sun Microsystems. He was a typographic advisor to most of those same firms and others during the Font Wars. Contact him at lucidafonts@gmail.com.

The Font Wars, Part 2

Charles Bigelow
Cary Graphic Arts Collection,
Rochester Institute of Technology

***Abstract*—In the early 1980s, letters rasterized from outline fonts at low and medium resolutions had irregular shapes. The lower the resolutions, the greater the irregularities, and the more that typographers criticized the type quality. PostScript fonts, launched in 1985, regularized medium resolution type with secret ingredients coyly called "hints" by Adobe. This prompted competing inventions of font regularization using techniques variously called "instructions," "delta exceptions," "procedures," "intelligence," and other terms sounding more like hi-tech snake oil than science. Further research and open publication, however, revealed their connections to traditional aesthetics of letter symmetry as well as to modern signal processing, pattern recognition, and psychophysics, thus expanding our understanding of typography in digital culture.**

BACKGROUND

Simplicity of Forms

WHEN EUROPEAN TYPOGRAPHY was invented in the 15th century, it imitated Latin handwriting, which used relatively simple, repeated letter forms. Vision scientists have proposed that the shapes of written forms evolved for ease of recognition and reading. Changizi and Shimojo, for instance, reasoned that the visual simplicity of strokes and joinery was the major influence on character shape.[1] Using a different approach, Pelli *et al*. combined a mathematical definition of character complexity with laboratory studies of character recognition and found that simple forms are recognized more efficiently than complex forms.[2]

At least since the Renaissance, however, handwriting experts declared ease of writing to be the major influence on letter shapes. The first printed handwriting manual, Arrighi's *Operina* of 1522, taught "littera corsiva" (running script) to Renaissance cognoscenti.[3] Alfred Fairbank, a 20th-century disciple of Arrighi, declared regularity, rhythm, and speed to be the desirable qualities of handwriting, while also acknowledging the importance of legibility.[4] Handwriting teacher and lettering artist, Gerrit Noordzij has argued that the characteristic forms and rhythms of Western handwriting derive from a fundamental unit, the "stroke" of the pen.[5] The pen was not the only

Digital Object Identifier 10.1109/MAHC.2020.2971745
Date of current version 6 March 2020.

Figure 1. Roman capitals based on the Trajan inscription in Rome (113 CE). Revived, brush-painted and hand-chiseled in slate by Edward Catich in 1957, showing "SPQR" (Senatus Populusque Romanus) with example letters and their basic brush strokes. Courtesy of the RIT Cary Graphic Arts Collection.

tool. Edward Catich analyzed Roman inscriptional letters and demonstrated that they were handwritten with combinations of brush strokes before being chiseled into stone (see Figure 1).[6]

Symmetries in Handwriting: Roman, medieval, and Renaissance scribes imparted regularity, repetition, rhythm, alignment, and symmetry to the patterns of text. Some symmetries in written forms were inherited from ancient Greek inscriptional styles, while others evolved in medieval scriptoria. Handwritten and typographic letter symmetries are visual but not geometrically exact. They include reflections such as b and d, p and q, rotations such as b and q, d and p, n and u, and translations such as n and m, v and w, V and W. [note 23]*

Symmetries in Printing: The invention of printing separated the shaping of letters from the motions of writing. In printing types, letters were prefabricated. For hundreds of years, the letter forms were first cut in steel by skilled artisans, then struck into matrices, and then cast as individual, three-dimensional shapes. Though technically free of demands for on-the-fly manual dexterity, early printing types nevertheless imitated the forms and rhythms of scribal handwriting and inscriptional letter carving. A few early typefaces, particularly the roman types of Nicolas Jenson, and the roman and italic types cut by Francesco Griffo for Aldus Manutius, are still viewed as early masterpieces in form and regularity, appealing to visual sensibility as well as functionality.

*References of the form [note n] continue the extended notes at https://history.computer.org/annals/dtp/fw.

Visual Regularity

The rapid progress in computer graphical descriptions of the shapes of single letter forms was based on decades or centuries of pure and applied geometry. Visual regularity in typography, however, emerges from combinations of forms.[7] Mathematical studies of spatial regularity were concerned mainly with abstract symmetry, such as one-dimensional (1-D) band patterns, 2-D tilings, or 3-D space fillings. Although mathematical analyses of symmetry have applications to crystallography and related fields, they have generally not been applied to typography, and in any case, have not addressed the information-signifying patterns of semi-symmetrical symbols in text.

A line of text is a rectangular band of symbols, and a column of text is a 2-D tiling in which the tiles are images of letters or characters. Typefaces in texts have periodic structures and partial symmetries of forms, including translations, rotations, and reflections, but their graphical patterns and repetitions of the symbols are determined by linguistic and orthographic rules, not purely mathematical structures.

Despite the long history of regularity in typography, it remains unclear why readers prefer regularity and symmetry. Early in the 20th century, anthropologist Franz Boas observed that in indigenous crafts such as basketry and weaving, "The judgment of perfection of technical form is essentially an esthetic one."[8] More recently, Dorothy Washburn and Donald Crowe have explored the mathematical, anthropological, and aesthetic role of patterns in culture.[9] [note 23]

Regularity and Resolution: Loss of regularity in raster displays was attributed to low resolutions. In the Font Wars, there were three technical and commercial levels of raster resolutions. CRT screens had low resolutions around 72 pixels per inch (ppi), initially corresponding to the

Figure 2. Low resolution rasterizations of capital letter H at three sizes, showing different thicknesses of bitmap stems compared to original outlines. Courtesy of Beat Stamm.

Figure 3. Rasterizations of capital letter H at seventeen incrementally increasing raster sizes, showing stem thickness variations and irregular losses of serif and crossbar features and symmetry. Courtesy of Beat Stamm.

typographic point system of 72 points per inch, but later increased to around 100 ppi. Commercial laser printers had medium resolutions, initially around 240 to 300 dots per inch (dpi), doubling to 600 dpi by the end of the 20th century. CRT and laser typesetting machines for professional publishing had high resolutions, initially around 600 dpi but increasing to around 1200 dpi by the end of the 20th century.

On low-resolution CRT screens, letter forms at text sizes are pointillist dot aggregations recognizable as generic letters but not as traditional printing types. At coarse resolutions, staircase patterns replace smooth curves and diagonal lines.

At middle resolutions, as in common laser printers, letter forms look more like familiar typefaces, albeit with some stair-casing, but other irregularities may be noticeable. Stem thicknesses may differ between and within letters; some letters may be too dark, others too light (see Figure 2).

At different raster sizes of the same outline letter, stem thicknesses may vary, thin features like serifs and crossbars may erratically disappear and reappear, and letter symmetry may be present or lost (see Figure 3).

Alignments of letters may bounce up and down on the baseline or wobble along the x-height line. The x-height is a basic dimension of lowercase letters (see Figure 4). [note 24]

Analysis of how digital resolution affects type forms has three main approaches. One approach is to calculate the sampling rate needed to adequately capture typeface images. Another is to calculate the acuity limit of the human visual system because type resolution presumably does not need to exceed what readers can perceive. A pragmatic method is improvement over time. Digital equipment manufacturers kept increasing output resolutions until users stopped complaining about inferior type quality.

Sampling Rates

Just as speech and music can be analyzed as audio frequencies, so letter forms can be analyzed as spatial frequencies—periodic alternations of colorings in a plane. In typography, usually black forms alternate with white background spaces. A line of text is a pattern of black strokes alternating with white spaces, in what has been termed “stroke frequency.”[10] Alternation of vertical strokes is a low frequency in text that should be kept in phase with the raster grid. Higher spatial frequencies of type are those needed to render thin elements such as serifs and hairlines, as well as subtle adjustments of alignments called “overshoots.” The sharp, high-contrast edges admired by typographers require yet higher frequencies.

In the 1920s at Bell Laboratories, Harry Nyquist proved that the sampling rate needed for faithful capture and reconstruction of the information in a continuous signal must be at

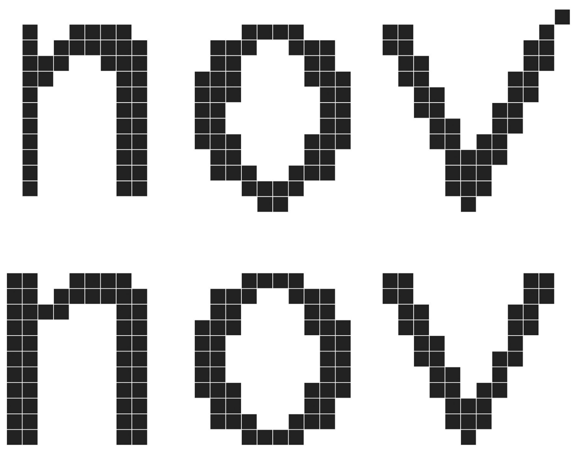


Figure 4. Above: Irregularity and misalignment in unconstrained rasterization of lowercase letters n, o, v. Below: the same letters hand-edited to improve regularity and alignment of x-height and baseline. Courtesy of Kris Holmes.

least twice the highest frequency in the signal. Hence, at a fixed sampling rate, the highest frequency that can faithfully be reproduced is slightly less than half the sampling frequency. The "Nyquist rate" denotes the minimum sampling rate for faithful capture of a signal, and "Nyquist limit" denotes the maximum frequency that can be faithfully captured at a given sampling rate. (The sampling theorem is also called the Nyquist-Shannon theorem.)

Device-independent letter outlines like those in PostScript Type 1 and TrueType font formats can be scaled to any arbitrary size and rasterized at any given resolution. For bi-level (black-white) raster devices like laser printers and early bitmap displays, the sampling rate is generally assumed to be the resolution, such as 300 dots per inch on a laser printer or 72 pixels per inch on a CRT screen. The resolution of 300 dpi used on Xerox printers was not determined by the visual quality of output but by the laser writing resolution that could keep up with the paper transport speed of the Xerox 9700 laser printer. Resolutions greater than 300 dpi took too much time to write a page before the next page was due.

Bigelow and Day discussed spatial sampling in a review of digital typography.[11] Digital sampling rates and type forms are analyzed and illustrated by Beat Stamm in the web site, "The Raster Tragedy."[12]

Different approaches, assumptions, and calculations have indicated that high-quality output resolutions range between 700 and 1200 bi-level pixels per inch. Peter Karow discussed and illustrated the effects and problems of font rasterization and resolution, in particular with Times Roman examples.[13]

Sampling and Visual Acuity: Sampling occurs at the encoding and transmitter end of the digital type. Human vision is at the receiver end. Readers with 20/20 visual acuity (according to a Snellen eye chart) can discriminate detail as small as one minute of arc, or 30 cycles (pairs of black + white stripes) per degree of visual angle.

Reading distance is an important factor in calculating requisite resolutions, though results vary depending on different assumptions. Karow[13] calculated resolutions and reading distances in relation to perceived type quality and found that at a reading distance of 25 centimeters (10 inches), a raster resolution of 700 dots/lines per inch provided adequate type quality, assuming visual acuity of one minute of arc. At a reading distance of 30 cm (12 inches), 600 dpi provided adequate quality, and at 40 cm (16 inches), often used as a standard in reading research, an adequate resolution for perceived quality would be around 450 dpi.

By a different method, Stamm calculated that the sampling resolution needed to render Times Roman adequately is around 1024 × 1024 pixels, though that calculation does not take into account reading distance and visual acuity.[12] [note 24]

Digital Output Resolutions: The resolutions of digital typesetters, image setters, and platemakers increased over time, from around 600 x 600 dots per inch to 1200 x 1200 and then 2400 x 2400 dots per inch. Karow[13] illustrated resolution with enlargements of output compared to actual size output of Times Roman at resolutions of 300, 600, 1200, and 2400 dots per inch. Karow concluded that a resolution of 600 dpi gives adequate print quality for desktop publishing and that at resolutions of 1200 dpi and above, pixel variations in stroke width are imperceptible and the font quality approaches a maximum. [note 24]

Psychophysical Analysis: A different approach to studying type forms stems from psychophysical research pioneered by Fergus Campbell and John Robson at Cambridge University, who studied the human visual perception of sinusoidal gratings—alternating stripes of varying spatial frequency and contrast. They found a nonlinear response to spatial frequencies, depending on the contrast between the light peaks and dark troughs of the stripes. At low contrast, the human visual system was most sensitive to stripes with spatial frequencies around 3–6 cycles per degree of visual angle, but at high frequencies, around 50 cycles per degree of visual angle, the stripes can only be distinguished when the gratings are at maximum contrast. [note 25]

As a rough approximation, common typefaces at 12 point size have stem frequencies around 4–5 cycles per degree at a reading distance of 12 inches, while fine details of letter forms may have spatial frequencies up to 50 cycles per degree under ideal conditions. For centuries, letterpress typography was high contrast because of the technology—

relief surfaces and heavy black ink impressed into paper—but high contrast also enabled fine details to be perceived.

Psychophysical reading studies have used Fourier analysis, spatial frequency filtering, and noise masking, among other techniques, to investigate whether, and how, frequency-tuned channels in the visual systems are involved in letter recognition and continuous reading.

Gordon Legge *et al.* found that low-pass filtered text remains legible down to a bandwidth around 2 cycles per letter.[14] This low-pass filtered text looks blurry but can be read about as fast as sharp-edged text, suggesting that the sharpness of letter edges and clarity of fine details of traditional printed type contribute to the aesthetic appearance of type at text sizes but may not contribute greater functionality in measurable reading speed. Nevertheless, readers apparently appreciate high contrast forms. There appears to be a difference between the spatial frequencies used in reading text at typical book and newspaper sizes, and recognizing letters at large sizes for which high spatial frequencies do contribute to the recognition of letter forms.[10] [note 25]

Hand Editing of Pixel Patterns for Visual Amelioration: Raster devices in the era of the Font Wars had fixed resolutions, so type had to be rasterized to fit output resolutions. At low and medium resolutions, irregularities of text could be ameliorated by designers' hand-editing bitmaps. Editing cannot increase the resolution but can make text look more regular and symmetrical, particularly through visual adjustments and equalizations of stems, bowls, serifs, and alignments. Greater regularity of text contributed to reader perception of higher quality display or output.

Hand-editing at each font size and resolution substantially increased the labor of making digital fonts because the amount of editing increased proportionally with linear size and/or linear resolution. A typeface comprising bitmap fonts in eight typographic point sizes, say 6, 8, 10, 12, 14, 18, 24, 36 point, with a 128 character repertoire would require 8 sizes $\times$ 128 characters $=$ 1024 glyphs. The larger sizes with proportionally greater perimeters would require more editing than the smaller sizes. A family of four fonts, roman, italic, bold, bold italic, in the same range of sizes, would require human editing of 4096 glyphs. Hence, manufacturers of digital typesetters, and later of laser printers, and laser printer researched ways to automatically generate raster fonts with greater regularity and symmetry.

Early Software Structures and Rasterization

Unlike shape descriptions based on the long history of analytic geometry and interpolation theory, irregularity of type forms in raster fonts was a new kind of problem, to which solutions tended to be *ad hoc* and based on a few ideas. One idea was to analyze typefaces into component elements that could be parameterized and regularized as primitive elements that could then be recombined. This approach could be extended by identifying repeated elements in the characters of a font, such as the repeated arches and stems of letters m and n, and marking them but without detaching them as separate parts. These ideas relied on local adjustment of outline shapes to conform to the metrics of digital grids.

Coueignoux: 1975, Componential Analysis: In researching compression of font data for his 1975 MIT Ph.D. thesis, Philippe Coueignoux invented a componential analysis of outline fonts.[15] He decomposed characters of a font into a small number of parameterized outline components or "primitives," analogous to the basic strokes of handwriting. To compose the various, separate primitive shapes back into characters for rasterization in a typesetter, Coueignoux wrote a generative grammar as part of the font technology.

Coueignoux drew an analogy between his method and linguistic signs. Simply put, a language has meaningful components such as words, of which there are many, composed of nonmeaningful component speech sounds, such as phonemes, of which there are few. Similarly, in Coueignoux's system, a small number of nonmeaningful graphical parts were combined into a large set of recognizable characters in a font.

Repeatable parts not only compressed font data but could potentially regularize characters because the parameterized primitives were dimensionally stable and used throughout the font, thus guaranteeing precise repetition instead of the small differences in dimensions of contours that were hand-digitized or automatically fitted from rasters.

Mini-computer speeds in the mid-1970s could not compose outline characters from primitives and rasterize them at speeds competitive with raster-based typesetters that dominated the market. On the font production side, the analysis of a font into component parts entailed designer labor. Automatic pattern-recognition and analysis of characters into font features were not used in digital font production until the late 1980s and early 1990s.

Coueignoux's thesis was ahead of its time, but similar ideas were incorporated into later outline font formats such as PostScript Type 1 and TrueType. At a basic level, separable repeatable elements such as accents and diacritics could be implemented as subroutines used with multiple characters. For instance, the letters e, é, è, ë, ê can be composed of a single "e" glyph with the accents acute, grave, umlaut, and circumflex. Those same accents can also be used on other letters when appropriate, such as a, o, i, u, and y. In font design software, repeatable parts like serifs are often used during the design process, though not implemented in the final fonts. Repeatable strokes have been extensively investigated to compress the large amounts of data in Chinese and Japanese fonts.

Karow: 1979, Pattern Recognition and Local Grid Adjustment: Around 1976, Peter Karow at the URW consulting firm developed "software-scanning"—generation of rasters from Ikarus cubic Hermite outlines. Karow found the software method more efficient than rasterizing analog letter images with an electro-optical scanner, then a common way to generate raster type. In 1979, Karow began to program what he called "intelligent scaling." An Ikarus module used pattern recognition techniques to find repeated stems, bowls, serifs, and other letter features that dimensionally cluster in groups. When these were determined, the software was programmed to scale characters with local adjustments, stretching or squashing character outlines to fit them to output raster grids. This approach entailed preparation of character outlines with on-curve points at geometric *X*- and *Y*-axis maximums and minimums, which URW called "extrema." The Ikarus software could drag the extremes of outlines to fit grids and smoothly interpolate intervening points.

The Ikarus module called PA or "Passe" ran on mini-computers to generate bitmaps for the Digiset typesetting machines made by Hell in Germany and the APS typesetters of Autologic in California. The preparation and analysis were done off-line. By 1981, Ikarus intelligent scaling involved recognition and marking of seven elements: vertical stems, horizontal bars, curved bows (vertical curve contours), arches (horizontal portions of curve contours), serifs, contour extremes, and inclinations. Karow[13] listed the steps to prepare character contours for digitization on low-resolution devices.

- Channel processing—adjusting vertical stems to be exactly vertical, and horizontal bars exactly horizontal.
- Cutting and pasting serifs—using repeatable parts.
- Unitizing—adjusting widths if needed to match a unit width system.
- Symmetrizing—equalizing alignments (like baseline) and overshoots.
- Setting extreme points on contours—to exact maximum and minimum grid values in *X*- and *Y*-axes.

URW continued to develop further elements and refinements for grid fitting control [note 26] and by 1989 had incorporated them in the "Nimbus" system that could be run in printer and typesetter controllers.[13]

Software Structures and Printers

Adobe: 1985, PostScript Type 1 Fonts: By the time that Adobe developed PostScript in 1983–1984, three major ideas for regularized font rasterization had already been invented. The first idea was the description of character shapes by high-order curve outlines. The second was analysis and parameterization of repeated character elements in fonts. The third was the regularization of shapes by adjusting character outlines to raster grids.

Adobe's PostScript Type 1 font technology was a technical and commercial breakthrough because it was the first to integrate all three ideas into a software system, embeddable in printer and typesetter controllers, that could scale and rasterize fonts on-the-fly during printing.

Although the specification of the PostScript Type 1 font format seems complex, the ideas are conceptually simple. First, identify salient and graphically significant elements of characters; second, analyze their groupings and dimensions; third, label them so the font interpreter can adjust the dimensions for aesthetically acceptable rasterization. Significant elements include vertical stems and bowls, for instance on H O and m n o p, horizontal elements, such as crossbars and curved horizontals, for instance on H O T and f t o p, and horizontal alignments including the font baseline (the imaginary line that letters sit or stand on), the x-line (the imaginary line demarcating the height of lowercase letters without ascenders), and the capital line (the imaginary line demarcating the height of flat-topped capital letters). In addition, there are small overshoots where rounded or pointed Y maxima or minima points overshoot the baseline, x-line, or capital line, and those can be identified and adjusted.[16]

Type 1 character contours are based on Bézier curve segments and straight lines, so Type 1 character hints are usually labeled at the end points of curves or lines as well as situated at geometric extremes of contours to enable the grid-fitting procedures to shift the extreme points and thereby adjust contours to fit the output raster grid.

Adobe called these structural labels "hints" because the important character regularities such as stems and alignments were declared in the font data, although the fonts themselves did not contain the procedures for interpreting the declarations. The term "hints" hinted at secret ingredients without revealing details. Under competitive pressure from the Apple-Microsoft development of TrueType, Adobe published the specification of its Type 1 font technology in 1990.[16] Type 1 characters include two different kinds of structural information: one is shape information defining character contours, and the other is the label information declaring the classes of regularities to which selected features belong. [note 26]

Intellifont: In 1978, the Compugraphic corporation, then the largest manufacturer of typesetting equipment in the US, licensed an Ikarus system from URW. Tom Hawkins, Ikarus system manager at Compugraphic, extended the concept of the Ikarus Passe program and developed a font format called "Intellifont" that combined structural information with Ikarus-derived character arc and line outlines. Intellifont software regularized font outlines prior to rasterization. In the Ikarus system, the Passe module was run off-line, but Compugraphic's Intellifont was a separate module that could be incorporated into typesetter and printer controllers. Hawkins filed for a patent on "Intellifont" in 1984, and the patent was granted in 1987, with assignee Compugraphic.[17] [note 26]

Bitstream: The Bitstream firm was founded in 1981 as the first all-digital font development firm. Its start-up business was producing outline fonts for license to Original Equipment Manufacturer (OEM) firms.[18] In the mid-1980s, Bitstream began developing software for fast rasterization of Bitstream outline fonts into bitmaps. The rasterization software and fonts were marketed to OEMs under the trade names "Speedo" and "Facelift." In 1991, Bitstream donated the Speedo rasterizing software to the MIT X Consortium for use in the X Window System. Bitstream's hinting method used structural zone information to capture outline character features that could be adjusted to specified digital grids and thus achieve font regularity in bitmap output. Bitstream's approach was the subject of three patents and a SIGGRAPH'88 course by inventor Phillip Apley. [note 26]

Imagen: In September 1984, at an Association Typographique Internationale meeting in London, Michael Sheridan, director of typography at the Imagen corporation, met with Peter Karow and proposed a collaboration between URW and Imagen on a scalable, structured outline font technology. Sheridan was familiar with the Ikarus system from his prior work at Autologic. He suggested that in collaborating with Imagen, URW could also license the technology to other firms. Karow demurred, so in late 1984, Imagen independently began to develop an outline font format based on conic splines researched by Vaughan Pratt, combined with outline grid-adjustment techniques invented at Imagen.

In Imagen's technology, designers labeled salient points on conic character outlines for grid adjustment by global regularization rules prior to rasterization. A notable feature of Imagen's approach was local, incremental shifting of specific contour points by "exceptions" to the global curve regularization rules. The exceptions could fix "drop outs"—where rasterization failed to turn on pixels in thin elements like serifs or hairlines of curves, leaving a gap in a stroke—and "clog-ups"—where a join of two or more strokes thickened or clogged up an interior corner or shape. Imagen's font technology began as a way to produce high-quality bitmap raster fonts for Imagen's ImPress printer controller and was later incorporated into Imagen's "DDL" document description language, a potential competitor to PostScript. In 1987, however, Imagen's sole big OEM customer, Hewlett Packard's printer division, canceled its license to DDL. The language and its font technology were thereafter canceled as products, and eventually, Imagen was acquired by a rival printer manufacturer, Quality Micro Systems.

Folio: In late 1986, Jacobo Valdés and Eduardo Martinez, who had been principal developers of DDL at Imagen, left to start a new firm, Folio, where they were joined by Michael Sheridan, formerly Imagen's type expert. Valdés and Martinez had not worked on Imagen's font technology but they and Sheridan found that all hinted font technologies involved laborious, designer handwork to mark and regularize master font outlines. "Hinting" was a bottleneck in outline font production. Folio solved the multifaceted problem by designing the font outlines, hint structures, and rasterizer together with pattern-recognition so that fonts could be produced without designer intervention. Folio's technology became known as "Folio Font Format" ("F3").

F3 used conic splines as described by Pratt, and had two major modules. "TypeMaker" and "TypeScaler." TypeMaker included: (1) a converter from other outline formats, for example, Ikarus arc and vector or cubic spline, to F3 conic outlines; (2) an automatic pattern-recognizer to identify and label contour points of significant features of individual character contours and determine rules of form adjustment throughout a font; (3) an automatic grid-fitting program generator to control character adjustments to raster grids.

F3 was based on a low-level, procedural programming language in which grid-fitting adjustments were programs for each character. An F3 program for a character, or a "symbol" in F3 terminology, contained outline shape information as contours defined by conic or cubic curves, a set of procedures specifying that contour points were to be moved, in which directions, and by what distances, to make the outline contours fit a given output raster grid. [note 27]

"TypeScaler" was a compact, fast, and portable interpreter of the F3 procedural grid-fitting programs. It was designed to be integrated into controllers of screens and printer. After an F3 outline character had been scaled and fitted to a specified raster grid, the TypeScaler imager rendered the shapes as bitmaps.[19]

Folio demonstrated F3 font technology to computer platform and software firms, including Apple, Microsoft, and IBM, as well as to font firms, including Linotype and Monotype. When Apple was researching font technology, they invited other font technology developers to disclose information about their formats, ostensibly to design Apple's future font technology so that other formats could be converted to it easily. Folio disclosed several F3 features to Apple. Folio was acquired by Sun Microsystems in 1988 and F3 font technology was released as workstation software and drivers for Sun printers in 1990.[20]

Nimbus and Nimbus-Q: In the late 1980s, URW developed "Nimbus," a scalable, grid-fitting, outline font technology that could be incorporated into printer controllers. It used structured featural information in an Ikarus-based arc and vector font format. In 1989, Nimbus entered the Font Wars when Mike Parker, former type director of Mergenthaler Linotype and a co-founder of Bitstream who later left the firm, became a marketing representative for URW. Parker arranged for Nimbus to be translated from Fortran into the C language for easier portability and greater familiarity to programmers using Unix. Called "Nimbus-Q" (the "Q" meaning "Quick"), the C version of Nimbus was marketed in the US by Parker's firm, "The Company."

By 1991, Nimbus could control eleven different character features for high-quality output, and by 1994, it incorporated more "intelligence," as Karow stated in a summary and discussion of the major forms of "intelligent font scaling."[13]

Although a few technology and printer firms were able to invent font technologies, they had few or no fonts of their own. For a font technology to be successful, however, it needed many fonts to support it, just as operating systems need many applications to be successful.

Nimbus-Q had potential access to URW's library of more than 1600 Ikarus fonts, more than any other digital font vendor of the time. In 1989, Microsoft investigated an acquisition of Parker's firm, "The Company," with rights to Nimbus-Q and access to the URW font library.

An obstacle to Microsoft's potential acquisition of Nimbus was Hewlett Packard's printer division preference for Compugraphic Intellifont. Compugraphic had a large library of digital fonts in Ikarus format that could be converted to Intellifont format, making Intellifont technology attractive. As a tactical partner to Microsoft, Hewlett Packard urged Microsoft to adopt Intellifont.

Intellifont Versus Nimbus-Q: In 1989, Microsoft arranged a juried "blind-test" of Intellifont versus Nimbus-Q. The jurors were presented with text output but were not told which firms and technologies had generated the output. Representatives from Microsoft and Hewlett Packard, along with an independent type expert, convened at Hewlett Packard's printer division and reviewed sample output from the candidate technologies. The samples included print-outs of letters at large sizes, to judge outline accuracy, clarity, and smoothness, and print-outs of smaller text at standard reading sizes, to judge the quality of rasterization, readability, and emergent qualities like texture, spacing, and weight. The jurors' ratings were recorded and analyzed. Despite the meticulous testing, the outcome was inconclusive; neither of the formats was a clear winner.

Microsoft then re-engaged in discussions with Apple on the "Bass-Royal" technology. Those discussions led to the Microsoft-Apple agreement announced at Seybold in September 1989. Nevertheless, Hewlett Packard did license Compugraphic Intellifont and included it in its PCL Level 5 laser printers, so, for a few years, Intellifont was the most widely distributed "intelligent" font scaling technology in laser printers. Nimbus-Q was not adopted by a major computer platform vendor.

Bass/Royal to TrueType: In 1987, Apple began research on a proprietary font technology, first code named "Bass," then "Royal," and finally "TrueType" after the 1989 Seybold Conference announcement by Microsoft and Apple. Although entering the font technology battles later than URW, Adobe, Compugraphic, and Imagen, Apple benefitted from its experience with PostScript fonts and from ideas developed and disclosed by other firms.

For outlines, Apple chose quadratic B-spline outlines like those in Apple's Draw program for Macintosh. (Some sources refer to the same splines as quadratic Bézier curves.) For grid-fitting, Apple adopted a procedural approach similar to that of Folio F3, but separated the grid fitting programs from the outlines, whereas the F3 code was integrated into the character. Apple called its grid-fitting code "instructions" because each TrueType character includes a low-level computer program that specifies which outline points should be moved in which directions to which grid points, and by what distance, in order to adjust the character outline to the output grid.[21]

TrueType instruction code also includes special-case "delta instructions" (sometimes called "delta hints") to handle local rasterization problems. These were similar to Imagen's idea of "exceptions" but implemented differently. Delta instructions apply only to specific shapes in certain characters at specific sizes, mainly to ameliorate localized drop-outs, clog-ups, or shape distortions. They enable very fine-tuning of font rasterization, but at the expense of greater labor in coding the hints. Delta-instructing has been called high-level bitmap editing because it adjusts low-level bits with adjustments to high-level outlines.[21]

The fine-tuning flexibility of TrueType not only enables precise adjustment of Latin fonts but also enables grid fitting of non-Latin alphabetic letter shapes and of non-alphabetic writing systems. Shortly after the release of TrueType in Windows 3.1, Microsoft commissioned fonts that included

Greek Cyrillic, Hebrew, Arabic, and other scripts along with Latin alphabets in the same font.[22]

Apple initially lacked its own fonts apart from the bitmap "City" screen fonts of the original Macintosh, but in 1989 Apple commissioned URW to create a translator from Ikarus to TrueType outlines, in order to establish a reliable source of fonts and translation software to help third party developers produce greater numbers of TrueType fonts. URW integrated the Ikarus-to-TrueType translator into Ikarus software, including a newly released Ikarus M application for Macintosh. The translator enabled URW and other font developers using Ikarus to develop TrueType fonts for Apple and Microsoft.

In May 1991, Apple released the first production version of the TrueType rasterizer in Macintosh System 7.0, bundled with a core set of Times Roman, Helvetica, Courier, and Symbol fonts in TrueType format. Macintosh System 7 also included scalable TrueType versions of the four most popular Apple "City" fonts (New York, Geneva, Monaco, Chicago). Those were originally hand-edited, low-resolution bitmap fonts in the first Macintosh system in 1984. TrueType was developed to rasterize high-resolution outlines as low-resolution bitmaps, but the reverse process was not feasible. The Apple bitmap City fonts were too low in resolution and too variable among different sizes to be scaled up. Instead, Apple commissioned new versions of City fonts designed, digitized, and instructed as high-resolution TrueType fonts.[23]

The next major milestone for TrueType came in April 1992, when Microsoft released the Apple TrueType rasterizer integrated into Windows 3.1. Subsequently, Microsoft redesigned and rewrote the code of the TrueType rasterizer for greater precision and reliability, an effort led by Eliyezer Kohen.

To prime the pump and encourage usage of TrueType fonts, Microsoft released a "Font Pack" simultaneously with Windows 3.1. The Font Pack contained 44 TrueType fonts, half of which were renditions of known analog typefaces and half of which were new and original type designs for digital technology, most of which are widely used today. Several million Font Packs were sold.

To facilitate hint editing of TrueType, Microsoft later developed an interactive tool, Visual TrueType (VTT) for type designers and engineers to apply grid-fitting instructions to TrueType fonts. Developed by Beat Stamm, the tool displays outlines of characters and editable linkages between their parts to regularize rasterization (see Figure 5). Although VTT is specific to TrueType, the abstract principles of regularization, proportion, alignment, and relationship between character parts apply to all hinting technologies.[24]

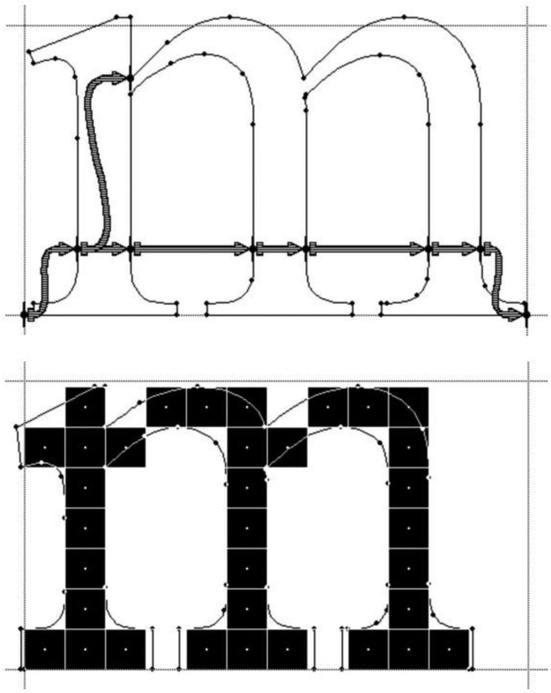


Figure 5. Letter 'm' in VTT editor showing constrained linkages between letter parts in outline and resulting bitmap. Courtesy of Microsoft Corporation.

Among graphic designers and users already accustomed to PostScript fonts, there was initial resistance to TrueType, which could not always reliably run on then-current PostScript image setters and graphics applications. PostScript Type 1 fonts continued to be preferred in graphic arts and design fields because in the early years of TrueType, high-resolution image setters designed to be driven by PostScript were not able to rasterize TrueType fonts without errors. Moreover, Adobe Type Manager (ATM) for Macintosh became popular among graphic designers and publishers after its release in 1989–1990, providing improved quality on screens.

In office environments dominated by Microsoft Windows, however, TrueType became the standard font technology and for the most part was transparent to users. Hence, the desktop publishing world and the office computer world supported two different font technologies, bifurcated by platforms, environments, and applications.

Metafont: In the late 1970s, Donald Knuth at Stanford began work on Metafont, a computer language for generating digital type, which he described in a 1978 lecture to the American Mathematical Society and subsequent publication.[25] [note 28]

Knuth created Metafont to make fonts for typesetting mathematics with his text composition and layout system (TeX), so that he could achieve a typeset output of high quality for his books, notably *The Art of Computer Programming*. He therefore desired to control font output down to the level of individual pixels. He also wanted both his Metafont and TeX systems to be archival, such that the same input would produce the same output after decades or more had passed (this has, in fact, proven to be the case). Knuth always considered his primary task to be writing books on computing, not distributing software, so he worked outside the commercial paradigm that drove several participants in the Font Wars; neither popularity nor profit was of concern to Knuth.

As a near-legendary computer programmer, Knuth implemented Metafont (like TeX) as a batch programming process, rather than drawing shapes on a screen, as in essentially all other font design tools. The designer (or a collaborator) of a Metafont font letter writes a program that generates the shapes. A true Metafont program is specified in plain text, giving instructions like "start character A" or "draw a line from point x to point y," analogous to a typical computer program written in, say, C, which has instructions like "start function f" or "assign 1 to the variable x."

Knuth's programming approach has the advantage of being generalizable to an endless variety of shapes, limited only by the skill and imagination of the designer/programmer. A Metafont program does not simply draw the shape a designer wants but provides structural and proportional rules for the computer (that is, the Metafont program) to draw that shape. For example, an oblique version of a font can be generated by giving one global instruction giving the angle of slant, and Metafont will do essentially everything else. Geometric positioning and alignments of paths, terminations, sans serif and serif styles, etc. are all extensible across an entire typeface family. Thus, Metafont made parameterization a foundation of typeface construction. Knuth provides for some 62 parameters in his Metafont program for his Computer Modern font family. [26]

A disadvantage of the programming approach is that font designers must also be programmers, but visual designers rarely have the time, inclination, or skill to write substantial programs instead of drawing shapes. Hence, despite much early enthusiasm for Metafont, it never became popular among designers.

Knuth was mostly self-taught at the beginning of his type design work. In his first version of Metafont, called MF79, a program for a character described the path of a virtual tool shape, such as an idealized pen or brush, following curves or lines in a two-dimensional raster plane, and leaving a bitmap trail in its wake. This was an appealing construction, hearkening back to scribal eras before typography, when nearly all texts for continuous reading were hand-written with a rhythmic pattern of strokes.

However, type design is not handwriting. After working with renowned type designer Hermann Zapf and other noted typographers for several years, Knuth completely revised Metafont. The final version is called MF84, and it is based on specifying arbitrary outlines for shapes, not restricted to what can be represented as the path of a writing tool, though the writing metaphor is still supported.[27]

Using Metafont, Knuth created a family of digital typefaces named "Computer Modern." Knuth's first books on *The Art of Computer Programming* had been keyboard-composed in metal fonts of Modern No. 8A, a type series produced by the (American) Lanston Monotype Corporation in the late 19th century, and eventually furnished with a wide range of mathematical characters and symbols, becoming a de facto standard for mathematical composition. When Monotype metal type composition became commercially obsolescent and too expensive for publishers, Knuth modeled his Computer Modern on Modern 8A to retain the look he favored in his publications.

Knuth wrote Metafont programs not only for letters, signs and a large collection of mathematical symbols; he also extended their typographic variations, producing oblique as well as cursive italics, sans serif styles, bold and light weights, and parametric variants for larger or smaller print sizes, following traditional proportional adjustments of fonts for different output sizes, rather than the linear scaling that has become most common for fonts today.

Computer Modern thus became a family of fonts more extensive and complex than the original Modern 8A; in many respects, it constituted a new family of original type designs, though its seriffed forms remained in the genre of late 19th century "Modern" styles.[26]

Computer Modern was and continues to be freely distributed like Knuth's TeX software system, thus bypassing font commerce fraught with piracy and plagiarism. The free availability of Metafont made it useful not only to mathematical and scientific publishing in Latin-based orthographies like English but also to non-Latin orthographies that lacked digital fonts in the early era of the Font Wars. From around 1979 to the 1990s, Metafont was used to produce fonts for Greek, Cyrillic, Arabic, Devanagari, and other non-Latin scripts. Experiments with it continue.

ATM and PostScript Type 1 Font Format: To compete with the Apple-Microsoft TrueType alliance that threatened Adobe's font hegemony and possibly PostScript itself, Adobe shifted engineers from Display PostScript, developed for screen display rather than print, to ATM, which was released for the Macintosh operating system in December 1989, well before the first launch of TrueType in Apple Macintosh System 7 in 1991. With ATM, Adobe separated Type 1 font technology from the PostScript language so it could be ported to Macintosh, Windows, and other operating systems. Adobe also improved the quality of Type 1 font rendering on screen to produce aesthetically regularized text without hand-editing of bitmap fonts. Adobe also automated aspects of the hinting of fonts to speed production. In October 1990, Adobe began shipping ATM for Windows 3.0, before the advent of TrueType in Microsoft Windows 3.1 in 1992.

To support PostScript as well as its Type 1 font technology, Adobe published the specification of the Type 1 font format in 1990.[16] The publication of the previously secret Type 1 format was intended to spur the development of Type 1 fonts by designers and vendors other than Adobe and its PostScript licensees. Publication enabled developers of font-making software to incorporate Type 1 font generation capability in their applications. As intended, several other font firms as well as independent designers began to produce PostScript Type 1 fonts.

But the publication of the Type 1 format had an unintended consequence. Because typeface designs are not copyrightable in the U.S., font pirates soon took advantage of the new font tools that could generate TrueType and converted most of Adobe's Type 1 fonts and those of its licensees' into TrueType fonts. The conversions from Type 1 fonts quickly resulted in a large supply of TrueType fonts but only a moderate increase in Type 1 fonts, the opposite of what Adobe had apparently intended. Adobe brought a lawsuit against one alleged pirate by claiming infringement of software copyright, not design copyright. After several years of litigation, Adobe prevailed in court, but meanwhile, the profitability of fonts had diminished. The ease and prevalence of digital font copying and distribution illustrate one of the fundamental premises of Paul Romer's model of the role of the nonrivalry of ideas and knowledge "spillover" in technological progress. Digital fonts had become more like freely exchangeable ideas than material objects. [note 32]

Fonts enrich the capabilities of operating systems and applications. Word processors, page layout programs, illustration programs, spreadsheets, and many other applications, as well as system, all benefit from the typographical expressiveness of a variety of type styles. The plethora of digital fonts, both authorized and pirated, relieved OS makers of the need to invest much in font development, after initial releases of core fonts bundled with the respective font technologies. Over the next decades, however, both Microsoft and Apple did commission original, proprietary fonts, sometimes for technical reasons, sometimes for marketing purposes, sometimes to control their corporate destiny by owning their system fonts, and sometimes for all such reasons.

Font Peace

After the main events of 1992, much of the 1990s seemed more like a time of diplomacy than outright war. TrueType became a widely used font standard along with PostScript Type 1. They co-existed in overlapping but somewhat different market sectors, from which neither could be dislodged—Type 1 format in the high-end graphic arts market, TrueType in the office and PC mass market. Eventually, Apple made a truce with Adobe to support PostScript and Type 1 fonts again on Apple printers, ending another battle in the Font Wars.

A new brushfire Font War erupted after 1994, when Apple released TrueType GX, an advanced enhancement to TrueType. Developed by Dave Opstad and Eric Mader at Apple, TrueType GX solved crucial problems in composing texts for writing systems more complex than the Latin alphabet. Since the European invention of typography in the mid-15th century, types had been Latin-centric, based on small sets of alphabetic characters, such as the capitals and lowercase used for most European and American languages. Latin character sets had been pruned over time. Gutenberg's 42-line Bible, the first major book printed with movable type in Europe, used a font with more than 200 characters, including multiple shape variations and "ligatures" (letters "tied" together) from the medieval manuscript tradition. Multiple letter variations were mostly abandoned in later decades of typography and printing.

Many non-Latin "scripts" or writing systems, however, contain sets of characters that change shape depending on the context of their occurrence in the text, according to rules specific to the script. For Latin type, only a few shape variant characters have survived, such as the f-ligatures including "fi," "fl," and "ff." In Arabic scripts, obligatory structural rules dictate how letters should assume different shapes depending on whether they occurr in a word-initial, medial, final, or isolated position, with additional rules governing ligatures and the positioning of letters at different heights relative to the baseline. The writing systems of the languages of India are also complex, with variant shapes, ligatures, and diacritics following traditional linguistic and graphical rules. Latin-based typography was not able to fully represent Arabic, Indic, and other non-Latin writing systems, often forcing fonts to limit shape variations and rules in ways that were aesthetically and linguistically unsatisfactory. [note 29]

Apple's TrueType GX was designed to allow the proper expression of non-Latin scripts. One of its most basic and important capabilities was glyph substitution, the changing of the letter and character shapes depending on shape change rules in the writing system. Apple's TrueType GX "line layout" technology analyzed a line of text and determined which shape variants should be substituted in which contexts. [note 30]

Some font industry observers assumed that Microsoft would eventually adopt TrueType GX for Windows, but Microsoft decided on a different course, for reasons that remain unclear or varied. In 1995, Microsoft published a different proposed technology, initially called "TrueType Open," to handle the same sorts of line layout and glyph substitution problems solved in TrueType GX. The initial specification was developed by Eliyezer Kohen and Dean Ballard.

Microsoft's announcement of a line layout format incompatible with TrueType GX caused application developers to delay or abandon software upgrades supporting TrueType GX.

In 1996, Adobe agreed with Microsoft on a newer font format called "OpenType," which would incorporate the overall data structure of TrueType, the additional line layout technology of TrueType Open, and the Unicode Standard character encoding.[28]

To be backward compatible with original TrueType, OpenType rasterizers were designed to process nearly all existing TrueType fonts, and also process Adobe's new Compact Font Format (CFF), a re-engineering of PostScript Type 1 format for greater data compression and faster execution, though retaining declarative hinting and Bézier cubic curves. Hence, an OpenType font could contain TrueType quadratic curves, or Adobe Bézier cubic curves, or both, thus making an armistice between the warring curve forms. [note 30]

It took several years after the 1996 announcement of OpenType for the technology to be supported by major computing platforms and applications, including Microsoft Windows 2000, Macintosh OS X, and Adobe InDesign. It took almost seven years for Adobe to convert its two thousand Type 1 fonts to OpenType CFF format. Most of the thousands of existing TrueType fonts, whether authorized or pirated, were also usable with OpenType technology.

Who Benefitted From the Font Wars: The Font Wars were a time of tremendously rapid technological innovation and intense competition in ideas, algorithms, implementations, and products It was perhaps inevitable but nevertheless unfortunate that they were called "wars," which emphasized the competition but underplayed the ideas, reducing the contest to winning and losing. It is more important to ask, who benefitted?

Literate Civilization: Worldwide literacy rates were already increasing in the 19th and 20th centuries, but digital reading devices have boosted worldwide literacy because of the typographical technologies developed and broadly distributed during the Font Wars This is especially true for readers of non-Latin scripts that analog typography had seldom fully represented. Now, fonts for more than 150 scripts and writing systems are potentially usable on computers, cell phones, electronic displays, and other digital devices. [note 30]

Billions of readers and writers in hundreds of languages now use digital font technology to communicate every day, throughout the world, without the technical complexities and limitations that made typographic literacy difficult or unobtainable by previous generations. Whether writing short text messages, emails, or long documents, writers in many languages and cultures now have easy access to literate tools.

Moreover, modern computers, smart phones, e-books, and other display devices can scale fonts to different sizes, reverse polarity (white type on black background), and change font styles and spacing to adjust graphical parameters of text to visual acuities and reading abilities. Among others, older readers, younger readers, readers with vision deficits, and struggling readers now have easier access to a wider range of potentially more legible reading materials. Similar affordances also enable typographic changes for aesthetic enjoyment of reading, often termed "congeniality" or "readability."

The Font Wars also benefitted today's giant social media companies, most of which had not yet been founded during the Font Wars but which have since relied upon digital typography to create and display the hundreds of trillions of messages sprayed across the Internet by billions of readers and writers every year. As writer Beatrice Warde foresaw, the galaxy of digital typography has dwarfed even the Gutenberg Galaxy.

Computer Platforms: Three of the four major warring platforms in the Font Wars survived and prospered Apple, Adobe, and Microsoft had been, at various times, allies, enemies, and/or both, as battlefields shifted and hegemonic alliances formed and reformed like warring city states in Renaissance Italy.

Exemplifying Paul Romer's theory of economic progress through technological innovation, the competing firms all benefitted from the creation, exchange, publication, and competition of ideas from individuals and institutions: supported by patrons, businesses, and polities. There is, for example, a remarkable correspondence between the early era of printing in the 15th century and the era of the Font Wars, five hundred years later, when both eras are viewed together through the lens of Romer's theoretical model. This comparison deserves a separate essay on its own but is at least sketched out in a note to this article. [note 31]

By the beginning of the 21st century, all three firms supported OpenType, the hybrid offspring of TrueType and PostScript Type 1. Notably, Apple, Adobe, and Microsoft had certain organizational and managerial features that contributed to their success. They derived most of their font technology ideas from prior university and corporate research. They looked beyond short-term markets and technical limitations to future opportunities and technical advances that would change the economics and capabilities of font technology. The same firms' leaders had the clairvoyance or luck to enter the commercial market with propitious timing, neither too early for current technology to support their ideas, nor too late to catch up to competitors already underway. They strove to establish standards, often self-declared before actual market dominance. They formed alliances: if small, they teamed with, or were acquired by, larger firms. If big, they worked with or acquired innovative smaller companies.

Yet, given all those advantages, a crucial factor was whether a firm's executives were committed and steadfast in supporting and advancing their own technologies, especially during times of high competitive stresses when fears, uncertainties, doubts, confusions, and temptations buffeted nearly all firms. [note 32]

The phrase "Font Wars" used throughout this history does not adequately acknowledge the achievements of many who contributed to the radical transformation of literate technology, the consequences of which are not yet fully known or comprehended. Yet "wars" they have been called, so this history tries to describe the often wonderful ideas and the people who discovered or advanced them. To the extent that this has been about "wars," it is a faint, distant echo of the first great history of wars, written some twenty-five centuries ago by Herodotus, whose "History" of the ancient wars between Greece and Persia, and of much else besides, remains the definitive account of the beginnings of western civilization. So, to end this history, we may recall how Herodotus began his:

"Herodotus of Halicarnassus here presents his inquiry, so that deeds done by man may not fade in time and so that great and wonderful feats – some by the Greeks and some by the Barbarians – may not lose their fame, and that the reasons they went to war against each other may not be forgotten."

ACKNOWLEDGMENTS

I wish to thank the readers who generously commented on multiple drafts: Karl Berry, Kris Holmes, Peter Renz, David Walden, and an anonymous reviewer. I am responsible for any remaining errors.

REFERENCES

1. M. A. Changizi and S. Shimojo, "Character complexity and redundancy in writing systems over human history," *Proc. Roy. Soc. B, Biol. Sci.*, vol. 272, no. 1560, pp. 267–275, 2005.
2. D. G. Pelli, C. W. Burns, B. Farell, and D. C. Moore-Page, "Feature detection and letter identification," *Vision Res.*, vol. 46, no. 28, pp. 4646–4674, 2006.
3. P. Standard *Arrighi's Running Hand*. New York, NY, USA: Taplinger Publishing Co., 1979.
4. A. Fairbank, *A Handwriting Manual*, Dryad, Leicester, U.K., 1932. Revised editions, 1947, 1961, and other instructional manuals.
5. G. Noordzij, *The Stroke - Theory of Writing*, P. Enneson, tr. London, U.K.: Hyphen Press, 2005.
6. E. M. Catich, *The Origin of the Serif: Brush Writing and Roman Letters*. St. Ambrose, IA, USA: Catfish Press, 1968.
7. C. Bigelow, "Form, pattern, and texture in the typographic image," *Fine Print*, vol. 15, no. 1, Apr. 1989.
8. F. Boas, *Primitive Art*. New York, NY, USA: Dover Publications, 1955. [Reprint of 1927 edition]
9. D. K. Washburn and D. W. Crowe, eds., *Symmetry Comes of Age: The Role of Pattern in Culture*. Seattle, WA, USA: Univ. Washington Press, 2004.
10. N. Majaj, D. Pelli, P. Kurshan, and M. Palomares, "The role of spatial frequency channels in letter identification," *Vision Res.*, vol. 42, no. 9, pp. 1165–1184, 2002.
11. C. Bigelow and D. Day, "Digital typography," *Sci. Amer.*, vol. 249, no. 2, pp. 106–119, 1983.
12. B. Stamm, "The raster tragedy at low-resolution revisited: Opportunities and challenges beyond 'delta-hinting.'" [Online]. Available: http://rastertragedy.com
13. P. Karow, *Font Technology: Description and Tools*. Berlin, Germany: Springer, 1994, Chapter 7: "Intelligent Font Scaling," pp. 105-149, Chapter 8: "Effects of Rasterization," pp. 151-172.
14. G. E. Legge, *Psychophysics of Reading in Normal and Low Vision*. Mahwah, NJ, USA: Lawrence Erlbaum Associates, 2007.
15. P. J. M. Coueignoux, "*Generation of roman printed fonts*," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 1975.
16. Adobe Systems. *Adobe Type 1 Font Format*. Addison-Wesley, Reading, MA, USA, 1990.
17. T. Hawkins, "*Method for producing a scaleable typeface data*," United States Patent 4,675,830, Jun. 23, 1987.
18. M. Carter, "The digital typefoundry," *Visible Lang.*, vol. 50, no. 2, pp. 26–37, 2016.
19. Sun Microsystems. *F3 Font Format Specification*. Sun Microsystems, 1991.
20. M. Marshall, "Sun brings PostScript to non-PostScript printers," InfoWorld, Oct. 1, 1990.
21. "TrueType 1.0 Font Files: Technical Specification," Revision, vol. 1, no. 66, Aug. 1995, Microsoft Corporation, Redmond.

22. C. A. Bigelow and K. Holmes, “The design of a Unicode font,” *Electron. Publishing*, vol. 6, no. 3, pp. 289–305, 1993.
23. C. Bigelow and K. Holmes, “Notes on Apple 4 fonts,” *Electron. Publishing*, vol. 4, no. 3, pp. 171–181, 1991.
24. B. Stamm, “Visual TrueType: A graphical method for authoring font intelligence,” in *Proc. Int. Conf. Raster Imag. Digit. Typography*, 1998, pp. 77–92.
25. D. E. Knuth, “Mathematical typography,” *Bull. Amer. Math. Soc. (New Series)*, vol. 1, pp. 337–372, 1978.
26. D. E. Knuth, *Computers & Typesetting, Volume E: Computer Modern Typefaces*. Reading, MA, USA: Addison-Wesley, 1986.
27. D. E. Knuth, *The METAFONTbook*. Reading, MA, USA: Addison-Wesley, 1986.
28. *OpenType font file specification*, Version 1.01, Microsoft Corporation, Redmond, WA, USA, 1997.

Charles Bigelow is Scholar in Residence with the Cary Graphic Arts Collection and former Cary Distinguished Professor with the Rochester Institute of Technology, Rochester, NY, USA. He is a MacArthur Foundation Prize Fellow and was Professor of digital typography with Stanford University from 1982 to 1995. With Kris Holmes, he designed the Lucida fonts distributed by Adobe, Apple, Bell Labs, Microsoft, and Sun Microsystems. He was a typographic advisor to most of those same firms and others during the Font Wars. Contact him at lucidafonts@gmail.com.

The Advent of Digital Typography

Interview With Liz Bond Crews

Paul McJones

Liz Bond Crews began her career as a systems analyst in Los Angeles, CA, USA, for RCA. She soon joined the Xerox Electro-Optical Division in Pasadena, CA, where she led a group building special systems for *National Geographic* magazine, Ginn & Company, and University Microfilm. In her final position at Xerox, she licensed high-quality typeface designs from Mergenthaler Linotype, among others, for use on Xerox's high-performance 9700 laser printer.

In the early 1980s, the Adobe cofounder John Warnock recruited Crews to Adobe Systems, where her first task was to license high-quality typeface designs to be rendered as Adobe Type 1 fonts for PostScript printers. While at Adobe, she was responsible for bringing high-quality typography to the Adobe PostScript page-description language, running corporate marketing, hiring graphics designers, and putting Adobe on a firm footing with professional designers and printers, greatly enhancing the value of their product. After Adobe, she helped found Electronics for Imaging and, after some consulting, retired in 1993.

Digital Object Identifier 10.1109/MAHC.2019.2929883
Date of current version 6 March 2020.

This interview, conducted by Paul McJones, is based on an oral history transcript produced by the Computer History Museum.[1] The original transcript was recorded on May 24, 2017 in Mountain View, California.

EARLY EDUCATION AND EXPOSURE TO COMPUTING

Liz Crews: I grew up in South Texas, on the border of Reynosa, Mexico. It was a bicultural experience. It was a very loving environment, and I was encouraged by my family to excel in whatever I did. I enjoyed school immensely, and I ended up doing a lot of activities in school. For example, I ran for president of the senior class in the student body government. In those activities, I think I really exceled in doing creative things. I'll never forget the campaign I used to run for student body president; I came up with slogans like "Liz knows her biz! Vote for Liz!" Sadly, I lost the election by six votes to the captain of the basketball team. It was an encouraging environment, but it was also a sheltered environment. I was fortunate to have a family who loved me, cared for me, and absolutely insisted that I excel.

Paul McJones: So they encouraged you to go to college. Had they gone to college?

Crews: Daddy had, yes. He was an orthodontist. After World War II was over, we moved to Altus, Oklahoma, where my father practiced dentistry. I was born in Oklahoma. But then the Korean War broke out, and they recalled all doctors. He was sent to Korea for two years. When he returned, we moved to Memphis, Tennessee, and he worked on his advanced degree in orthodontics and maxillofacial surgery. After that, because they enjoyed Corpus Christi, we moved to McAllen, Texas.

When I graduated from high school, I wanted to do something different. I wanted to get out of that environment, so I went to Bradford College, a junior college north of Boston, Massachusetts. It was a small school, and in fact, there were only two girls from Texas in the entire school. It was an interesting cultural switch for me from growing up in a small town in Texas to going all the way back east to Boston. I enjoyed that environment very much. I loved the art history, the cultural aspects, and the opportunities I had back there. But the classes I enjoyed the most actually were of math. At Bradford College, I was an editor of the annual and quite involved in activities. After I graduated, I was told I couldn't stay back east and had to come home to Texas.

From there, I went to the University of Texas and finished my degree in math and computer science. In those days, undergraduates were not allowed to enter in the computer room. We did everything by punch cards and submitted our programs to the computer center. If you dropped your box of cards, you were really in trouble.

McJones: There weren't too many women, I suspect, in that program with you.

Crews: No, there were not many women in the university's math program. Ironically, I really wanted a computer science degree. In order to get that, I had to have a double major. Math was obviously the right double major. I wasn't that interested in the theory side of math, but I liked the application side. It was like problem solving to me and the programming was fun.

FIRST POSITION AT RCA

McJones: You graduated in 1968, correct?

Crews: Yes, 1968. The question was, what would I do with the degree? In those days, interviewers came to your campus and you interviewed with everybody. I had an opportunity to go to several different organizations—Cape Canaveral and NCR had a program. I interviewed at quite a few places around the country. I had about decided that I wasn't really that interested in the programming aspects entirely, so I sort of leaned toward organizations that were offering me the opportunity to be more like an analyst because I enjoyed solving problems and debugging. I felt I would rather do that than just be on a programming team. That's why I ended up with RCA.

I enjoy talking to people and discovering the real issue. Still, this is obviously the very beginning stages. When I was a system's analyst at RCA, I was assigned full time to a corporate account. Software developers would come in and say, "I'm having problems. I can't solve this." I'd go through their core dumps with them. We'd find the error and figure out what to do about it.

McJones: Tell us about some of your customers at that point.

Crews: Signal Oil and Gas was an interesting one that I enjoyed working with. They had a large department and I went through a lot of dumps with them. Another client was Glendale Hospital in the Los Angeles area. That was my first time to get involved with Cobol. The developers were using the RCA Spectra 70 computers and had written an accounting system in Cobol. They were having difficulty in expanding it. I went in to help them document the system, figure out how we could expand it, and then wrote code for that. That was an interesting project because I discovered very quickly how important it was to succinctly document your code and leave the right opportunities for future modifications. It was an enlightening experience because what I inherited to fix was really not very good.

McJones: You began with Fortran in college, but at this point, you were programming in assembly language and Cobol, so you had a broad range of technologies.

Crews: Yes, when I joined RCA, one of the things that appealed to me is they sent everybody back to Cherry Hill, New Jersey, for a three-month program. I was surprised at the number of people in that systems analyst program who didn't have any math or computer background. It was easy for me because I had that background, but it was a really good training program.

McJones: RCA was in the LA area?

Crews: Yes. I'd wanted to go to San Francisco, which to me seemed like the ideal place for my first opportunity. In those days though, there was nothing going on in San Francisco in the computer industry, so Los Angeles was it!

MOVE TO XEROX ELECTRO-OPTICAL SYSTEMS

McJones: How long were you in that job?

Crews: Unfortunately, in the early 1970s, RCA disbanded its computer business, selling it off to Sperry Rand. At that point, I thought, "Now what am I going to do?" I went about interviewing with different organizations, and I even contemplated becoming a stockbroker. I interviewed with Xerox Electro-Optical Systems in Pasadena. I was fascinated with that organization because it was a group of scientists and they were looking for a systems person. I thought, "My gosh, this is where I should be," so I joined EOS.

There I encountered a new set of problems. In the beginning, EOS did a lot of government contract work. One of the tasks I did came as a result of one of the salespeople in the Washington DC area who had been exploring the opportunity to join up some of Xerox's technology in National Geographic magazine. They asked me to look at the magazine and see what we could possibly offer in automation to help them produce the magazine. I spent probably six months at the *National Geographic* Society interviewing and talking to people, figuring out their workflow.

We discovered that the main issue there was the proofing. How do you proof the pages and the mock-ups? At that time, Dale Green was doing some wonderfully interesting work with a color printer, and it was being electronically driven. So, we proposed a page markup system, and its proof device would be the Xerox color printer. We priced that out as a one supply solution, but the actual price tag was ridiculous because Dale Green only had one machine and he really wouldn't want to put it in a user environment.

University Microfilms had another interesting issue. They were recording all dissertations on film. The corporation said, "Wait a minute, maybe we should be thinking about digitally capturing this information." Rather than doing what we would do automatically today—scan it and store it in a digital file—we chose to actually rekey the dissertations. In other words, the abstracts would come in, and then somebody would rekey them into a digital format. Then we would just create a page layout. Of course, dissertations abstracts have a consistent layout form, so it was easy to design a descriptive page format that you could print. We used Xerox Alto computers.

My group developed the software to do this page layout system for University Microfilms. The biggest problem we had was how to connect different dissertations together so that you could actually print them in a format. We used Ed Taft's FTP work that was done at PARC. At that time, Ralph Kimball also had done some interesting typesetting of formulas, so we could actually typeset mathematical formulas. Using Ralph's work, we developed a page layout system with the hope that we would be able to actually digitize all the abstracts came in-house so they wouldn't have to go to cut and paste. That system ran for about a year, but there were anomalies. At some point, we said, "This is not going to work because it's not a production hardware, and we cannot solve all the issues." But the client agreed that it had been a good test.

McJones: How big of a team did you have working on something like this?

Crews: I had three software engineers and myself.

ADVANCES IN DIGITAL FONTS AND TYPEFACES

McJones: You mentioned Ed Taft. Are there other people that you were interacting with a lot from PARC?

Crews: Yes. I followed both Alan Kay's and Adele Goldberg's works. Of course, I knew John Warnock and Chuck Geschke, Bob Taylor, and Butler Lampson. I didn't work directly with Butler Lampson, but I interacted a lot with Larry Tesler and Tim Mott. That was what was thrilling about working at Xerox at that time, particularly at EOS. It gave me the flexibility to really go around the corporation and work with different people.

After exploring these different opportunities with the Xerox Education Group, I was given the opportunity to take over managing the Xerox Corporate Font Center for the Printing Systems Division in El Segundo, California. By this time, I thought that was a good idea. I'd been doing other separate projects. The development team was a small group, but this sounded like a really interesting problem to address.

When I got down to Xerox, I discovered some interesting things that were going on through the

Corporate Font Center. The majority of the electronic printers were Model 9700 high-speed printers. The first industry to adopt the high-speed technology was the insurance companies. That was a perfect application for those high-speed printers because the majority of it is boilerplate—all you have to do is insert the name of the insured person, the policy number, the dates, and things like that. There is a lot of static information, and then digital information is inserted.

For this system, the customer base would design a new form, a new thing they were producing, such as a policy. They would come back to Xerox and say, "I want my fonts delivered to me in 10.2 characters per inch." Remember those were the days of the fixed-pitch typewriter. Next, they'd want a font done in 10.4 characters per inch. We were making bitmaps, literally a bitmap of every character. We primarily made Universe, or a bastardized version of Universe, that was similar to the Helvetica sans serif font. I was flabbergasted by the number of versions we had made. We were just inundated with this burden of creating all these fixed-pitch fonts!

The company wanted desperately to use Roman PS (proportionally spaced) and Courier. These were particularly desired in England at Rank Xerox. We didn't have access to those, so I went to negotiate the contracts for licensing Roman PS and Courier from Eric Bauer in Neuchâtel, Switzerland. He was not at all happy with the contract because he developed type wheels, and he was Swiss, so he wanted to make it! He didn't object to it, but he thought the idea of me paying him a royalty just so I could use the design was not really what he wanted to do. He wanted to sell print wheels. That was an interesting negotiation.

McJones: Describe briefly the font development process and the tools that were used to go from artwork to the apps.

Crews: The tools were written on Alto. I didn't develop those tools. They were literally a raster generator. The user would bring up each character on the screen and modify it until the shape was created. That meant literally designing it so that they would match the character's requirements of fixed-pitch type faces.

McJones: Did you use the spline-based Fred editor[2]?

Crews: Yes, we used that. That was the first step. Then I realized that the corporation was going to reach a point of not being able to deliver these printers because we didn't have enough typefaces and our internal department could not generate them. We were killing ourselves just generating the other fonts for the existing customer base.

So, I went off to meet some type designers and type developers and was introduced to Mike Parker at Mergenthaler Linotype. Rick Freeman and Xerox negotiated a contract for Mergenthaler Linotype to produce typeface families for us. They would do the bitmap development and deliver the bitmaps to us.

Of course, then the question is, "How many sizes do we make?" because we were now talking about proportionally spaced typefaces. How small of a typeface size could we actually legibly produce on a 300 dpi printer? We defined ten typeface sizes. We knew we were primarily going to be printing on 8.5×11-in paper, and they were going to be text-oriented faces, not display faces, not faces that would be used for advertising work and such. After we got the contract signed, they set off to develop some typefaces for us. I can tell you it was a challenge to get that contract though.

McJones: Was it a question of them being uncomfortable translating their designs into your format?

Crews: Yes, there was a question about which typefaces would lend themselves best to 300 dpi printing. I also worked with International Typeface Corporation (ITC) at that time. Everybody I approached in the type industry was interested in digital electronic printing. In fact, they were fascinated about it. It gave them an opportunity to say, "Can we do this? Can we make 300 dpi fonts?"

There was discussion at that time about whether we needed to incorporate a copier and a printer, if the next version should be a copier as well as a printer. Most of the salespeople were used to selling copiers and discussing the pages per minute and cost per page, rather than appreciating what document production could do. For example, in the color copier area, I remember a lot of people were not convinced that it was going to go anywhere. What would you copy? What really needs to be copied in color? It was a slow progression to where we are today.

McJones: Tell us about your staff while you were doing this font development work. What sort of team did you have? What skills and what people?

Crews: The actual people doing the typeface production, working on the Alto computers, were probably not very skilled. They were just people we trained to use the tool, not programmers.

I don't think they had a strong appreciation of type, and I am not too sure they saw value in what they were doing. It was like a manufacturing area.

McJones: You mentioned you also had some technical people.

Crews: When I got there, all the software had been developed and they were using it. I left the software and the actual production going as it was. But I realized that just adding more terminals wasn't going to solve our problem. We had to figure out a way to do it differently, to also get the typefaces available for the corporation where we had no expertise. It seemed appropriate for me to look outside to bring that expertise in, and I think if I'd stayed at Xerox Corporate Font Center after Mergenthaler would have delivered, we would probably have moved on to a new technology of developing fonts. But at 300 dpi, where you have to make a bitmap for every size you are producing, that's a lot of bitmaps.

By going to Mergenthaler, I outsourced the production. It seemed the only thing appropriate to do at that time, because we had to get some typefaces available for the electronic printing market, and we didn't own the rights to them. By licensing them and having Mergenthaler produce them, we also got the rights to the typeface. We weren't copying anybody else's. We were using the original designs. That was important to me.

McJones: You had credibility because you were part of Xerox.

Crews: As I approached all the type designers in the industry, I would say, "We're Xerox Corporation." Because of that name, I was able to open doors that probably no one else could at that time. They were interested and saw all the opportunity of what was going to happen. They wanted their typefaces licensed by Xerox.

McJones: Were you really moving into the world of typography personally in terms of joining organizations and so on?

Crews: Yes, when I began working with these people, I said, "We have to set the standards of what excellence in typography looks like, because the publishing industry sees Xerox as this organization bringing typography to the world. Therefore, we have to set a level of excellence in what we're doing."

By going to different trade industry meetings and shows, I met an awful lot of people and was asked to join the National Composition Association Board of Directors, which was a group of typesetters. At that time, I was curious why they wanted me. Later, I realized that they were interested in having an insiders' input: "What's Xerox doing? What's the next thing coming?" They wanted to have a dialogue with somebody. Ultimately, I think they realized that they were going to have to expand their business. At the meetings, I suggested we all share what we were doing, how things were moving. Quite a few of them thought, "I need a new business strategy, or I am going to go out of business." I enjoyed that interrelationship, working with these different people.

McJones: Were you meeting the designers as well as the management?

Crews: Yes. In particular, I think one of the vehicles I was enabled to do that was through International Typeface Corporation. They took me under their wing and recommended to join ATypI (Association Typographique Internationale or International Typographic Association). I went to their international conferences, where I met all the type designers and showed them what we were doing. I brought specimens, which we talked about, among other things. I think that what prompted them to think was, "What designs might work in this new xerography?"

McJones: You just mentioned specimen sheets. Would you print various kinds of things using the Xerox hardware, but with their type?

Crews: Yes, I would print things. At that time, Altos were everywhere. At Xerox Printing Systems Division, we were using Altos and Bravo, which was Charles Simonyi's development for word processing. We had one or two typefaces, so I produced examples of specimen type things that I could produce at that time. Of course, I couldn't do much. I could only print a single point size. But I showed them exactly what was happening. I showed them what happened when xerography puts down the toner; it smushes it and certain strokes get heavier. I showed them how we were doing things to make up for that. For example, if a stroke was too heavy, we took out every other bit so that, in essence, when the impression was made on the sheet you had the appropriate stroke. That fascinated the type designers. They hadn't even thought about something like that.

JOINING ADOBE

Crews: Something happened then that was probably the most exciting thing in my whole career: John Warnock came to visit me. Chuck Geschke and he had left PARC by this time, and they came to visit me in late 1982 or early 1983. John showed me exactly what PostScript could do in type, and I was blown away. It would've solved everything. In other words, he proposed that we use PostScript software as font development software. At that point, I went up to the organization saying, "This is it. We need to move to this technology. We'll be able to make the typefaces we need, and we will be able to move forward and supply the corporation with the type." But I couldn't sell it internally.

McJones: Were you trying to convince Xerox to adopt Adobe technology?

Crews: I wanted Xerox to commit to buy PostScript and use it internally for our font generation, but I couldn't sell it internally. I can think of reasons why it wouldn't sell. When I told that to John, he said, "I think you ought to come talk to us."

When I went to talk to John and Chuck in Palo Alto, they showed me more of what they were thinking about—device independence. They had a strong desire and a passion for fine typography. They didn't have direct contacts with type designers or the type foundries. They knew about them, but they didn't have an "in" with these people, so they said, "Why don't you come and join us? Let's do this."

So, I decided to quit Xerox after eleven and a half years. I realized I'd reached a glass ceiling; I was not going to go anywhere, and it was time to move on. It was exciting.

I was employee number 15 when I joined Adobe. I don't remember what my title was. In those days, we were just engineers; you went out and did what needed to get done. The first thing that John and Chuck needed, that Adobe needed, was to license the typefaces so we could generate the type. Right away, I called Mike Parker at Mergenthaler. I told him and Jonathan Seybold that I was leaving Xerox before it was announced publicly—I'd told Xerox, but the industry hadn't been told.

I'll never forget Mike Parker coming over. I sat with him on my back patio in Southern California as we daydreamed together what this could mean. What would it mean if you had device independence and printers were pervasive? He said, "As you know, typefaces cost $35 a typeface." I said, "Mike, it ain't going to go. We can't do that. Just think of the volume and numbers if a copier in your office becomes your printer. They're going to be everywhere." I think he started generating, cogitating, and realizing that the whole pricing scheme for fonts had to change, so that it was a good meeting.

McJones: That was a fascinating story. Let's jump ahead. A font price of $35 would have been for one size?

Crews: No.

McJones: For one typeface.

Crews: It would've been for one typeface and remember that CRT typesetters could generate any size.

McJones: I see.

Crews: What I heard was encouraging, because we wanted the licensing rights to use the design. We did not want to have them produce it for us. I was suggesting that he needed to rethink his strategy. He wasn't selling fonts. He was selling designs. He was selling the Mergenthaler quality and, therefore, deserved a licensing royalty versus a sale. That was a new way of thinking for him, that he now had a royalty-based revenue stream for his designs.

McJones: Would they deliver camera-ready artwork?

Crews: They'd deliver artwork that we would then use. We had the excellence of their artwork, and then we would actually produce the typefaces ourselves.

McJones: Were you running the type production at Adobe?

Crews: No. I didn't do that, mainly because it took a lot more energy and effort to do these contracts, to get things set up. The actual production was done by a group of engineers. My expertise was the ability to get these licenses. That relationship with Mergenthaler helped to get the licensing and the typefaces, which then enabled our salesperson to negotiate the contract saying that they would use PostScript in a lot of typesetters. We wanted the high-end typesetters because we knew that we had the ability of device independence. It had to be both at the high and low end (300 dpi). Mergenthaler Linotype was really a key in the beginning, and I enjoyed giving them the opportunity to get an inroad there.

McJones: Because you already knew these people, you were trying to help them move to a new technology.

Crews: Yes. I'd known the people for quite a while, so it was a good relationship-building opportunity there.

LINOTRONIC MACHINES AND TYPESETTING

McJones: Did you design specifically for the Linotronic machine?

Crews: Yes. The Linotronic was the machine that we drove, or "bundled," our software with, or put PostScript into.

McJones: Describe that machine briefly. Modern audiences are aware of things like laser printers and ink jet printers, but they may not be familiar with typesetters.

Crews: The Linotronic was a film recorder. A lot of typesetters are film recorders, and this was a digital machine, so we were able to put PostScript in. Then, instead of generating output at 300 dpi on a xerographic-based engine, it produced 1270 dpi on film. After it was produced, the film would then be taken to a printer, who would make plates. Then you would replicate as many copies or print as you needed.

Today we think of the machine in our office as being our printer; we don't, in essence, create a plate. We create an image in memory that is produced on the actual output. With the Linotype machine, you, in essence, were going through a production process, which involved going to a printer who makes plates.

McJones: That was on your table when you first started at Adobe, to start negotiating that deal.

Crews: Yes, that and the ITC typefaces. We were also working with Apple at that time. Steve Jobs realized the importance of having typography in the first laser printers. When the first printer came out, I believe it had only two typefaces: Helvetica and Times Roman. (Editor's note: It also had Courier and Symbol. All but Symbol included roman, italic, bold, and bold italic versions, making 13 faces total.) I helped Steve's team pick the typefaces that would be in the LaserWriter Plus. (Added to the original 13 face were four faces each of ITC Avant Garde Gothic, ITC Bookman, Helvetica Narrow, New Century Schoolbook, and Palatino as well as Zapf Chancery Medium Italic and Zapf Dingbats.)

In doing that, I don't consider myself a type designer. I think I have some good graphic design sense, but I looked to the industry experts to help me figure out which typefaces we should recommend to Apple, so that came from a group of people. That is how we ended up with the 35 typefaces that were in the first LaserWriter Plus.

McJones: Yes, there were 13 and then 35 faces.

POSTSCRIPT EVANGELISM

Crews: Once those sorts of things had gotten going, we were working on trying to get original equipment manufacturer contracts. We had this thing called PostScript, but a lot of people had problems understanding what it was because you couldn't see it, and you didn't buy it. It was embedded in a printer. I spent some energy and time going to industry conferences and giving talks about PostScript: what it is, why it is important, and why you were going to need it. I did an awful lot of talking, which I enjoyed. I found it challenging to try to get that information out, to take materials to these conferences and trade shows, etc. I had to write PostScript programs because there were no tools. It was interesting to literally produce your own slides by writing a software program.

McJones: You were going back to your old skills from a long time before.

Crews: After doing that, I ended up being the marketing communications person for the company because we did not have anyone. Of course, I used every opportunity I could for us to speak at a conference. I asked John if he'd do it. Chuck and he were often willing to speak.

Our challenge was, how do you show off what PostScript's going to do? That's when I realized that we needed some graphic arts quality people in-house. I went to the Rhode Island School of Design (RISD) to talk to some of the people there. There were several graphic arts people who were talented and needed summer work. I said, "Why don't you come out to California and work for me? You can use some of these new tools." Hugh Dubberly came out. At the same time, I found Russell Brown and Luanne [Seymour] Cohen and some very talented graphic artists who were willing to leap into this new technology. I hired them for a limited time, but I think John and Chuck realized quickly that they were assets for our company, so we were able to convert some of them to full-time work.

PROOFING VERSUS HIGH-RESOLUTION PRINTING

McJones: Let's back up a bit and talk about the relationship between proof printing and high-resolution printing. I'm wondering if you were introducing two different groups to each other and you had to adapt your story for each. For example, the LaserWriter could be thought of as just a printer, a simple tool for the high-end people, but for the office people it was the main thing and the high-resolution printer was this sort of remote thing. Were there these two cultures? How did they fit together?

Crews: That's interesting. There are two cultures. At that time, this was mostly about black and white printing. We had not introduced color at all into it, so to the high-end person, the one that's looking to produce editions of something, the LaserWriter looked like a simulation of what you were going to get off the printing press; to them it was a proofer, a proof printer. I think that's what the people from the typography world and the industry felt when they saw the first low-resolution printers coming into market. They sold them as proof devices.

Now, of course, the people who were interested in having that capability in their house or in their graphic design studio would say, "My gosh, I've got the final piece. I don't have to go to a printer." So, in essence they were creating in-house, camera-ready art. It's interesting to see how the two industries interrelated. Because the printers were stable, black and white, you did not have any stability issues. Unless your toner ran out, it was a nice solution.

Still, in the beginning all we had was text, so at Adobe we said, "Wait, there is power in PostScript—a device-independent page-description language—that's in these printers that's not even being touched." For example, one of the pieces we would show in the beginning was type on a spiral that went around and around and around, because that was very difficult to do. People were flabbergasted. That's what led, I think, to John Warnock deciding that we should put some energy into making a drawing package that would take advantage of this technology and allow people to do illustrative work. We developed Adobe Illustrator for that, and the packaging was fun because we were doing everything internally. Luanne Cohen was the one who came up with a design using Botticelli's "Venus" for the outside of the packaging.

McJones: You also created a newsletter.

Crews: When you're a startup company, how do you get your message out? You can go to conferences, you can talk to people, you can talk to industry leaders, you can talk to your potential customers, but how do you get your message out to a broad group of people? It occurred to me that maybe we should do a newsletter. This was before newsletters were really ever done. In the newsletter, which we called *Colophon*, we talked about what PostScript could do. A colophon is the thing that you find at the back of a book that lists how you put the book together (the typefaces, the typography, etc.) so that seemed like an appropriate name.

We produced *Colophon* all in PostScript because there were no other tools. We did a large form size and had it printed, and we sent it out to every possible person that would have any possible interest. It was for people we had made contact with. Of course, it was not cheap to produce this because we had to print it and send it. We probably did it over two years. It was a way for us to communicate, between meetings and conferences, with the potential users of this technology, saying, "This is an example of what you will be getting and doing."

ADOBE'S TYPE CATALOG

McJones: Adobe also had to start producing catalogs of typefaces.

Crews: Yes. We were producing type, so we had to produce catalogs for the typefaces. That was a challenge. We did that and packaged typefaces, so it was an interesting time.

By this time, we had developed a type group. I had known Sumner Stone from previous work (ATypI and all) and introduced Sumner to Chuck. Chuck and Sumner hit it off beautifully. Sumner came in to run our type development group, and he brought in some very talented young people.

They actually produced some of our own designs. They're Warnock designs, and we used Chuck Bigelow's Lucida typeface. It was a wonderful group of talented people.

McJones: I remember the names Robert Slimbach and Carol Twombly.

Crews: Yes. Carol was a wonderful type designer. Charles Bigelow was an asset because Sumner was very connected with him. I believe his students came out of RISD (Rhode Island School of Design) and other places where Charles Bigelow had worked. He had also been at Stanford, so we had some people from Stanford. Cleo Huggins came in, and I think she was through our Stanford connection. It was a talented group of people, and Sumner did a wonderful job of positioning the company as not only the first to produce and have typefaces, but as also having a type program. Of course, there were other people at this time who were also starting their own type foundries that wanted to develop PostScript fonts. Adobe Type I format was absolutely critical, and then it became available.

McJones: Would other developers develop in your format?

Crews: Yes, we decided to open that up, so other typeface developers could develop in our format and we would have more on the market—more faces, more reasons to print, more reasons to develop good typography, and more reasons to use our printing engines. The type and the applications were just another reason for PostScript to become pervasive and provide an amazing solution for the industry.

In the beginning stages, it was common to have a document that you wanted to print on somebody's else printer, but it wouldn't print. You couldn't even send it. That was the building block for desktop publishing. If we hadn't had PostScript, the page-description language that allowed us to have a route for fine printing as well as proof printing and production printing, we wouldn't be where we are today. But to do all that, we needed the applications and the typefaces to actually do publishing.

McJones: So, type was a catalyst. It wasn't necessarily a huge profit center, but it established that everybody should have PostScript in their printer.

Crews: Yes, it did. I call type the glue that made it all happen, because if we hadn't had that, we just wouldn't be where we are today. Type caused the whole revolution, and I can't stress the importance of the fact that we brought quality to the market. It was absolutely imperative. When I joined Adobe, I agreed 100% with John and Chuck that if we didn't do it right, if we didn't do it with the quality it deserved, we weren't going to change the world at all. That was one of the reasons I joined Adobe, because they believed in fine typography and the point that's intellectual property. Those typeface designs are intellectually beautiful pieces of art. They should be respected and honored. They shouldn't be just ripped off like other organizations were doing.

GRAPHICS AND COLOR

McJones: Say a little bit more about the onset of graphics and color.

Crews: During my time at Adobe, it was all black and white. Color brings in another set of technological problems, so the question becomes, how do you manipulate more memory? How do you handle halftones? How do you produce color separations? If you're going to print on a press, you're going to have to produce four-color separations. When will we have a color printer?

I was fascinated by that next step being into color. Who was going to solve that issue? I was approached by Efraim Arazi, who was at Scitex Corporation. He developed Scitex and wanted to start a company in Silicon Valley. He had hired Don McKinney as his director or vice president of sales and marketing. I had known Efi from industry conferences, and he asked me if I'd help him start EFI, Electronics for Imaging, and be his vice president of corporate communications.

By this time Adobe had grown, and we had launched into Europe and had divisionalized the marketing communications activities to help support each of the divisions. John Warnock had said, "Liz, why don't you now be the spokesperson for the company?" So rather than running the marketing communications activities, I joined EFI.

Looking back, Adobe was probably the best job I ever had, but it was time for me to move on and find another problem to solve. How do you communicate where we're going next? At EFI, we were developing mapping tables to allow people to integrate into their software the ability to say, "I must have this mapping table if I want to go to a color printer of this type or the display has to be this type of mapping. Am I going to go RGB or CYMK?" We were dealing with those types of color mapping issues. The EFI had that technology with Scitex from Israel, so I went over to help him start that company, and we worked on getting that message out. I will never forget going to one of the conferences, Seybold or MacWorld. We wanted to show that color copiers were unstable and that you had to have mapping tables. We sent an engineer around to make color copies from different copy centers throughout the San Francisco Bay area. He came back with examples, and you could see how the colors had migrated. That was one of the ways we put together a demonstration to show people that, "This is a problem. We are going to have to work on this."

McJones: Did EFI make complete printers that included the mapping?

Crews: No. They just made the controllers. It was called an EFI controller; it went on to your color printer and gave your mapping.

McJones: Were you at EFI for a few years?

Crews: Yes, just a couple of years. At the beginning, when we started the company, the question was, how were we going to get funded? There was a general patent that Efi had from the Massachusetts Institute of Technology days. He used that as a way to acquire funding and suing people who used that patent.

McJones: Talk briefly about his background, about what Scitex did.

Crews: EFI came out of the printing and publishing industry. One of the things that they needed was a page makeup system that allowed them to place halftones, color, text, and graphics. In essence, a picture page layout system like we have today, one of the book composition programs where you place pictures and text. That was done on a high-end Scitex machine, probably in the 1980s. This would've been an example: if you had to emulate something on the desktop, you would have said, "I want that Scitex capability." That is what we now have today.

McJones: After EFI, you did a little consulting work.

Crews: Yes. Robert Norton, who was a type designer from England consulting at Microsoft, wanted me to look at their program and recommend what they could do, so I did. I recommended a type program for them and the faces to balance what Adobe was doing. At Adobe I had taken on all the responsibility for the public relations work in the beginning, as well as the advertising. So, when Microsoft was getting ready to do something, I went back to a Seybold conference in Boston, and I wanted some of those Microsoft people to meet certain people in the industry. I had a breakfast meeting and a couple of them didn't show because they were working up in their suite. I talked to my husband that evening. He said, "Are you having fun?" and I said, "Not really." He said, "Why are you there?" I said, "That is a good question."

I decided this consulting for companies was not the path I really want to go down. That's what triggered me saying, "It's a time to go take your expertise and skills and move them to some other area, like nonprofit work." That precipitated our move to New Mexico. I continued doing some technology work for the Museum of New Mexico Foundation because they had no online presence, and I worked on some ideas for them.

FINAL WORDS

McJones: Anything else you want to add?

Crews: I think if you maintain your sense of ethics, sense of balance, and never give up a sense of the best possible quality that you can produce, you'll go a long way.

I'm also delighted to see more women in the industry now, a lot more women. That's exciting and encouraging.

ENDNOTES

1. *Computer History Museum* "Oral history of Liz Bond Crews," Interview by P. McJones, Reference no. X8213.2017, 2017.
2. P. Baudelaire, "The Xerox Alto Font Design System," *Vis. Lang.*, vol. 50, no. 2, Aug. 2016; http://visiblelanguagejournal.com/issue/202

Paul McJones retired in 2009 from his career alternating between software research and product development (including stints at Xerox and Adobe). He is a Computer History Museum Volunteer in the area of software preservation. Contact him at paul@mcjones.org.

Interview With Larry Bohn

David Walden
Annals Editorial Board

Interleaf, inc., was founded in 1981 and was acquired by Broadvision in 2000. Larry Bohn was with Interleaf in 1986–1993, serving in a variety of senior management positions. This interview took place in Cambridge, Massachusetts, on September 24, 2019, at his place of work, General Catalyst.

Walden: Please tell me about your youth and education.

Bohn: I grew up in Milton, Mass. I went to public schools. My dad ran a luncheonette in Boston, and he died when I was 16 years old. He had been a World War II pilot, got shot down, spent a couple of years in a POW camp, so as a result of that I actually inherited his GI benefits, and so I was able to go to college. I went to UMass, majored in English, came out, went to graduate school at Clark University, and got a master's degree in English Linguistics while I taught writing. I taught writing for a couple of years at junior colleges in the Boston area and realized this was not something that I wanted to do for the rest of my life.

I had a couple of skills. I was a good writer, and I knew a little bit about computers from taking one Fortran programming course; so, I was able to actually get a job at Data General, which was just an emerging company in the area, and I was a software technical writer. I wrote books about programming computers, and that led me to my interest in document processing because it was really at the very birth of word processing, text processing, etc. I was at Data General for a couple of years, and then I went to Digital Equipment Corporation, where I managed a couple of groups. This was the time when some fundamental parts of the declarative markup language was being developed, called SGML, along with the way in which you could develop documents and structure documents, etc. I became very interested in that and ultimately went to Apollo Computer. Apollo was a workstation company. It was one of the first companies to integrate text and graphics, and so it provided the benefit of actually being able to show a document in what was called WYSIWYG form, What You See Is What You Get, and it was one of the first platforms at this startup in Cambridge called Interleaf—actually built its software on it.

Walden: At Digital, you said you led a couple of groups.

Bohn: They were the Operating System Documentation Group; I led that; and then I had a small development group that was building

Digital Object Identifier 10.1109/MAHC.2020.2968141
Date of current version 6 March 2020.

some software to automate the text processing business.

Walden: Before we move on to Interleaf, I read your website where it talks about your unusual path to becoming a venture capitalist: delivering groceries, driving a taxi, working on a farm, being a short-order cook, and managing a pool hall.

Bohn: That was before and while I was going to college. And then, when I went to college I did work on a farm, sadly, in exchange for housing, and I managed a pool hall in college.

I was sort of the scrappy guy. I had to sort of find my way early on and help a family whose dad had passed away; so I would say I learned to work hard early on, and that has benefited me throughout my life.

Walden: So at Apollo you were already seeing text and graphics.

Bohn: Yes. Just to back up: I learned a lot about document processing at Digital because it was at the beginning of when Donald Knuth, who was a famous computer scientist, developed what was called TeX, and TeX was a programming language for documents. I learned a lot about that and was very interested in the whole way in which computer markup worked, typesetting, and the interface to typesetting. It was a very proprietary world that was starting to open up, and I became very fluent in the technologies that were emerging around document processing. I then went to Apollo. The thing about Apollo was it was the first workstation with a big screen. You could see a document, and Apollo was very interested in moving the whole documentation process online so people could retrieve documents electronically. I actually led a project with a small development group that was in the document retrieval business, using some underlying technologies around TeX to do it. Then, we found Interleaf, a startup company in Cambridge—very small. It was a handful of people, probably ten people, but it was building something very, very advanced. I got to know the founders.

Walden: This is while you were still at Apollo.

Bohn: This is while I am still at Apollo. I met the founders of Interleaf, Dave [Boucher] and Harry [George]. I met the development team, Bob Morris, Steve Pelletier, etc., and I negotiated a license deal between Interleaf and Apollo so everyone at Apollo could use the Interleaf software. It was a big thing for Interleaf, and it was a big thing for Apollo.

Walden: Early software before any real release?

Bohn: No, it was just being released, so I helped them get Apollo systems. Originally the software was developed on Sun's Microsystems, which was a competitor. I helped fund the effort for them to port the software to Apollo, and then we used that within Apollo. About a year later, I actually talked to Dave and Harry and joined the company as head of product planning at Interleaf. The first job I had there was Vice President of Product Planning, working with the development group on sort of where the product should go, how it should be built, what the market was, etc.

Walden: What technologies were there from the TeX world that you used? Do you remember by any chance?

Bohn: TeX was a very low-level language for document processing, but there was a higher-level version called Scribe, and Scribe was a more declarative language. I think it might have even used TeX. Brian Reid developed it, and it became very popular, and it was something that we started to use at Apollo to do our documentation and to use it so that we could produce both high quality copy and online versions. So these were the main sort of developments in document processing at the time; and so when I went to Interleaf, Interleaf was very different because Interleaf was a completely interactive system. It did not rely on any sort of low-level language. It did not rely on actually putting declarative markers in a document. It had a user interface that allowed you to create what are called components, and so it was really in many ways the first interactive structured document editor.

Walden: My understanding is that some of at least the Interleaf prototype came from the Etude project at MIT.

Bohn: A number of people from the Etude project, which was Mike Hammer's project. That was a project in office automation. The document processing was one of the central parts of that. I would say that was the basis for Interleaf. The inspiration came largely from Xerox PARC and the Star document editor. The Etude project was sort of the foodstuff of the first Interleaf version.

Walden: One of the things I have been trying to understand is that while Etude was written in CLU at MIT, I believe Interleaf's original project was written in C. Do you know who did that conversion?

Bohn: I cannot say who actually did the conversion. Most of the Interleaf people who were developers were very strong C developers, and so the original product was written in C. Ultimately one of the things that Interleaf innovated on was building an interpreter as the customization layer. It was an AI language called Lisp.

Walden: So, at the very beginning, the programmers, as far as I can tell, who were at the company to for instance get the first demo going in 1984 were Bern Niamir . . .

Bohn: He came from MIT. He was sort of the person who came from the MIT project, but he was not the lead developer.

Walden: Other relatively early programmers were Jim Crawford, Steve Pelletier, and several others that Pelletier brought in, such as Mark Dionne, Kimbo Peebles-Mundy . . .

Bohn: What happened is Pelletier had worked for a company out in Colorado, and he had recruited Kimbo and Deborah Landsman and Kimbo's wife . . . to the company because they had worked on a word processing system. They brought that sort of heritage, and they were very strong developers. Crawford came from Harvard. He was a brilliant developer.

Walden: Do you have insight into why the founders, David and Harry, chose a publishing system as their product?

Bohn: They were very interesting guys. Harry was sort of a poet by background although he later became the CFO, and Dave was—I think he might have been an English major at MIT. He was not a hands-on engineer. But they had done a lot of research in office automation, and I think the relationship with Mike Hammer was fundamental to understanding that there were a lot of problems in document processing around cutting and pasting text and graphics together, and so they saw that that core innovation—being able to take text and graphics on a page and show it—could provide real benefit, and that that was fundamentally enabled by high-performance workstations.

Walden: I have read in one of the Seybold Reports that when Dave Boucher was invited to an early Seybold technical publishing meeting, he turned it down because he said, "We're not a technical publisher. We're in the graphic arts business." And then a year later, he said, "Whoops, we are in the . . ."

Bohn: What happened to the company, which is true of a lot of startups, is that they basically believed that they were creating a new kind of product that could be a widespread product basically like a super word processor. Some of the fundamental benefits were: it was WYSIWYG, it was easy to use, it was highly interactive. But one of the things they failed to realize was that the cost of deploying an Interleaf system was so high because it ran on a 32-bit workstation, which cost $80 000. That relegated it to a sort of specialized high-end market. The first market of the two early markets for Interleaf was the graphic arts business and companies that were doing directories and things like that, so Donnelley was a customer—big publishers like that. But the bigger market was what I would call it the military specification technical document market. The reason for that is, you have to understand, that the software only ran on high-end workstations. Most of the high-end workstations at the time were being sold into defense contractors around electronic CAD (computer aided design) because this was the Reagan buildup at the time—Reagan's Star Wars. There was a massive investment in the defense area, and a lot of it was around embedded electronics, so these workstations were actually being used for ECAD [electronic computer aided design] to design circuits. They were also needed to produce all the complex documentation around these embedded systems, so all the people in the military who were producing these Mil-Spec technical documents saw the potential of the Interleaf system, and they had a lot of money, so they were more than willing to fork over lots of money for both the workstations and the software. That basically guided the company to focus on the technical documentation market.

Walden: There also were big companies that need documentation.

Bohn: Yes, exactly; so basically Interleaf moved from sort of an office automation orientation to much more of an industrial one.

Walden: As I understand it, the very earliest Interleaf systems had some Interleaf hardware in them like an Interleaf scanner of some sort.

Bohn: Yes.

Walden: I think that Jon Barrett developed some kind of a raster image processor which was licensed out. Can you say something about the idea of building their own hardware in addition to the software?

Bohn: So, the original Interleaf business model was to build their own workstations. And what Dave and Harry realized very early on—it was a very smart decision—was that they were not going to be able to compete in the dedicated workstation market and that Sun Microsystems was coming out with the Sun-1, which was the original workstation, and that could be a platform for the software that would make it much more economically advantaged. But one of the core investments that they made in terms of the technology was in raster image processor [RIP], the thing that put together a page on the screen around a printed page; and one of the things that they saw would be an opportunity would be to actually build a printer, a laser printer, that used that RIP and could produce documents at very high performance. We actually did a deal with Dataproducts. Dataproducts was a big printer manufacturer. They were a laser printer company, and so they licensed the Interleaf RIP to embed in that printer. For the first handful of years most of the Interleaf customers bought a custom printer from Interleaf that had the Dataproducts printer along with the raster image processor.

Walden: Do you know how the founders got connected up with Bob Morris?

Bohn: Morris was teaching at UMass. He was a pretty well-known professor of computer science. He was also a developer, a pretty good developer, so he came. I do not know how he was recruited into the company. He was in the company when I came.

Walden: Regarding the founders, what do you see as their strengths and the different roles each played over the years?

Bohn: Dave Boucher was sort of a visionary, and I think he had a great vision for the company and the technology and what it could become. He was not, I would say, a great hands-on manager. He was much more of a product visionary. Harry [George] was a very good fundraiser, and we needed to raise a lot of capital to fund the company, and he was very good at that, and he became the CFO in the company. The other person who was very notable early on was George Potter, who was in charge of sales. They recruited George from Wang, where he sold word processing equipment, and he was a very sort of aggressive, boisterous sales exec and did a great job landing the first handful of accounts and OEM deals, etc. One of the things that is sort of notable about the company is Interleaf developed a somewhat problematic relationship with workstation vendors early on because the product was really impressive. It was a great product, and all the workstation vendors wanted to have it on their system, and so the company would work with these workstation vendors to have them pay a lot of money to port the software to these different workstations. The problem was that it created a huge amount of channel conflict because all these companies were selling to the same end-user customers, so someone like Boeing would have five different vendors trying to sell it Interleaf software. The most notable of these was Kodak. The first big OEM deal that George did was with Kodak, and Kodak literally bought a full perpetual license to the software to run on their workstations, and that did bring a big revenue stream into Interleaf, but it meant that when Interleaf was starting to sell its own product directly it competed with Kodak everywhere, and so it was a really difficult situation.

Walden: Was that product stream different than the investment that Kodak made in the company?

Bohn: No, it was part of that. They made an investment in the company, and they did a commercial deal in which they had license to sell the Interleaf software on Sun workstations with their own printers, etc.

Walden: Were you at the company at the time of the IPO?

Bohn: I was there. They had two IPOs.

Walden: I have heard that the first IPO was rescinded.

Bohn: Yes. When I came to the company, part of the reason I came was they needed some management. I had some management experience, and this was at a time when the company had just been awarded a huge government contract; the contract was out of the Army, and it was called 600-S, and this was a huge multimillion dollar contract that really made the company and allowed for the company to plan for an IPO. The government contract was to automate the way the Army did technical manuals, so you could see this as multiyear, millions of dollars. The system integrator was EDS [Electronic Data Systems]; EDS won this award to automate the way in which the Army did technical manuals. We were a subcontractor, but we were going to get rich off it; so, as we won that contract, we filed to go public. What happened is that there was an impropriety in the way in which EDS met with or solicited some of the Army officials. They basically met in violation to what they were supposed to do. The whole contract was rescinded, and it was rescinded on the day the company went public, so that is why the whole IPO had to be rescinded because the basis of the IPO was gone. This caused a huge storm in the company. There had only been one IPO that had been rescinded before that; but to the company's credit, it readjusted its numbers, and it had a strong business and was able to go public for a few dollars less a share a few weeks later—I think it was not even a month later. Ultimately the company did go public, but it did not have that backstop of the big contract.

Walden: You said that it caused turmoil in the company. Was that at all levels? Were the employees were looking forward to this?

Bohn: Yes, because it was a big event, a big celebratory event, so when ultimately it was rescinded it was like, "What happened? What do we do now?"

Walden: Was there despair or did everybody know that it was going to be taken care of?

Bohn: There was definitely despair, but I would say the company handled it pretty well and was able to go public despite the loss of that contract.

Walden: Did the IPO hiccup matter much to the financial markets or to the customers?

Bohn: I do not think it mattered much to the customers. The company was valued less than it would otherwise, but the company was able to go public because the company had a strong business. It really did.

Walden: Once the IPO was successful, did the company change?

Bohn: Well, the company grew quickly, and the company was under pressure to make numbers. The company I would say was generally successful in the public markets although there were times in which the company did miss its numbers. As the company grew it became clear that Dave did not want to be the CEO of a public company. Probably the biggest thing that happened in the company after it went public was that we recruited in a terrific guy named Dave Collard to be the CFO; Dave had been the CFO at Prime Computer, and when he came into the company he looked at the cost structure of the company. You have to understand, Interleaf through the time it went public was a systems company. It would sell software and hardware together with printers, and they did that because there were few ways you could distribute workstations because you could not buy them in a computer store. The only place you could get them was directly from the manufacturers, and the manufacturers were focused on selling them through OEMs like MentorSystems and Cadence, and Interleaf was an OEM. We basically bought the computers, loaded them with software, and shipped them to customers. During that period after we went public, the distribution model changed a lot, so people were able to buy hardware directly from the manufacturers rather than through OEMs, so it meant that it was very hard for a company like Interleaf to charge the premium for a system, so when Dave Collard came in he looked at the cost structure of the company and realized that the company should be restructured: it should get out of the systems business and just sell software. What is interesting is the company's revenues, which were growing very fast, continued to grow but not as fast as before, but the margins for the company increased dramatically, so overall it was a very positive thing, but it was a big shakeup in the whole company because we ended up laying off a lot of people in manufacturing, etc.; we had built a big manufacturing facility.

Walden: Before this shift, there already were lots of direct sales offices around the country and in certain parts of the rest of the world. Were they selling to OEMs or were they selling directly to users?

Bohn: They were selling directly to users.

Walden: You had these OEM channels, and also you had all this direct sales force.

Bohn: Yes, there was a lot of conflict, and part of it was that when you are in the systems business you are selling a two or three hundred thousand dollar system to a defense contractor in Los Angeles, and you have to support that system as an OEM, so you have to have hardware technicians, and it is a very expensive operation. In the early days you had to do it, but as there was more distribution of the hardware directly from the manufacturers it became uneconomical, so there was a big changeover. Interleaf was one of many companies in this transition. All the ECAD players went through the same thing, and so ultimately the restructuring of the company was a huge shakeup to the organization, the business model, etc., but the company actually came out the other side in a much better way.

Walden: In 1990 when this transition was sort of happening the company lost sixteen million dollars. Was the problem only that revenues and expenses were not well matched, or was there also a big restructuring charge?

Bohn: I think it was probably the restructuring charge then.

Walden: What other roles did you play in the company? You started, you said, as Director of Product Planning.

Bohn: I became Vice President of Product Management and Planning. I became VP of Marketing at one time. I sort of managed a lot of the peripheral parts of the development group, the groups that did the porting and other things. I was one of the senior execs that sort of managed a lot of parts of the business. At a certain point in the company's history, Dave Boucher brought in Bob Weiler [who replaced Boucher as president and CEO in 1990], who was an exec from Lotus, and it was at a time when we were trying to concentrate on certain markets. So, I led an effort to focus on the aircraft industry to develop the product more in line with new requirements that were coming out of that. That was pretty successful.

Walden: People I have told that I was going to interview you have said, "He's a great guy."

Bohn: Well, that's nice of them.

Walden: Can you tell me something about your theory of managing people and technology and so on?

Bohn: In my career, I always worked with very, very smart engineers, and I think I learned that there are ways in which you can work with engineers and there are ways you cannot work with engineers. It is very hard to tell engineers what to do. You have to sort of work with them around common goals, and at the end of the day you learn a lot of management skills around how do you organize people to get the work done as a company and at the same time get the work done of their interests, and so I would say I came into Interleaf as one of the first outside managers. And I think I was successful because I had good interpersonal skills, I could communicate and really understand both the side of the engineers and the side of the salespeople. I was right in the middle, right? And in the tech business it is not like a factory. In these companies, the brains run in and out every day, so a very important part of managing a technology organization is try to find alignment across the different organizations, especially in the technical roles. The way in which you manage technical teams is quite different than the way you manage sales teams. Those teams are very coin operated based on incentives of making money. Technical teams are in some ways very based on meritocracy and the ability to execute, create creatively, and usually there is a lead developer that can inspire and shame developers into doing great work, and Pelletier, who was the VP of Engineering, was brilliant at that. He was very good at managing people to both their own interests and to the company's business.

Walden: I have been making kind of an approximate list of Interleaf's product offerings over time. Can you say something about Interleaf's product strategy?

Bohn: One of the things about Interleaf's product strategy is that you pay a penalty for being first to market, and the penalty that Interleaf paid was that it literally needed to invent itself all the components of the system. You know today, with open source software, you assemble stuff. Everything's already been built. You just reuse it. In contrast, one of the first groups I had to manage was literally the font group. I had a group of graphic artists who were developing typefaces for the product. Now, the amazing thing was that, fast forwarding three or four years, you could go to the computer store and buy fonts for, you know, $30, right? There was a massive change in the availability of underlying technology to support document processing. But Interleaf had to develop its own proprietary user interface, proprietary font system, and proprietary window system. As it was developing functionality for just core document processing, it was also in the process of having, at different times, to unbundle the things that they had built previously to use more standard fonts, windows systems, and user interfaces. And the interesting thing was that the classic point was when we did the ports of the Macintosh. Steve Jobs was really excited about Interleaf porting its software to the Mac because we were the leader in this high-end desktop publishing—he really wanted to own that. But when we ported our software to the Mac, the first version had our own user interface. They went nuts because everything else on the Mac was the seamless experience and everything looked the same and here was this different Interleaf user interface. So, we were under wicked pressure to reengineer our product to work within the framework of the Mac user interface, which took a long time.

Walden: Back to the fonts for a second, you said you had developed your own font. Do you happen to remember what font technology you used?

Bohn: What we did was we literally had artists who would do pixel placing. They would literally draw fonts on the screen. It was our own proprietary technology. They were raster fonts. Ultimately, we did go to Bitstream; we licensed technology from Bitstream.

Walden: Can you say something about the competition over this time?

Bohn: Here is what I would say that happened in the market. When Interleaf first came to market and had a certain belief that it was in the office automation market, it realized that it could not compete in that because of our system requirements. So, the word processing market went to Microsoft and Microsoft Word, and in some ways, people will say that Word was a poor man's imitation of Interleaf. Our system did handle a bit more structure. It ran on a PC and, at one point, we did port our software to an IBM PC. We did a big deal with IBM. I think it was an early version of IBM's PC, but the problem was that the PC required—it required so much memory that it was just out the league of most companies so . . .

Walden: Was this the IBM RT or something else?

Bohn: No, the RT was a UNIX workstation. We also attempted a port to OS2? The first IBM product ran on DOS with an extender—a memory extender. There is a company—I think it was from Phar Lap. They had a memory extender so we were able to run under Microsoft's DOS with a huge extender for memory, but it was sort of kludgy because it required so much extra memory that no one could afford it.

Walden: In the second *Annals* special issue on desktop publishing, there was an article by one of the Frame founders, and he says, "We directly targeted . . .

Bohn: Interleaf. Yeah.

Walden: . . . Interleaf."

Bohn: Totally.

Walden: The Frame founder said, "Interleaf was the market leader and we went after them."

Bohn: One of my partners here, David Orfao ran sales at Frame. That is how I knew him. The founder of Frame was a brilliant British engineer, Charles Corfield. One of the things he realized was that by the time that Frame entered the market, there was enough of a substrate of window technology, font technology, etc., so you could build on top of that, and build an easier-to-use, lighter-weight product that you could sell indirectly and compete very effectively with Interleaf. So, what happened is when Frame came after Interleaf, it came after the segment, which I would call the sort of secondary segment—the occasional user. It was not the dedicated publishing group. It was much more the engineers who are doing ECAD, who needed to do specs, etc. So they did a lot of OEM deals with all these CAD vendors, and in some ways it really ate away at the bottom end of the Interleaf market. So that forced Interleaf to focus much more on what I would call the long complex document market, which was much more the Mil-Spec—hundreds of thousands of page documents, etc. They did good job. They targeted us. They undercut us on price.

Walden: But ultimately Moore's law made the small hardware more powerful so somebody could take that business away, too. Yes?

Bohn: Yeah. The high-end of the market Interleaf tended to own because it became a sort of a very specialized market because of the features and functions and it was defensible, but Frame did a very good job taking away a bunch of the commodity level market.

Walden: When we were talking a little while ago about distribution and direct sales versus OEM, was that the same internationally, as well? There was both direct sales and OEMs.

Bohn: There was. And early on, you know, we expanded a lot to Europe. We did a deal with Japan. We expanded internationally very quickly. Probably, it was a mistake, and we set up offices and the offices were expensive and the personnel were expensive. So we invested a lot. You have to understand, the early—in the system's business—setting up all these offices, etc.—you had to hire field service people. It was a nightmare.

Walden: Did the customers pay separately for field service and maintenance and all of that or was that somehow bundled with the price of the product?

Bohn: No. Basically you would buy the product and then there would be a maintenance plan.

Walden: And was that a useful, stable revenue stream?

Bohn: It was. Everyone bought it, but I would say it was not sustainable as a business model long-term.

Walden: What do you see the key mistakes Interleaf made that caused it to have problems? I think you sort of said that they were early and innovative, and the world changed out from under them too quickly.

Bohn: I think the key thing was the company did not anticipate how quickly standardization would come into the market and this was standardization on things, like, you know, page description. Adobe came in and changed the world in terms of, you know, PDF, etc. Motif and Xwindows, which was a workstation windows system, came in and took over. You know, fonts came in. So the company was too tied to its own technology, which in many ways was superior. This was true of, you know, if you follow the history of Apollo computers, it is very similar. They built the wrong windows system. They built their own . . . And some of this people claim was sort of an East Coast liability—that companies on the East Coast would be very good a building fundamental technologies but hang on to them too long as standardization came into the market, and Interleaf was not quick enough to abandon what it had built and move on to standardization.

Walden: But, of course, you had this big legacy installed base that presumably, wanted to be supported.

Bohn: Yeah. So that was part of it but I would say the company was just not agile enough to make those transitions quickly enough to meet the market.

Walden: Did Interleaf have a user organization or more than one of them?

Bohn: Interleaf had a really big user group—a very passionate user-group. Yeah. A lot of users were, say, these sort of young women who sat in front of terminals and did document processing. They loved the product. They just loved it. There was very, very high satisfaction with the product, especially in the early days. And there was an international user group . . .

Walden: And the company supported that?

Bohn: Very much so. You know, it is one of the things I tried to do is to really be close to customers and see where customers were going. The Interleaf user group was both a real asset and a demanding part of the company. They were demanding to the company about new features and future developments.

Walden: What you say about "demanding to the company," being in this very high-end business, or quite high-end business with companies doing different kinds of things, I assume that each company was asking for modifications to address its detailed needs.

Bohn: Well, let me give you a couple of examples because, in some ways, I learned very interesting lessons in the tech business when I was at Interleaf. As Interleaf emerged into more demanding markets, like, the Mil-Spec technical manual market or the pharmaceutical market, these markets had very specialized requirements.

So, for example, in the Mil-Spec technical market, there was a security clearance requirement so that if you had a certain level of security discussion on a page, you had to indicate it up in the top, right part of the page. Every page was earmarked with what level of security. So, this was a super-important feature for people who are doing these documents because you have got thousands of pages and trying to keep track of where the security clearance is. I remember working with the development organization, pushing them to solve this problem of Mil-Spec technical manuals and to make it so that people who were documenting these pages did not have to worry about how the documents would get published with this upper right-hand security mark. It took forever. I fought with the development group to do it. They finally did it, and it was, like, a knife through butter in the market. As soon as we had that, people ran to us. It solved such a big pain point, but it was a very specific vertical market feature for the Mil-Spec market.

Similarly, in the pharmaceutical market, they had a requirement, which was called the "big page number," which means that when you do an NDA, a New Drug Application, you are putting together hundreds of documents into literally a million pages, and what happens is you have to, at some point, repage the entire collection with a big page number on every page. Again, this was one of these things that you see when you are building out this big published set. To do that was sort of complicated and hard because it meant that you are taking collections that had been paginated. You have to repaginate it. Finally, we did it, and again, the same thing happened. It meant that it was so easy to sell to pharmaceutical companies because we had the big page number problem solved.

Walden: And is this an example, as well as, of the fact that sometimes you have to cajole the engineers?

Bohn: Yes. They hated it. They hated the idea of doing it.

Walden: Do you have any insight about the effectiveness of the board over the years of Interleaf—from founding to acquisition?

Bohn: I feel Mike Hammer was helpful. Mike was a great guy and he sort of was an enthusiast on the board. The others, the investors, were from Advent. Advent was the biggest institutional investor, and there is a guy named Clint [Harris], who later ran another fund. I think he was helpful. But George Potter was on the Board and Harry and Dave. There were no real industry outsiders on the Board, which I think was a limitation. I would say, I think the Board it did its job, but I do not think it was super influential.

Walden: There was a succession of CEOs. Bouchet, Weiler, Rupert, Koepfler . . . there was Rory Cowan on an interim basis before Ellertson came.

Bohn: So Ellertson was a workout guy and he was famous for doing these workouts.

Walden: Workout means what?

Bohn: He will take over a company that is sort of in distress. He knows how to rework it in the public markets. He is a money-maker.

Walden: So getting the big price for acquisition . . .

Bohn: He did that. Yeah. He did a very good job. The other people I would say . . . Weiler was a pretty interesting guy. He was energetic. He dealt with some of the issues. He was only there for a year. Rupert was his sales leader, and he was very much a sales guy, and in many ways, he became CEO and he alienated many parts of the company.

Walden: Then there was Koepfler.

Bohn: I do not know what his background was. I think I left when he came.

Walden: Many people, primarily engineering kind of people that I have been talking to because that is my connection, have said that Interleaf was the best place they ever worked.

Bohn: It was a great place.

Walden: And my question is, what is your perception of the culture?

Bohn: It was a very product-driven company. Very innovative company. It allowed engineers to take on projects on their own and to develop ideas. The benefit of that was that out of this sort of, you know, petri dish, there was a lot of really interesting technology that was developed. The downside was, if you look at the company, it is engineering expense was off the charts. It spent way too much on engineering and when you look at a standard public company; but it was a very high performance development team that challenged each other. It was very much a meritocracy, and I think Steve [Pelletier] and others led it that way, and if you were not up to the task, you were gone. In some ways, it was very demanding place but it was a very energetic place.

Walden: Was there a collaboration or friction among the functional organizations?

Bohn: I would say there was both, but there was some good friction. You know, the sales people wanting certain things. The developers not wanting to do them. There was that classic tension. I was often in the middle of that.

Walden: Mark [Dionne] mentioned that early on, the software engineers did the product design. Do you have an opinion of how things went once product managers came on the scene?

Bohn: I think it was in the classic case as the product developed. It was hard for engineers to be close enough to customers to really know what to build. Product managers came in. It was also at a time when the market had changed so more standardization was coming in. So, there was definite tension between the product managers who said for instance, "We need to support this PostScript printer because everyone's supporting it," and the developers saying, "Well, we have this RIP printer; it is five times as fast."

Walden: You mentioned, for instance, the benefit of building a small thing—maybe hard—but small that the customer really needs such as in the security area and the big page number area as something you learned at Interleaf. Were there other lessons from Interleaf that helped you in your later business?

Bohn: One of the things I learned, is that you have to be very careful about too much pride in authorship. In other words, you know, you develop a product and you put your heart and soul into it, but you have to understand, the markets evolve very quickly, and if you cannot adapt to market changes, you are going to die. Some of what I learned is that. You know, I keep relearning this—that what works today is likely not to work in a few years, and you have to really anticipate change and embrace it. So that is one lesson I learned, and the other one I learned is that you really have to stay close to customers. At this level. at the application level, you can invent things and you can be ahead of customers but at a certain point, you have to satisfy their needs, and usually, an early technology solves a small, but important problem, but overtime, you have to solve more of the problem. You have to be close enough to customers to both anticipate what they are looking for and to be able to develop it in time.

Walden: How did you come to leave Interleaf?

Bohn: What happened is I had been there about seven going on eight years. I had done a lot. I had been with the company from the early days. The company had changed a lot. It was becoming slower growth. Weiler left, a new CEO came in, and it was clear that I had done about everything I could there. And by that time, I felt like I had developed enough confidence and I said I want to run my own company, and I got recruited by a company in the document management market called PC Docs, and this was in the network document management market. One of the things that happened, if you look at the history of publishing, is that the first generation was all around creation systems—tools to create documents; and then the next generation was around managing them. So, there were companies like Documentum and PC Docs, and we [at Interleaf] had a product called RDM [Relational Document Management]. So RDM was a document management product but it was a pretty clunky product, and we competed against a company called Documentum, which was a West Coast start-up that did very, very well. PC Docs was sort of in the volume end of the document management market around the emergence of PC networks. I wanted to get into running a company, the software business, application level in the volume business, and so this was a really good fit, and so I was recruited to be president of the company. I think I started out as the executive vice president. I became president of the company and ran the company. It was actually a division of a Canadian holding company that was in several different products in the legal software business, and one of our biggest markets at PC Docs was the legal business. I ran that for a few years. We took the company public. It was a very successful company, and then toward the end, we had some acquisition offers that made a lot of sense and I wanted to do it and the chairman did not, and we decided to part ways.

Walden: After you left Interleaf, did you keep following what was happening up through the acquisition by Broadvision?

Bohn: Pretty much. Yeah, not super-closely. But the company, I think, went into sort of a moribund state because it was, you know, flat growth, the public market. What happened is that Interleaf, in some ways, anticipated but missed the Internet boom. I remember looking at Mosaic at Interleaf when Mosaic, the browser, first came out, and I think there were a lot of opportunities for Interleaf to participate in sort of an early Internet applications, but for a lot of reasons—people had left—the company had lost a lot of innovation. It did not really innovate through the first internet wave of technology. So what happened is Broadvision, which was in the website development business, was a big company. It grew and at a certain point, it too had run out of gas, but it could buy. One of the things that Interleaf had was all these customers, and all these customers needed websites, as well as documents, and so it bought Interleaf and it paid a good price.

Walden: $840 million.

Bohn: $800 million, yeah, which is a good price, which sort of saved the company. It actually saved the company because I think Interleaf would have just declined after that. It was a very good outcome for the shareholders.

Walden: When you meet people and they learn you were at Interleaf, do they know about it?

Bohn: Today very few people remember Interleaf. But people who worked in Boston at that time and anyone who was at Interleaf, really remembers it fondly because it was one of the really two software companies in the Boston area that was very notable. It was Lotus. It was Interleaf. Really, those two. The ex-Interleaf list called "Interleft" was an active mailing list for 20-something years.

Walden: Was Interleaf already at Canal Park in Cambridge when you joined the company or it is still on Mass Avenue?

Bohn: I joined on Mass Ave. and made the move to Canal Park. (I invested in HubSpot, and HubSpot is in Canal Park now.)

Walden: Let us talk about after Interleaf, I read again your website about you like building companies and you have had some successes. You mentioned one of them that you went to after Interleaf.

Bohn: PC Docs.

Walden: What else happened?

Bohn: So then, I was at PC Docs for a few years, took it public. We ended up selling the company. I left. I took some time off and then I took over a company called Net Genesis. Net Genesis was an MIT company—very early company. I took over CEO in I think 1998, and took the company public in 2000, and it was one of the first web analytics companies. It would analyze logs and tell you what visitors were doing on websites—like Omniture and companies like that. I was a very early pioneer in the web analytics market. It was a good company. Young founders out of MIT. You know, the "go-go" days of the Internet. It was super fun and exhausting.

Walden: You mentioned that the company you went to immediately after Interleaf was in kind of the document business. Were any of the others in the document world?

Bohn: No. I went from Interleaf to PC Docs to NetGenesis to here.

Walden: Can you tell me what you do in the rest of your life, besides work for General Catalyst—hobbies and so on?

Bohn: I am an avid biker. I am a 21-year rider in the Pan Mass Challenge and I play a little golf. I read, I travel, have two kids who are grown up now, which is great.

Walden: Is there anything else that you would love to speak about that I have not asked?

Bohn: Not really. I think this is a pretty good history. When I look back at Interleaf, I remember a company with brilliant promise. An example is, Interleaf and Adobe sort of got started about the same time, and it is a great example of West Coast/East Coast and focus. Adobe, you know, was a very focused company, built, you know, the page description language. Used that as an OEM—OEM printers—then build on that. Today, it is one of the biggest software companies in the world. It built a market around its technology and expanded it very, very effectively. Interleaf came out with a product that was brilliant—everyone will say it is brilliant—but it was very proprietary, and the proprietary nature of the product prevented it from becoming as wide-spread and adaptable as it would need to build a huge company. It is so true of companies in some ways on the East Coast. You look at DEC and you look at Apollo and people have long commented on this is—that the East Coast mentality was very much systems oriented, proprietary technology, very advanced technology. The West Coast was very components oriented, standards oriented. At the end of the day, the West Coast model became more adaptable and sustainable. And so, while Interleaf had really good success, especially early on, it did not endure. It did not endure and there is some sadness about it, but it was a great company. I loved working for it and had I not worked at Interleaf, I could never have had the career I have had as a CEO or Venture Capitalist.

Walden: Years ago I heard someone say, "The VC world in California is just much more willing to take risks and so on than the VC world in New England."

Bohn: I think that there is some truth to that. Absolutely, but, you know, VCs do not make companies. Entrepreneurs do, and I would say if you look at what happened, you know, there's a famous book by a woman who was a professor at Berkeley. It is about comparative advantage [Annalee Saxsenian, *Regional Advantage: Culture and Competition in Silicon Valley and Route 128*], and it talks about the history of the technology business on the East Coast and the West Coast, and if you look back to the 1970s and 1980s, both coasts were very much the same. They were very defense oriented. The culture our of Stanford, the culture out of MIT. The change was that the semiconductor business developed very much out of the West Coast and it was a components business, and the East Coast developed into a systems business. You know, like, DEC, right, was building its own computers, its own chips, etcetera, and that orientation towards integrated systems versus assemblable components showed up in a lot of different ways. So it showed up in Interleaf. In other words, you know, Interleaf was building all of its own pieces, and the trouble was that at a later stage of life, it ended up having to unbundle itself to reinvent itself and that was a big cost; whereas, Frame and others were able to take advantage of these technologies. It is not so much the VC; it was sort of the orientation, and I think there is still some of it. It led into why the West Coast built great Internet companies.

Walden: This has been fascinating. Before I knew only a little about Interleaf—the company eventually was in a building I would drive by on Route 128, in Waltham, with "Interleaf" on the side of the building; also, I believe we had an Interleaf system at BBN.

Bohn: You did.

Walden: Okay. Thank you very much for taking the time to do this interview.

Department: Anecdotes
Editor: David Walden[†], dave@walden-family.com

Interleaf, Inc.—1981 to 2000

Mark Dionne and David Walden

Interleaf, Inc., was an early entry into the computer-based publishing business, initially based on computer workstations and focusing on technical publishing. The company was formed in 1981, shipped its first product in 1984, which included a variety of hardware and software innovations, and was in a strong position to go public in 1986. In the later 1980s, other companies entered the business in competition with Interleaf, and personal computers then had the power that once required an expensive workstation; there was also standardization around window systems such as the X Window System, Motif on top of X, and Microsoft Windows. Also some of Interleaf's innovations were no longer competitive advantages as the industry coalesced around alternative technologies such as PostScript and Acrobat Reader. Redirecting its business strategy once and then again, Interleaf continued on with up and down success until 2000 when it was acquired by BroadVision.

The materials in the Interleaf archive at history.computer.org/annals/dtp/interleaf may be useful as one reads the story below. Also at history.computer.org/annals/dtp/interleaf/anecdote-additional-notes.pdf is additional narrative that did not fit within the *Annals* page quota along with less important notes and references; the places where these "Webnotes" would have been located in the main text are marked with successive superscripted lowercase letters (after the letter z, aa, ab, etc., are used).

EARLY YEARS, 1981–1984

Both David Boucher and Harry George had been with Kurzweil Computer Products company, which was acquired by Xerox in 1980. They were interested in starting their own company and looked at a variety of different possible businesses. While temporarily helping someone else with a plan for a new business, Boucher looked into workstation applications including electronic publishing. Boucher and George then broadly investigated electronic publishing for themselves.[1,a,b] This revealed the potential for companies to replace traditional manual integration of text and graphics into documents; money could be saved by using workstations to combine text and graphics into a high-quality printed documents. Boucher says that for their business plan they focused on consulting firms, where

Digital Object Identifier 10.1109/MAHC.2020.2968198
Date of current version 6 March 2020.

[†]This anecdote was edited by the issue guest editors.

they could calculate the potential average dollar savings per page over manual methods which would be compelling data when seeking financing. Boucher and George started Interleaf (as in interleaving text and graphics), Inc., in January 1981. Boucher was the President and George was Chief Financial Officer, and their equity interests were equal.

Boucher says that initially they believed that they had to develop a workstation and appropriate printer, and to begin work on these they hired Joe Barrett and Allen Anderson. Two other early hires were Jim Crawford, a biochemist but also an exceptional computer programmer, and Robert Morris, a professor in computer science at the University of Massachusetts at Boston, who had a great interest in typography and who took a leave of absence from UMass to join the company.[2]

Boucher says that they always knew that the software for their workstation would be the critical component and looked around for software systems that could be relevant to Interleaf's new business. They heard about the Etude system being developed in Professor M. Hammer's office automation research group at MIT. Etude was aimed at increasing "the functionality of office document production systems ...while *reducing* the complexity of the user interface."[3,4,c] The Etude system ran on a DECSystem 20, eventually using the NuMachine (a prototype microprocessor-based, networked workstation) as a graphics terminal.[d] Boucher says that the moment he saw Etude, he knew this was what Interleaf needed—"it was a different concept of what a word processor could be." Professor Hammer (increasingly well known as a business consultant) also joined the board of directors of Interleaf.

One of the programmers on the Etude project at MIT was Bahram Niamir (known informally as Bern) who had converted a subset of Etude from the CLU language to C and also made it run under Unix and added mouse input. He called it Ecrit. Niamir officially joined Interleaf in July 1982 (about the same time as Crawford and Morris), but had been working for a few months already producing the basis for the Interleaf system from Ecrit.[5] Steve Pelletier remembers that when he arrived at Interleaf in early 1983, "there was a running version of the eventual Interleaf product that was based on Ecrit plus business charting."[6,e,f] Eight or so months after Pelletier joined the company (by this time appointed to lead engineering), he brought in several additional strong programmers (including the first author of this paper).

The original product concept was for a powerful graphics arts system running on an Interleaf-developed workstation. The intended market was technical publication shops, which were manually cutting and pasting text and graphics together into documents and getting proofs from a print shop. Just as Pelletier was arriving at Interleaf, the decision was made instead to use OEM workstations such as Sun and Apollo for the turnkey system the company intended to sell.[7] Boucher states that Barrett telephoned Andy Bechtolsheim at Sun, and that resulted in their having perhaps the first Sun-1 on the east coast.

According to a Seybold Report,[8] Boucher initially turned down an invitation late in 1983 to participate in Seybold Seminars 1984 (in March) because "Interleaf did not sell a publishing system" in Seybold's sense of the words. However, he still "tested the waters" by bringing a demonstration system to a Seybold Seminar, where the system was enthusiastically received by potential customers from the publishing systems marketplace. It was a WYSIWYG system (with its graphical user interface designed and implemented by Interleaf including use of a 3-button mouse), and potential users liked its functionality and especially its speed and that it ran on off-the-shelf workstations. The company shipped its first product in May 1984. From then on the company sold computer-based publishing systems.[8,g] The company went into the publishing systems market that was on the verge of developing, and Interleaf was perhaps the earliest significant player. The same 1986 Seybold report said, "Thus far, Interleaf has been the biggest winner in the 'tech-doc' revolution. It has sold fast, easy-to-use, and cost-effective systems to a lot of first-time users, most of whom had not typeset documents in the past."[h]

Starting out with a turnkey hardware-software system mindset, Interleaf primarily used traditional direct sales methods and quickly built marketing, sales, and support organizations. By 1984, the company had approximately 100 employees.

They counted on the margin on the hardware they OEM'ed as well as direct sale of their own software to generate revenue. They also began to make deals to have the Interleaf system run on workstations of other companies (e.g., DEC, Apollo), which had their own sales forces to sell the Interleaf system along with the company's hardware.[i]

Regarding the competitive situation in the early years, Harry George has emphasized[9] that the "opportunity that Interleaf was pursuing was enabled by the Canon laser printers coming on the scene and computers becoming fast enough to deal with placing a (few) million pixels for each document page versus dealing with pages of text by the character." "Companies such as NBI, CPI, Wang, and Harris-Lanier were the leaders in the whole character-based word processing world." George and Boucher figured they would "develop the pixel-based software needed to drive the new laser printers and most likely one of the [above] companies would buy [them]. That did not happen. Instead each of [those] business failed or got out of the word processing business." Half a dozen competitors did see the opportunity, such as Texet and Xyvision in the Boston area, "but Interleaf beat them out."[j]

An obvious question is, where did funding come from in those early years when Interleaf was not yet a public company? Of course, there was the profit on sales (both product sales and being paid to port the Interleaf software to vendor workstations), although this was not enough in total to reach profitability until 1988. Initially, the founders put in a little of their own money; next the Massachusetts Technology Development Corporation (a state agency trying to grow business in the state) invested some money; then angel investors put in some hundreds of thousands of dollars; then three venture capital companies invested some millions of dollars; then the same VCs plus Kodak invested tens of millions of dollars; finally most previous investors plus others invested some millions of dollars.[10] There were bank lines of credit both before and after Interleaf's initial public offering, but the company tried to minimize use of them. After the IPO, there were three limited partnerships to develop technology (for \$3.5M in 1987, \$3M in 1988, and \$2.250M in 1989).[11] (For more details about the pre-IPO funding of the company, see the Webnotes.[k])

GREAT SUCCESS, 1984–1988/9

With prior typesetting systems (e.g., Atex, TeX), the user specified where to place on a page each bit of text and graphics. WYSIWYG systems need to make assumptions about how users want to display things and implement higher level commands for enabling those displays. Interleaf was selling turnkey systems that included expensive hardware. Customers tended to be big companies and government organizations, and such entities have lots of different needs for how things are displayed. Thus, Interleaf had to implement lots of different capabilities. Such users also often have large or very large documents—hundreds or thousands of pages. WYSIWYG systems also need to instantly change what the user sees in a computer window when the user types another character or drags something somewhere on the page. The underlying system cannot recompose the entire long document from the beginning—that would be too slow. Thus, the system must be implemented in a sophisticated way so that what the user is seeing is changed immediately and any consequent changes throughout the document cause no detectable delay for the user. All of this—plus adding basic capabilities not in the original system and keeping up with competitor offerings—led to the necessity for a series of software releases.

OPS-2000 and TPS Releases 2.5, 3, and 4

OPS-2000 was the name of the first production system (internally it was release 2.0 with a release note dated 2 November 1983).[1] The sales brochure[12] for that system emphasized: the benefits of using the computer instead of prior manual and slow turn-around methods; creating graphics within the system as well as inputting text; availability of multiple fonts; enabling the same stored document to be displayed in different ways according to the "property sheet" used; accepting ASCII input from other computers of a variety of types (word processors to minicomputer to mainframes); ease of use; and a high resolution display [for the time] and output to a high resolution laser printer. The first system delivered ran on a Sun workstation with a Motorola 68010 processor[12] and one megabyte of memory, a 17-inch display, a 42 MB disk, and included a Canon LBP-10 laser printer (240 dpi). It cost $52 000 (the equivalent of about $130 000 today).[13] The system was quite revolutionary for its time.[m,n]

From this first release on, "filters" were an important aspect of the system. Filters were mechanisms for converting documents created in other systems, particularly word processors and later PCs, into Interleaf documents by translating codes for document properties, such as font and layout markup, into Interleaf-specific codes.

The next several releases after OPS-2000 had a new name—Technical Publishing System. TPS 2.5, 2.75, 3, and 4 are sketched in the Webnotes.[o]

The Market and Market Pressures

During this same period, competing products entered the market: Aldus PageMaker for the Mac (1985) going after small-office and individual customers, Ventura for the IBM PC (1986) with a goal of handling both short and long documents; FrameMaker for the Sun workstation (1986) directly competing with Interleaf, and QuarkXPress for the Mac (1987) but for high end users such as professional typesetters. Eventually, Microsoft Word also became popular, especially once Windows was replacing DOS. Adobe's PostScript had also come on the scene in 1984.

While Interleaf was trying to follow its competitors into the market of less expensive computers and less sophisticated publishing systems, it also was working hard at providing total publishing solutions to its traditional more high-end customers.

Over the series of releases mentioned above, the system was also made to work on more platforms with their various differences. For instance, by Release 4, the system could run on the IBM VM and MVS and DEC VAX/VMS mainframe systems as well as various workstations and was moving into the personal computer world with two new products—PCEditor and PCViewstation. These could run on off-the-shelf PCs; both could be part of a local network of TPS systems with the former a less expensive station but only handling text and the latter allowing a less expensive device for viewing documents anywhere in the organization.

The "corporate-wide solution" that Interleaf was offering and promoting allowed companies and government agencies to "employ electronic publishing at every level of the organization, at every phase of the document development cycles." Document style throughout the organization could be controlled from a central location. Example customers were Boeing, Ford, Monsanto, General Electric, Federal Express, the U.S. Department of State, and the Federal Reserve Board—big companies and institutions with needs for big and complex documents. "Books" is an example of a capability for handling the sometimes very large documents (perhaps hundreds of thousands of pages, maybe more) of large organizations such as those just mentioned. A "book" could contain any number of documents, which could be formatted consistently with the page number sequence running through all selected documents; there could be page number cross references such as "see page nnn of chapter NN"; all selected documents could be indexed together and have a common table of contents; and so on.[p]

Nondomestic (foreign) sales were important to Interleaf, providing one-third to two-fifths of company revenue, again primarily big dollar sales to big organizations.

Company Organization

As shown in a 1989 organization chart at tinyurl.com/interleaf-org, Interleaf had a fairly standard looking functional organization (and had had since its early years).

Founders David Boucher and Harry George were still the president and VP finance, and early employees George Potter and Steve Pelletier were VP marketing and chief technical officer. There were also VPs of the product management and planning (Larry Bohn), systems integration, and operations.

In 1989, Interleaf was selling turnkey systems (hardware and software) and had a large operations (manufacturing) organization, which put together the pieces of hardware for each customer order and had lots of hardware components (e.g., workstations and printers) in inventory. Within a year, much of that organization was significantly reduced in size as the company switched from selling turnkey system to selling only software (described in section "Software and Services, 1989–2000").

The company had been building a field sales and support organization as part of its marketing organization since the early days of the company. By the time of the first annual report in 1988, Interleaf already had 21 U.S. field offices, five offices in Canada (a joint venture with the Nexa venture capital firm of Ottawa), a London office, and offices about to open in Paris and Milan. At the time of the 1989 organization chart, more than half of the 750 employees of the company were in one of the boxes under Marketing.[q]

Over time the number of Canadian and U.S. offices varied a bit as did other international offices (or locations with partner companies) including Melbourne, Sydney, Brussels, Paris, Frankfurt, Hamburg, and Munich, Milan, Tokyo, Hong Kong, Madrid, Stockholm, Zurich and Montreux (Switzerland), and London. In some instances, the field activity was technically a subsidiary of Interleaf. The company sold both directly to customers and through other companies such as workstation manufacturers. The groups both at headquarters and in the field offices did customer support and maintenance for both the direct and indirect channels. Interleaf had the usual conflicts resulting from competing with its distributors.[r] (Also at headquarters were other marketing activities—some quite innovative.[s])

The organization under the Chief Technical Officer (called "engineering" below) included the developers of the base TPS system and many of its options. It also included quality assurance and documentation, and (at the time of this chart) the font group. The quality assurance activity was unusual in that each system developer had an assigned QA person, and overall there was 1-to-1 ratio of QA people to developers. (A hint at how this worked may be found in Tracy Kidder's book,[14] which describes the interaction between Paul English and his QA person Brenda White.) The engineering group was doing highly innovative work. Boucher has said, "the product was the most important thing, and particularly the engineers who created the product were given freedom to be creative." Some members of the Interleaf engineering organization have described it as the best place I ever worked or "an engineering playground." However, not all the development work happened in engineering. Some was done in the systems integration group, and some might have been done in the maintenance and support organization.

A later leader of the system integration group has said, "Systems Integration was oriented around integrating Interleaf's main products, across platforms, and with other products of other sources or manufacture. Most often these activities grew out of specific customer involvements. Each of these projects involved substantial customer-oriented work."[15]

The product management and planning organization (about 30 people in 1989) had been created to better direct how the company invested in research and development. (For examples, see the interview of Larry Bohn in this issue.)

It is interesting to note that departments throughout the company used the Interleaf system constantly for all documents, including annual reports and all of the extensive user documentation, which provided a huge base of "free" QA testing. People called this "eating your own dog food."

Product Development and TPS Implementation

Interleaf's product development activity was always a big source of the company's competitive capability.

The base TPS development group (product development on the 1989 organization chart) was only 12 people; 25 people were working on peripheral technology (5 in font design, 5 in printer systems, 10 in special projects, plus 5 more); 32 were doing workstation engineering (4 for the 386, 3 for Apollo, 3 for Ultrix, 4 for DEC VMS, 6 for Sun, 4 for IBM RT, 6 for release engineering, and two more); distributed publishing technology had 9 people (Lisp, batch, PC editor, mainframe, and Viewstation engineering); engineering operations had 20 people (7 in administrative services, the rest in network services); quality assurance had 48 people (8 for product development, 10 for peripheral technology, 13 for workstations, 8 for PC and distributed products, 5 in operations control, plus 4 more). One can see from this long list of people and functions that there were a lot of variations in adapting the base system to customer platforms and Interleaf options.

At the time of the 1989 chart, there were 114 people in what we are calling engineering (15% of company employees); nearly half of them were doing quality assurance. This is consistent with what Boucher told us saying he "always felt that the product was the most important thing," "the system had to work right all the time," and "the QA people were to get rid of the bugs; it was a complex system, and it was hard to test all the parts."[1]

The core product development group was small, it was an elite group of programmers, and its manager knew the strengths of its members when assigning projects.[t] The Release Engineering group (part of Workstation Engineering) played a big role in integrating the components together into an overall system; naturally, they also had a big role in rolling out new releases to customers.

By 1985, Interleaf had an "advanced Continuous Integration process in place"[16]—"doing nightly builds and nightly test"—"quite consciously" doing process innovation.[17] Steve Pelletier gives Joe Mahoney much credit for the company's process innovation that "made engineering much more manageable." Paraphrasing Mahoney,[18] the big innovations were released engineering with nightly builds for the various platforms on which the Interleaf system ran with their various user interfaces, and software testing with QA and product development tightly coupled (not the model of the tester working separately from the developer) and with the QA people having domain expertise. Pelletier also notes Mahoney's "innovative hiring practices," which "sought out liberal arts graduates with a demonstrated affinity for software, sort of a groupie psychology, but with both intellectual rigor and high empathy for software engineering personalities."[17]

The base TPS system had always been coded in the C programming language and also from the beginning used an object-oriented paradigm. Program editing was primarily done with Emacs, within a "home-grown integrated development environment (IDE) based on emacs Lisp. The IDE included version control and source merging tools which made it easy for some engineers to work across the entire codebase. There was a custom bugs database, and all source code changes were carefully documented in a templatized 'release mail' sent to the entire team with each code release."[19]

Because of its early entry into the electronic publishing market, Interleaf developed its own user interface. (This was before standard graphical user interfaces were developed for Windows, Mac, and Unix.) Interleaf's graphical interface design included two important menu design ideas that made it different from what the market became accustomed to later: *smart defaulting* and *stickiness*. An example of smart defaulting was that after a Copy operation, the default would be Paste, and a quick mouse click[20] would cause a paste without bringing up the menu, since that was very frequently the desired operation after a Copy. An example of stickiness would be that if you selected a graphic object and then rotated it 20°, if you selected another object and did a quick click, the same rotation would be performed.[u]

Additionally, Interleaf's graphical user interface featured windows that could be resized, moved around the screen, and overlap each other. Interleaf was an early (maybe the first) commercial product to use overlapping windows, a choice that quickly became standard in other window systems.[21]

TPS immediately became known for its speedy response to user commands, and this was an explicit goal of its implementors. Several clever techniques were used.[v]

As part of TPS 4 (released in 1987),[22,23] a new desktop command language was added to the system; it was written in Lisp. With it, users could modify base system operations, including creating sequences of operations.[w]

TPS 4 was a first step toward Interleaf 5, whose release is noted in section "Software and Services, 1989–2000," but we will say a bit about its implementation and capabilities here.[24,25] In prior releases, Interleaf used its own user interface implemented as part of the system. Starting with Interleaf 5, it depended as much as possible on the user interface of the underlying operating system and implemented (with the Lisp system added in TPS 4) the look of the TPS interface on top of the operating system interface. Interleaf 5 also moved on from the modularity of TPS 4. Interleaf 5 was a modular system programmable at every level, supported on many different hardware and software systems (more than a dozen platforms[26]) and further allowing users to extend the base systems pretty much however they desired. Structured and active documents were an important component of such flexibility.[27,28,x,y] Such extension capabilities were important for Interleaf's customers with complex document requirements.

Font and RIP Technology

The basic parts of the Interleaf system were a workstation running the Interleaf system and able to print to a relatively high resolution printer (e.g., 300-dpi laser printer).[8,29] With these, a user could mix text and graphics on the workstation screen and then print camera-ready copy on the printer. When Interleaf was founded, while it was developing its system leading to its first sale in 1984 and for years afterward, there was little standardization of font and printer technology.[30,z,aa] Various companies had their own systems of sending pages of text to a printer including in some instances their own raster image processor (RIP) technology to convert their internal font and page description technology into a bitmap that could be printed.

Although Interleaf had given up the idea of developing its own workstation, it carried on with Barrett's RIP Development Project.[31] This RIP was cross-licensed with printer manufacturer Dataproducts. Interleaf also preferred to use this RIP with printers such as the Canon CX. Using the Interleaf RIP provided better control of output typefaces and spacing.

Already by August 1983, Interleaf hired Les Snow from Compugraphic to work in the font area. In February of 1984, Kathy Nitchie, also from Compugraphic, was hired. Over time the font group varied in size, having up to half a dozen people.

For use in Interleaf's systems, the font group developed bitmap fonts in eight sizes for its Classic (somewhat like Times Roman) and its Modern (somewhat like Helvetica) typefaces, optimizing both for 75-dpi screen display and 300-dpi printer output. To keep character proportions equal in characters for screen display and printer output while maintaining legibility of the low resolution font, in small fonts the font people sometimes had to make a character have more pixel width than was usual.[ab]

Each typeface had roman, italic, and bold fonts. Product releases also included two typewriter fonts. a math font, an extended math font, a symbol font, and a Greek font. Occasionally the group was asked to add a custom character to a font. To get more typefaces to keep customers happier, Interleaf eventually licensed fonts from Bitstream and cleaned them up in the same ways—still bitmaps harmonized for screen display and printing. They did not move from bitmap to outline font technology until 1991–1992 (Interleaf 5) when they began to use Speedo technology fonts from Bitstream.

The detailed work of originally cleaning up, testing, and later maintaining the bitmaps for all the typefaces and sizes was a massive effort for the font group, even before they had to deal with other printer resolutions, adding special characters, supporting non-English languages, and their accented characters, and eventually also dealing with outline font technology. To try to deal with the effort, they developed Flexifonts, where the idea was generic sets of low resolution bitmap characters of different widths that could be used for screen displays where the width and point size of the high resolution printer font being used would determine the Flexifont parameters for screen display.[32]

In the long run, Interleaf moved into the world of PostScript printers and Adobe, Apple, and Microsoft font technology, as did other companies which had developed font and laser printer technology or were using other page description languages and font technology.[ac]

Sources of Revenue

Curves prod, serv, and othr in Figure 1 show Interleaf's sources of revenue over the period through 1988/9 described in this section. (Curves rev and inc are explained in a later section.)

From 1985 to 1988, the product sales (curve prod in Figure 1) went from \$4.8M to \$47.1M, while services revenue (curve serv) went from

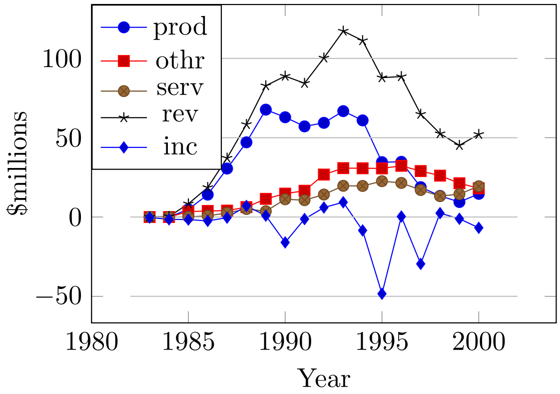


Figure 1. Financial summary.[33]

almost nothing to $5.0M, and software license and royalty revenue plus a little other revenue (curve othr in the figure) moved from $3.4M to $6.2M. (From 1988 on, curve othr is maintenance revenue.) The company became profitable, approximately $7M, in 1988; and thanks to its 1986 IPO (next section), it had plenty of working capital. Not too surprisingly, the company thought it was on a path to great success in what was essentially a software/hardware systems business. Interleaf was first into the market, their systems addressed a major problem for big corporations, and the system was highly popular. (This was despite the entry into the market of the desktop publishing companies and systems mentioned above and the existence of Adobe's PostScript.) With regard to PostScript, Interleaf's goal was for its page description language, Printerleaf, and WorldView viewer to become the industry standard.

A more thorough, two-page financial summary is at history.computer.org/annals/dtp/interleaf/interleaf-financials.pdf

Initial Public Offering

Revenue growth in the first five years, getting closer to profitability, and future prospects justified an initial public offering. It was the usual opportunity for investors, founders, and employees with stock options to cash out at least a bit and for the company to have cash for future growth. In particular, investor Kodak was able to sell its 7.4% stake in the company; having a share of Interleaf was increasingly awkward as the two companies found themselves competing with each other.[ad]

On May 16, 1986, the company did its SEC registration for an IPO, at a share price of $11.[ae] The registration statement noted a potentially large subcontract under an Electronic Data Systems printing services contract with the U.S. Government Printing Office that was being protested by competing vendors. On June 20, after a few hours of trading, Interleaf rescinded its IPO because the government had rescinded its contract to EDS.[34] This was a shock to employees ("there was despair"[1] although not apparently to the financial markets or customers). On June 26, the IPO actually happened at the reduced share price of $10.[35,36] Boucher credits this quick turnaround to co-founder and chief financial officer, Harry George: "Harry George was key to that coming out OK. With investment companies one needs a relationship of trust, and Harry was good at building relationships of trust. No one had ever heard of an IPO being rescinded before …But the repeat IPO was possible in a short time because of the relationships Harry had built."[1] (For Harry George's description of the IPO events, search for "the IPO, etc." in Webnote k.)

SOFTWARE AND SERVICES, 1989–2000

As shown in Figure 1 and in charts at history.computer.org/annals/dtp/interleaf/interleaf-financials.pdf, revenue growth had been excellent and the company was moving toward profitability, which it achieved in 1988 (approximately 10% of revenue). In addition to bringing out Release 4.0 (fiscal 1987), the PCEditor and PCViewstation were released in fiscal 1988; the former only handled text; both could be part of a local network of TPS systems providing lower cost terminals than the systems that could run full TPS. Other offerings were Interleaf Publisher for the Mac II and IBM Publisher for the IBM PS/2 and the IBM RT PC. Things were looking good.[af]

However, in 1989 the profit had disappeared despite a 50% revenue increase, and in 1990 the company lost $16 million (on revenue of almost $89 million, which was only 7% growth over the prior year).

These problems had several sources. First, the company was investing heavily in moving into the personal computer market in addition to remaining in the workstation market. Second, the company was caught with lots of aging inventory it had to sell at a discount (and to move to a more just-in-time system of operation). Third, in general platform prices were going down quickly and there was less margin on hardware sales to be had. Consequently, the company left the turnkey system business, “re-focusing Interleaf’s resources on high-margin software and services.”[37] This incurred restructuring costs of $13.5 million—severances for 135 people, inventory and fixed assets write-offs, and costs of consolidating facilities.[38] However, the company still needed a big direct sales effort because the desired customers still had sophisticated publishing environments. (By this time, international sales were 30% of revenue.)

The platform user interface technology was another problem. Interleaf had had its own graphical user interface for years, but the computer world was increasingly using Microsoft Windows, the Mac user interface, and Unix window systems—interface users were coming to expect and all of which were developed after Interleaf’s own innovative user interface (described as early as 1983 in the preliminary manual for Release 2.0[39]).[40]

For the rest of Interleaf’s years as an independent company, it followed rapidly changing market and competitive forces with new releases of its base system and other software packages (some developed internally and some acquired from external sources) to continue to sell software and services to its customers. The results are shown in the total revenue (rev) and net income (inc) curves of Figure 1.[ag]

Through these years, there were four changes of president of the company: David Boucher through FY1989, Robert Weiler in 1990, Mark Ruport 1991–1994, Ed Koepfler 1995–1996, and Jamie Ellertson from 1997 on. There were changes in organization. There was downsizing. There were changes in strategy—moving into document management in parallel with high-end electronic publishing and simultaneously trying to compete in consumer level electronic publishing, getting deeply involved with SGML, and playing catch-up in getting involved with the Internet.[ah] Central to the company’s development efforts, early in this period Interleaf 5 (a new name for what would have been TPS 5) was released with the system significantly opened up to allow it to fit into more application situations.

In some years, there were hopeful signs without big losses; however there was a loss of $48M in 1995 and a loss of $29M in 1997. More details are in the Webnotes.[ai]

Jamie Ellertson became the President of Interleaf in the last quarter of fiscal 1997 and announced that the company would be going back to core competencies of distributed publishing on more platforms, more standards, etc. The next year, 1998, there was a $2.4 million profit on an 18% reduction in revenue; and in 1999 there was a $1.1 million loss on a further 15% reduction in revenue. Over these two years, the company did lots of market research, met with many customers, and reduced its international sales force significantly while increasing the size of its domestic sales effort. The company would focus content management and on complex publishing (using proven Interleaf 6, which had been released in 1994, and the coming Interleaf 7).[aj]

Late in fiscal year 1999, Interleaf announced Interleaf 7, “a next-generation e-content publishing solution that allows users to author and publish complex documents to a large variety of Web and e-content formats, including HTML, XML, SGML or PDF.” Having perhaps not jumped on the Internet bandwagon when it originally had an opportunity, Interleaf in 1999 seems to have been positioning itself solidly in the Internet/Web world.

ACQUISITION BY BROADVISION

In early calendar 2000 (before the end of the fiscal year on March 31), Interleaf was acquired by BroadVision of Redwood City, CA, USA. Apparently, Interleaf had positioned itself sufficiently well to take financial advantage of the later stages of the dot com boom.[41] In the prospectus for the acquisition, Ellertson of Interleaf says, “At Interleaf, we have spent the past three years repositioning the company around new XML and wireless technology. This past year we have successfully penetrated the e-business marketplace by delivering state-of-the-art XML-based e-content management tools.” In the same document, Dr. Pehong Chen, President, CEO, and Chairman of the Board for BroadVision, said, “Through the acquisition of

Interleaf, BroadVision will be able to quickly expand our leadership in delivering personalized e-business applications across multitouch points such as web and wireless." Larry Bohn has suggested that BroadVision was a big company in the website development business, sort of stalled in its growth as Interleaf had been, but it had money; and Interleaf had lots of customers, and the customers needed websites in addition to Interleaf's capabilities. Whatever the reason, BroadVision paid $851.6 million to acquire Interleaf—certainly a good conclusion for its shareholders.

RETROSPECTIVE ASSESSMENT

Interleaf was an early entrant (if not the first) in what became the desktop publishing industry. Given the early state of the platform technology (workstations, not personal computers), RIP and printer technology (before the Apple LaserWriter), and page description and font technology (before PostScript, PDF, Type 1), innovation was essential. The early hardware technology was expensive, and thus customers tended to be large organizations with extensive and often unique publishing needs. In this early environment, Interleaf hoped its technology could become industry standards as the desktop publishing industry developed while at the same time it was in the business of selling systems adapted to a customer's environment and needs.

As the use of personal computers spread, competitors delivered products oriented to small users (individuals and small offices) of which there were many more than there were customers for Interleaf's system. It was hard initially to fit the Interleaf system into these smaller machines, and simultaneously the company came under great pressure to replace its original graphical user interface with the Mac and Microsoft Windows user interfaces that personal computer users were used to seeing. Interleaf was always playing catch-up in what became the mainstream of desktop publishing. However, their system business—the business that they really had always been in—the business where they supported so many different system configurations—remained viable for quite a few additional years.[ak]

Ultimately, most of the desktop publishing companies went away. Aldus and Frame Technology got themselves acquired by Adobe. Ventura was acquired by Xerox. Microsoft Word became the de facto standard for almost all low-end desktop publishing users and for much of the publishing industry. Interleaf lasted as an independent company longer than the others (not counting Microsoft).

In the end, BroadVision paid more for Interleaf than the acquiring companies paid for Aldus, for Frame Technology, or for Ventura. Interleaf had a base of systems customers that was complementary to BroadVision's existing business and was thus particularly valuable to BroadVision. In terms of longevity and the price of its acquisition, one might argue that Interleaf was the most successful of the desktop publishing companies.

AFTER INTERLEAF

BroadVision renamed the Interleaf system as QuickSilver and has continued to support and improve the system for many years.

Interleaf was a good training ground; and, even before the acquisition by BroadVision, people from Interleaf were taking influential or founding entrepreneurial positions in other companies. First stops after Interleaf for various people (or groups of people) were Intuit, Booklink/AOL, Viaweb, Kurzweil Educational Systems, Lotus, and Virtual Ubiquity. In engineering in particular, Interleaf had collected a significantly sized group of excellent-to-world-class computer programmers and strong technical managers and clusters of them moved on to other companies. For instance, nearly a dozen programmers from Interleaf were collected at Boston Light Software, which in turn was acquired by Intuit for its people rather than for its product.[42] Later many of those people plus other Interleaf people went to Kayak.[14] Various onetime technical people from Interleaf (and its founders and some other senior managers) went on after Interleaf to serial entrepreneur careers, and several became venture capitalists. Venture capitalists often mentor fledgling entrepreneurs, and one ex-Interleaf programmer co-founded Y-Combinator, which provides relatively small amounts of seed funding to a large number of startups.[am] Various middle and senior managers from Interleaf also went on to major management positions in other companies.

A number of artifacts of Interleaf's existence remain (beyond BroadVision's QuickSilver).

- Samples of Interleaf's website may be found at archive.org. At archive.org, put interleaf.com in the search window and click Go; this takes you to the archive's Calendar window for interleaf.com; there click on Site Map and there click on some of the earlier year numbers and there click on various of the colored circles or arcs.
- There was also an active comp.text.interleaf discussion group and there is a FAQ from that discussion group: www.faqs.org/faqs/interleaf-faq.
- For many years, Bob Treitman and Ian Poynter maintained the "interleft" mailing list, which is still active.
- Pulitzer Prize winning author Tracy Kidder had a good bit about Interleaf in his book *A Truck Full of Money*.[14]
- As part of this Interleaf history project, we are putting many documents we have found during our research in an online Interleaf archive at history.computer.org/annals/dtp/interleaf—for the benefit of future historians.

ACKNOWLEDGMENTS

The authors thank the many people who helped in various ways as they did the research and prepared the manuscript for this story. Especially helpful were D. Boucher, H. George, J. Barrett, B. Niamir, B. Morris, S. Pelletier, and K. Mundy—who knew the early days of Interleaf—and later employees R. Alter, T. Anderson, S. Baum, K. Berry, L. Bohn, P. English, C. Morris, J. Scaro, D. Weinberger, and K. Zola; F. Romano provided access to historical documentation; B. Beeton did proofreading. They thank the guest editors of this issue of the *Annals* for inviting this anecdote, reviewing it, and providing essential editorial feedback.

ENDNOTES/REFERENCES

1. D. Boucher, private communication, Oct. 21, 2019.
2. Interview of R. Morris, May 29, 2018, [Online]. Available: tug.org/interviews/morris.html
3. M. Hammer *et al.*, "Etude: An integrated document processing system," in *Proc. Office Autom. Conf. Dig.*, Houston, TX, USA, pp. 209–219, Mar. 1981. [Online]. Available: http://tinyurl.com/hammer-etude
4. J. Ehardt and P. Seybold, "MIT's Etude and Ecole systems," *Seybold Rep. Word Process.*, vol. 3, no. 9, pp. WP-6–WP-10, Oct. 1980.
5. B. Niamir, private communication, Oct. 17, 2019.
6. S. Pelletier, private communication, Oct. 3, 2019.
7. Once the proprietary hardware approach was abandoned, the hardware person, J. Barrett, continued with a more modest hardware project—developing an Interleaf laser printer with a proprietary raster image processor (RIP), the RIP was later licensed to Dataproducts.
8. B. Solimeno and J. Seybold, "Interleaf: A fast-moving tech-doc supplier," *Seybold Rep. Publishing Syst.*, vol. 15, no. 22, pp. 3–21, Jul. 1986. [Online]. Available: history.computer.org/annals/dtp/interleaf/1986-07-seybold.pdf
9. Harry George, private communication, Dec. 10–12, 2019.
10. *The Boston Globe*, Oct. 12, 1984.
11. Note L of the Interleaf 1989 annual report. [Online]. Available: history.computer.org/annals/dtp/interleaf
12. Circa 1984 OPS-2000 Interleaf sales brochure. [Online]. Available: tinyurl.com/interleaf-ops2000
13. F. Egan on Interleaf, perhaps at a meeting of people from Business Land, 1986, youtube.com/watch?v=ctpZ2a6vrxE
14. T. Kidder, *A Truck Full of Money*. New York, NY, USA: Random House, 2016.
15. R. Alter private communication, Oct. 9, 2019.
16. [Online]. Available: en.wikipedia.org/wiki/Continuous_integration
17. S. Pelletier, private communication, Nov. 30, 2019.
18. J. Mahoney, private communication, Dec. 2, 2019.
19. P. English, 2020-01-13 email which also notes, "Kimbo Mundy, one of the strongest engineers at the company, was the architect of all of these tools."
20. Interleaf's mouse was a three-button mouse. Left button marked a selection point; right button extended the selection; and middle button chose a command.
21. Robert Morris's 1985 overview description of the Interleaf system has much more information than we have presented here, both about the user interface and the overall architecture of the system. [Online]. Available: walden-family.com/interleaf/morris-see-enough-interleaf.pdf
22. Interleaf, *Seybold Rep. Publishing Syst.*, vol. 12, no. 15, Apr. 1988, pp. 12–14. [Online]. Available: history.computer.org/annals/dtp/interleaf/1988-15-seybold.pdf

23. [Online]. Available: history.computer.org/annals/dtp/interleaf/1989-interleaf-sales-brochure.pdf
24. M. Walter, "Interleaf 5: A complete overhaul of TPS," *Seybold Rep. Publishing Syst.*, vol. 20, no. 5, pp. 3–20, Oct. 1990.
25. See the Dymalski and Weinberger books. [Online]. Available: tinyurl.com/interleaf-bitsavers
26. "Translation of interview about Interleaf 5 by *Super ASCII* magazine." [Online]. Available: tinyurl.com/interleaf-japan
27. P. English *et al.*, "An extensible, object-oriented system for active documents," in *Proc. Int. Conf. Electron. Publishing*, 1990, pp. 263–276, [Online]. Available: walden-family.com/interleaf/english-interleaf.pdf
28. P. English and R. Tenneti, "Interleaf active documents," *Electron. Publishing*, vol. 7, no. 2, pp. 75–87, Jun. 1994. [Online]. Available: cajun.cs.nott.ac.uk/compsci/epo/papers/volume7/issue2/ep114pe.pdf
29. K. (Nitchie) Zola, personal communications, Oct. 2019.
30. The "font wars" articles in this issue describe the evolution of the font industry.
31. J. Barrett and K. Reistroffer, "Designing a raster-image processor," *Byte Mag.*, vol. 12, no. 5, May 1987, pp. 171–180, [Online]. Available: archive.org/details/byte-magazine-1987-05
32. D. Mellinger and K. Nitchie, "Font matching with Flexifonts," in *Proc. ProText IV*, J. H. Miller, Ed. 1987, pp. 111–117. [Online]. Available: walden-family.com/interleaf/flexifonts-paper.pdf
33. The 2000 dollars in the graph are an extrapolation to a full year of the financial results of Interleaf through the three quarters of fiscal 2000 before acquisition by BroadVision in the fourth quarter of 2000.
34. *InfoWorld*, June 30, 1986, [Online]. Available: tinyurl.com/ipo-rescinded
35. [Online]. Available: history.computer.org/annals/dtp/interleaf/1986-06-26-Interleaf-IPO-Prospectus-final-smaller.pdf
36. [Online]. Available: tinyurl.com/interleaf-s1-changed
37. 1990 Interleaf annual report. [Online]. Available: history.computer.org/annals/dtp/interleaf
38. November 13, 1989, Press release from chairman and CEO David Boucher and president Robert Weiler. [Online]. Available: history.computer.org/annals/dtp/interleaf/1989-Interleaf-exits-turnkey-systems-Nov-13-1989.pdf
39. [Online]. Available: history.computer.org/annals/dtp/interleaf/1983-Interleaf_2.0_SignPosts_Preliminary.pdf
40. R. Morris, "The interleaf user interface," in *Protext III, Proc. 3rd Int. Conf. Text Process. Syst.*, J.H. Miller, Ed. Oct. 1986, pp. 20–29. [Online]. Available: walden-family.com/interleaf/morris-user-interface.pdf
41. The stock price varied from 2 7/8 to 88 in 2000.
42. K. Berry, email of Oct. 27, 2019.

Mark Dionne was with Interleaf from 1983 to 1996. At Interleaf, he worked on the development of various key projects including filters, document formatting, the equation editor, tables, the Kanji version of the system, the desktop publishing systems, and Cyberleaf. He has also worked at two Kurzweil companies, Partners in Health, and Kayak. Contact him at markdionne@gmail.com

David Walden was a computer programmer, technical manager, and general manager at Bolt Beranek and Newman Inc. Since retiring from business, he studies and writes about computing history and digital typography. Contact him at dave@walden-family.com.


HOST
2020
4–7 May 2020 · San Jose, CA
REGISTER NOW!

CALL FOR PAPERS

IEEE 21st International Conference on Information Reuse and Integration for Data Science (IEEE IRI 2020)

Las Vegas, NV, USA / 11 – 13 August 2020

The IEEE IRI conference is a recognized forum for researchers and practitioners from academia, industry, and government to present and exchange ideas that address real-world problems with real-world solutions.

This conference is now seeking excellent, novel, and contemporary papers covering all aspects of Data—including Scientific Theory and Technology-Based Applications.

The conference includes, but is not limited to, the areas listed below:

- Application—Autonomous Vehicles, Business, Education, Engineering, Healthcare, the Internet of Things, Math, Military, Multimedia, NLP, Robotics, Science, Security, Social Networking, Space, Vision, et al.
- Contemporary as well as Novel Data Mining Techniques
- Data & Knowledge Representation and Management
- Data Science & Technologies—Heuristic Acquisition
- Data Visualization
- Graph Models
- Machine Learning & AI
- Predictive Data Analysis & Intelligence
- Predictive Modeling
- Recommender Systems
- Statistical Analysis
- Theory

This year, for the first time, IRI will hold classified military sessions at the nearby Naval Information Warfare Center (NIWC) in San Diego. Presentation of classified IRI military papers and attendance at these sessions is open to those holding an active US government Secret clearance. A clearance is not required for any other IRI sessions. SECRET presentations pertaining to the following areas of interest are encouraged:

- Artificial Intelligence, Heuristics, and Explanation-Based Learning
- Machine Learning
- Computer Vision
- Hypersonic Flight
- Autonomous Vehicles
- Predictive Maintenance
- Logistics
- Silicon Compilers
- Quantum Theory and Application

www.bit.ly/iri20-cfp

Important Deadlines

Abstract submission: 15 April 2020

Full paper research/industry/application/gov't track deadline: 22 April 2020

Short paper track deadline: 22 April 2020

Poster and demo paper track deadline: 8 May 2020

Full/short paper acceptance notification: 1 June 2020

Poster/demo paper acceptance notification: 15 June 2020

Camera ready submission deadline: 20 June 2020

Author registration due: 1 July 2020

IEEE COMPUTER SOCIETY

IEEE

IEEE COMPUTER SOCIETY

Education Awards Nominations

Call for Award Nominations
Deadline: 1 October 2020

Taylor L. Booth Education Award

A bronze medal and $5,000 honorarium are awarded for an outstanding record in computer science and engineering education. The individual must meet two or more of the following criteria in the computer science and engineering field:

- Achieving recognition as a teacher of renown.
- Writing an influential text.
- Leading, inspiring, or providing significant education content during the creation of a curriculum in the field.
- Inspiring others to a career in computer science and engineering education.

Two endorsements are required for an award nomination.

Read more information at **bit.ly/taylor-booth**

Susan H. Rodger
Duke University
2019 Award Recipient

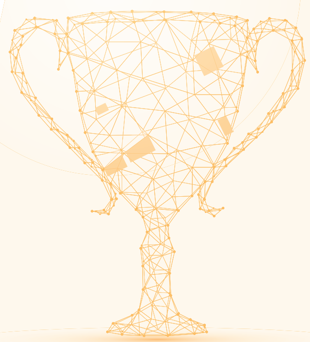
Mary Kenneth Keller Computer Science and Engineering Undergraduate Teaching Award

A certificate and $2,000 honorarium are awarded to recognize outstanding contributions to undergraduate education through both teaching and service and for helping to maintain interest, increasing the visibility of the society, and making a statement about the importance of undergraduate education.

The award nomination requires a minimum of three endorsements.

Read more award details at **bit.ly/cs-eu**

Robert R. Kessler
The University of Utah
2019 Award Recipient



Nomination Deadline

Submit your nomination by **1 October 2020**

Contact us at **awards@computer.org**

IEEE