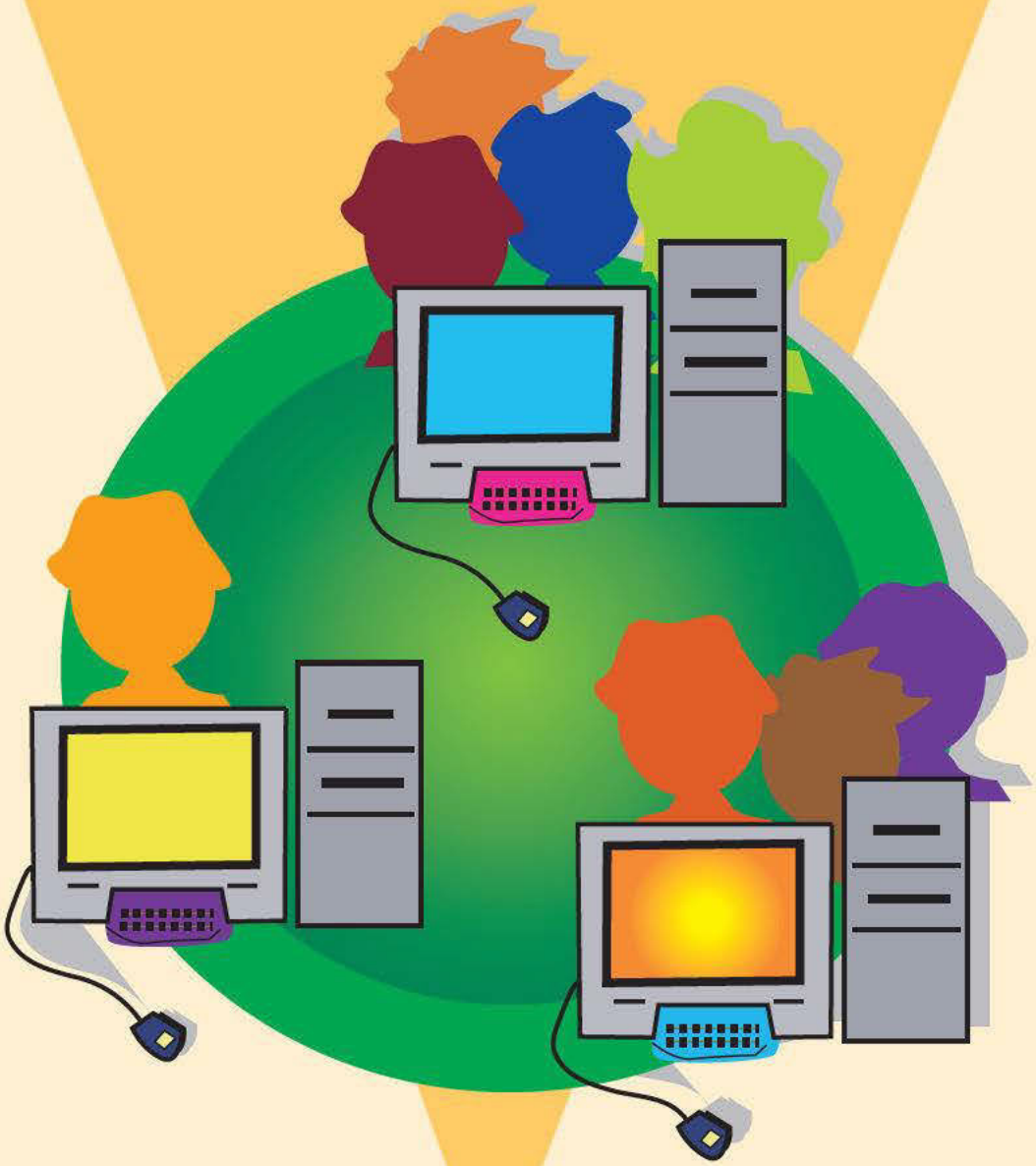
A Brief History of Human-Computer Interaction Technology

Brad A. Myers



Research in human–computer interaction (HCI) has been spectacularly successful and has fundamentally changed computing. One example is the ubiquitous graphical interface used by Microsoft Windows 95, which is based on the Macintosh, which is based on work at Xerox PARC, which in turn is based on early research at the Stanford Research Laboratory (now SRI) and at Massachusetts Institute of Technology.

Another example is that virtually all software written today employs user interface toolkits and interface builders—concepts that were developed first at universities. Even the remarkable growth of the World Wide Web is a direct result of HCI research: applying hypertext technology to browsers allows one to traverse a link across the world with a click of the mouse. More than anything else, improvements to interfaces have triggered this explosive growth. Furthermore, the research that will lead to the user interfaces for the computers of tomorrow is being carried out today at universities and a few corporate research labs.

This paper attempts to briefly summarize many of the important research developments in HCI technology, emphasizing the role of university research, which may not be widely recognized. By "research," I mean exploratory work at universities and government and corporate research labs (such as Xerox PARC) that is not directly related to products. By "HCI technology," I am referring to the computer side of HCI. A companion article on the history of the "human side," discussing the contributions from psychology, design, human factors, and ergonomics would also be appropriate.

Figure 1 shows the time span for some of the technologies discussed in this article. including when they were introduced. Of course, a deeper analysis would reveal significant interaction among the university, corporate research, and commercial activity lines. It is important to appreciate that years of research are involved in creating and making these technologies ready for widespread use. The same will be true for the HCI technologies currently being developed that will provide the interfaces of tomorrow.

Clearly it is impossible to list every system and source in a paper of this scope, but I have tried to represent the earliest and most influential systems. Further information can be found in other surveys of HCI topics (see, for example, [1, 11, 36, 41]). Another useful resource is the video *All The Widgets*, which shows the historical progression of a number of user interface ideas [27].

The technologies discussed in this paper include fundamental interaction styles such as direct manipulation, the mouse pointing device, and windows; several important kinds of application areas, such as drawing, text editing, and spreadsheets; the technologies that will likely have the biggest impact on interfaces of the future, such as gesture recognition, multimedia, and three-dimensionality; and the technologies used to *create* interfaces using the other technologies, such as user interface management systems, toolkits, and interface builders.

Brad A. Myers
Human–Computer Interaction Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891

bam@cs.cmu.edu

Basic Interactions

Direct Manipulation of Graphical Objects

The now ubiquitous direct manipulation interface, where visible objects on the screen are directly manipulated with a pointing device, was first demonstrated by Ivan Sutherland in Sketchpad [47], the thesis of his doctoral dissertation in 1963. Sketchpad supported the manipulation of objects using a light pen, including grabbing objects, moving them, changing size, and using constraints. It contained the seeds of myriad important interface ideas. The system was built at Lincoln Labs with support from the U.S. Air Force and the National Science Foundation (NSF).

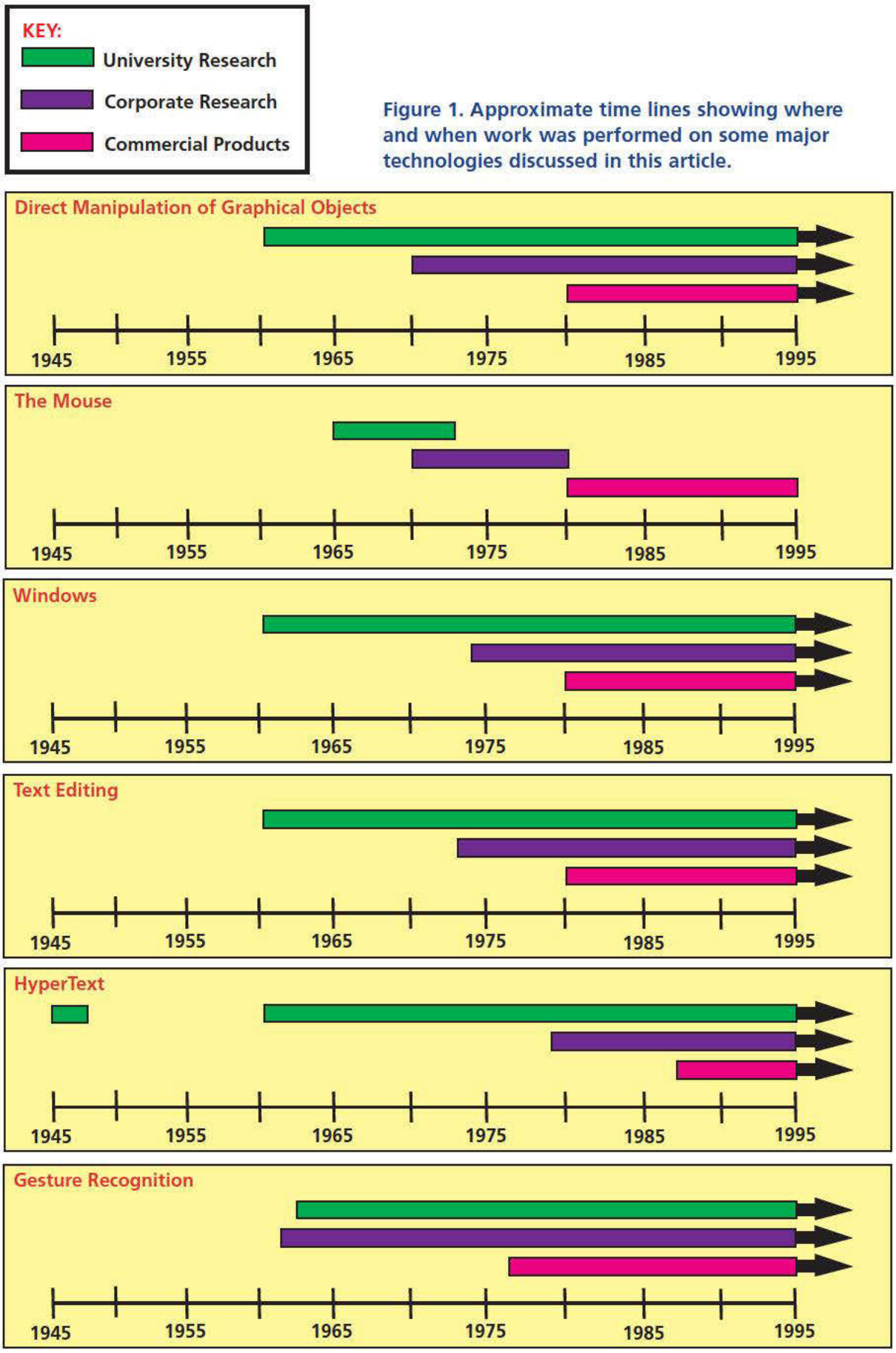


Figure 1. Approximate time lines showing where and when work was performed on some major technologies discussed in this article.

William Newman's Reaction Handler [33], created at Imperial College, London during 1966 and 1967, provided direct manipulation of graphics and introduced Light Handles [32], a form of graphical potentiometer that was probably the first "widget." Another early system was AMBIT/G (implemented at the Massachusetts Institute of Technology's (MIT) Lincoln Labs in 1968 and funded by the Advanced Research Projects Agency (ARPA). It employed, among other interface techniques, iconic representations, gesture recognition, dynamic menus with items selected using a pointing device, selection of icons by pointing, and moded and mode-free styles of interaction.

David Canfield Smith coined the term "icons" in his 1975 doctoral thesis on Pygmalion [44] (funded by ARPA and National Institute of Mental Health – NIMH). Smith later popularized icons as one of the chief designers of the Xerox Star [45]. Many of the interaction techniques popular in direct manipulation interfaces, such as how objects and text are selected, opened, and manipulated, were researched at Xerox PARC in the 1970s. In particular, the idea of "WYSIWYG" (what you see is what you get) originated there with systems such as the Bravo text editor and the Draw drawing program [11]

The concept of direct manipulation interfaces for everyone was envisioned by Alan Kay of Xerox PARC in a 1977 article about the Dynabook [18]. The first commercial systems to use direct manipulation extensively were the Xerox Star (1981) [45], the Apple Lisa (1982) [54], and the Macintosh (1984) [55]. Ben Shneiderman at the University of Maryland coined the term "direct manipulation" in 1982, identified the components, and gave psychological motivations for direct manipulation [43].

"THE SKETCHPAD SYSTEM MAKES IT POSSIBLE FOR A MAN AND A COMPUTER TO CONVERSE RAPIDLY THROUGH THE MEDIUM OF LINE DRAWINGS."

— IVAN SUTHERLAND [47, P. 329]

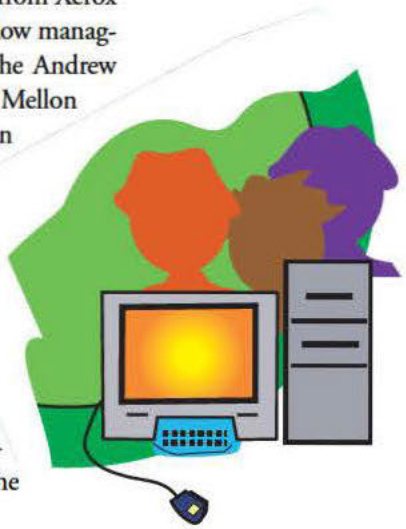
Mouse

The mouse was developed at Stanford Research Laboratory in 1965 as part of the NLS project (funding from ARPA, NASA, and Rome ADC) [10]. It was intended to be a cheap replacement for light pens, which had been used at least since 1954 [11, p. 68]. Many of the current uses of the mouse were demonstrated by Doug Engelbart as part of NLS in a movie created in 1968 [9]. The mouse was then made famous as a practical input device by Xerox PARC in the 1970s. It first appeared commercially as part of the Xerox Star (1981), the Three Rivers Computer Company's PERQ (1981) [25], the Apple Lisa (1982), and the Apple Macintosh (1984).

Windows

Multiple tiled windows were demonstrated in Engelbart's NLS in 1968 [9]. Early research at Stanford on systems like COPILOT (1974) [49] and at MIT with the EMACS text editor (1974) [46] also demonstrated tiled windows. Alan Kay proposed the idea of overlapping windows in his 1969 doctoral thesis [17], and overlapping windows first appeared in 1974 in his Smalltalk system [12] at Xerox PARC, and soon afterward in the InterLisp system [50].

Some of the first commercial uses of windows were on Lisp Machines Inc. (LMI) and Symbolics Lisp Machines (1979), which grew out of MIT Artificial Intelligence (AI) Lab projects. The Cedar Window Manager from Xerox PARC was the first major tiled window manager (1981) [48], followed soon by the Andrew window manager [35] by Carnegie Mellon University's (CMU) Information Technology Center (1983, funded by IBM). The main commercial systems popularizing windows were the Xerox Star (1981), the Apple Lisa (1982), and most important, the Apple Macintosh (1984). The early versions of the Star and Microsoft Windows were tiled, but eventually they supported overlapping windows like the



Lisa and Macintosh. The X Window System, a current international standard, was developed at MIT in 1984 [42]. For a survey of window managers, see [26].

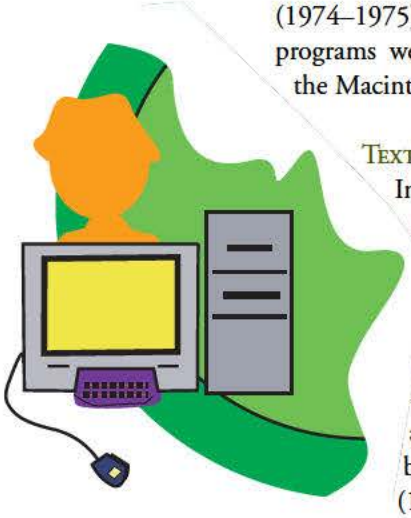
Applications

Drawing Programs

Much of the current technology for drawing programs was demonstrated in Sutherland's 1963 Sketchpad system [47]. The use of a mouse to manipulate graphics was demonstrated in NLS (1965). In 1968 Ken Pulfer and Grant Bechthold at the National Research Council of Canada built a mouse out of wood patterned after Engelbart's mouse and used it with a key-frame animation system to draw all the frames of a movie. A subsequent movie in 1971, *Hunger,* won a number of awards and was drawn using a tablet instead of the mouse (funding by the National Film Board of Canada) [3]. William Newman's Markup (1975) was the first drawing program for Xerox PARC's Alto, followed shortly by Patrick Baudelaire's Draw, which added handling of lines and curves [11, p. 326]. The first computer painting program was probably Dick Shoup's Superpaint at PARC (1974–1975). The first widely used drawing programs were MacPaint and MacDraw on the Macintosh (1984).

"THE DIFFICULTY OF MANKIND'S PROBLEMS WAS INCREASING AT A GREATER RATE THAN OUR ABILITY TO COPE. (WE ARE IN TROUBLE.) [SO] I ... COMMITTED MY CAREER TO 'AUGMENTING THE HUMAN INTELLECT.'"

— DOUG ENGELBART, ON THE NLS SYSTEM [8, P. 189]



Text Editing

In 1962, at the Stanford Research Laboratory, Engelbart proposed, and later implemented, a word processor with automatic word wrap, search and replace, user-definable macros, scrolling text, and commands to move, copy, and delete characters, words, or blocks of text. Stanford's TVEdit (1965) was one of the first CRT-based display editors that was widely used [51]. The Hypertext Editing System [53, p. 108] from Brown University had screen editing and formatting of arbitrary-sized strings with a light pen in 1967 (funding from IBM). NLS demonstrated mouse-based editing in 1968 (Figure 4). TECO from MIT was an early screen editor (1967), and EMACS [46] was developed from it in 1974. Xerox PARC's Bravo [11, p. 284] was the first WYSIWYG editor-formatter (1974). It was designed by Butler Lampson and Charles Simonyi, who had started working on these concepts about 1970 while at Berkeley. The first commercial WYSIWYG editors were the Star, LisaWrite, and, later, MacWrite. For surveys of text editors, see [24] and [53].

Spreadsheets

The initial spreadsheet was VisiCalc which was developed by Frankston and Bricklin (1977–1978) for the Apple II while they were students at MIT and the Harvard Business School, respectively. The solver was based on a dependency-directed backtracking algorithm by Sussman and Stallman at the MIT AI Lab.

HyperText

The idea for hypertext (by which documents are linked to related documents) is credited to Vannevar Bush's famous MEMEX idea from 1945 [4]. Ted Nelson coined the term "hypertext" in 1965 [31]. Engelbart's NLS system [9] at the Stanford Research Laboratories in 1965 made extensive use of linking (funding from ARPA, NASA, and Rome ADC). The "NLS Journal" [11, p. 212] was one of the first on-line journals and included full linking of articles (1970). The Hypertext Editing System, jointly designed by Andy van Dam, Ted Nelson, and two students at Brown University (funding from IBM) was distributed extensively [52].

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.