Volume 16
Issue 1 *Sacred Heart University Review, Volume XVI, Numbers 1 & 2, Fall 1995/ Spring 1996*

Article 4

February 2010

User Friendly: A Short History of the Graphical User Interface

Susan B. Barnes

SUSAN B. BARNES

User Friendly: A Short History of the Graphical User Interface

In the history of computing, 1995 will be remembered as the year that Windows 95 was introduced. Launched on August 24, 1995, Windows 95 is a new and improved version of Microsoft's popular Windows graphical user interface software.[1] *Advertising Age* reported that Microsoft spent $200 million on a mass-market global advertising campaign ``running in more than 20 countries in more than one dozen languages."[2] But behind the current Windows 95 hype and hoopla is thirty years of historical development. My essay briefly traces the development of the graphical user interface in the United States between 1970 and 1993.[3] The objective is to examine the decision-making process of interface developers and distributors.

Raymond Williams' theory of intentional technological development provides a useful theoretical grounding for my study, although, as I will indicate later, it may need certain modifications.[4] According to Williams, the development of technology cannot be separated from society. The purposes and practices of developing a new technology are known social needs, to which the technology is not marginal but central. Simply stated, the inventors of a technology know how they want the technology to be used before they invent it. Thus, technology is intentionally developed with the social purposes already defined.

In contrast to Williams, the conclusions of my study suggest that the graphical user interface developed through four distinct stages, not just one intentional stage. These stages can be identified as ideals-driven, play-driven, product-driven, and market-driven. During the development process the goals of the inventors and the

---

Susan B. Barnes is an Assistant Professor in the Department of Communication and Media Studies at Fordham University. This paper was presented at Sacred Heart University on November 5, 1995 as part of a symposium on The Implications of New Media Technology sponsored by the Media Studies Department.

practical use of graphical interface technology altered several times. As

Sacred Heart University Review, Vol. 16 [1996], Iss. 1, Art. 4

a result, two social paradoxes are now emerging.

The ideals-driven stage began with the pioneering work of Douglas Engelbart. Engelbart developed window-style display screens, the mouse, and interactive concepts that today are major features of computer interfaces. Engelbart was a visionary. His decision to create interactive computing was driven by neither recognition nor financial gain. Instead, he was influenced by the catastrophic events of World War II and the emerging social commentary of the wartime and post-war period.

Specifically, Engelbart was moved by Vannevar Bush's 1945 article, `` As We May Think,'' which encouraged scientists to create instruments of peace after the war.[5] Bush described the creation of a new organizational information system to enable people to make informed decisions. Engelbart devoted his life to developing the technology described by Bush. He called his interactive computer system the ``augmentation system,'' a system that he hoped would give society a new tool to facilitate complex decision-making in the post-war era.

In order to fund his augmentation project, Engelbart wrote and published a paper called ``A Conceptual Framework for the Augmentation of Man's Intellect.''[6] As a result, he received money from Bob Taylor and J.C.R. Licklider through the Advanced Research Projects Agency (ARPA). ARPA's funding was allocated as a response by the United States Government to the Russian launching of Sputnik. In 1963, the ``space race'' was on, and funding was available for any project that could potentially place the United States in a technologically superior position to the Russians. As head of ARPA, Licklider's personal mission was to develop the concept of interactive computing. In 1960, while still an M.I.T. psychology professor, Licklider wrote a paper called ``Man-Computer Symbiosis.'' He proposed that people should be able to think interactively with a computer. Licklider and Engelbart shared a common vision to create better decision-making tools for a peace-time society.

By 1968, Engelbart had a working prototype and he demonstrated his vision of interactive computing at the Fall Joint Computer Conference. This turned out to be a seminal event in the history of computing because it inspired the next generation of computer developers. Engelbart and his team showed window display screens,

the mouse, hypertext, and multimedia applications.
But major funding for Engelbart's vision came to a sudden halt with the Vietnam War. Controversy over American involvement combined with the passing of the Mansfield Amendment in 1970 cut off money for non-military computer projects. Thus, the first ideals-driven stage of graphical interface development came to an abrupt end.

Simultaneously, Xerox was reaching the billion dollar mark in assets and they wanted to invest research money into developing ``the office of the future." Xerox hired ARPA manager, Bob Taylor, to recruit computer scientists for its newly created Xerox PARC (Palo Alto Research Center) facility. Taylor had supported Engelbart's concept of interactive computing, but his point of view was different. He wanted to develop computer systems for individual office workers rather than for small-group decision-makers. As far as Taylor was concerned, computers needed to be developed for individual or ``personal" use. Individual systems could then be used to support the sharing of information through networks. In 1969, Taylor had established the ARPAnet, a network of information sharing by research sites that would later become the Internet.

Under Taylor, Xerox hired computer researchers away from ARPA and other defense-related projects. These computer scientists were already experienced in networking and interactive computer systems. Among those Taylor recruited, was Alan Kay, who was just finishing his Ph.D. in computer science at the University of Utah. Taylor set Kay up with a research team to assist him in the development of his Dynabook research project. The Dynabook was a small personal computer with a graphical interface. Kay's purpose for developing interface technology followed the goals of Bush and Engelbart. He wanted to invent computer-based tools to organize information and support the decision-making process. Beyond that, Kay's intention was to turn the computer into a medium of communication that was easy for children to operate and program. Therefore, he was interested in developing new types of visual programming languages and interfaces.

At Xerox PARC, Kay and his team experimented in a playful way developing object-oriented programming languages and graphical interfaces. The designs were then tested with children. The result was the creation of Smalltalk and the first graphical user interface. Kay

Sacred Heart University Review, Vol. 16 [1996], Iss. 1, Art. 4

describes his design model as ``Doing with Images Makes Symbols.'' This model was based on Jerome Bruner's concept of learning mentalities. According to Bruner, children learn to construct models of the world first in the enactive (action) mentality, and then in the iconic (visual) mentality before they begin to construct a world view in the symbolic language mentality. Kay applied Bruner's concept of learning mentalities to the development of graphical user interfaces to construct a model to teach children how to ``read'' and ``write'' on a computer. According to Kay, the slogan ``Doing with Images makes Symbols'' implies that one should start or be grounded in the concrete by working with images and then move into the more abstract symbol systems used by computers.

The following is Kay's model:

DOING	mouse	enactive	Know where you are, manipulate
with			
IMAGES	icons, windows	iconic	Recognize, compare, configure, concrete
makes			
SYMBOLS	Smalltalk	symbolic	Tie together long chains of reasoning, abstract[7]

Neither Smalltalk, an object-oriented programming language, nor Kay and his team's graphical user interface was developed as a commercial offerings. Instead, Xerox considered them to be intermediate steps in the long term strategy to develop the office of the future. Remarkably, the PARC researchers were given the freedom and opportunity to invent without the burden of turning their research into short-term products. However, Xerox's long-term research strategy did not consider the new emerging technology of the personal computer.

At the same time Xerox researchers were playing with traditional computer systems, not too far away, a group of hackers in California started tinkering with the first microcomputers. People interested in computers formed a club called the Homebrew Computer Club to exchange information freely. Group attendance grew so rapidly that their meetings soon filled an auditorium at Stanford University. Members of the Homebrew Club were some of the first hardware hackers to play with building microcomputers. Essentially, these enthusiastic computer hobbyists were building computers one chip at a time, starting with the microprocessor. Members of this club were also some of the first people to purchase and build the Altair, the first microcomputer kit. Both the invention of the microprocessor chip and introduction of the Altair led to a play-driven stage in the development of microcomputer or personal computer technology.

This play-stage of hardware hacking was also influenced by social ideals. Lee Felsenstein, an original member of the Homebrew Club, was simultaneously involved with a group of Berkeley hackers who were trying to take computers out of the control of corporate structures and put them into the hands of individuals. The development of the microcomputer was a technological step towards reaching this goal. As Theodore Roszak notes:

> From its beginning, the microcomputer was surrounded by an aura of vulgarity and radicalism that contrasted sharply with the mandarin pretension of the high tech mainstream. This is because much of the new, smaller-scaled technology was left to be developed outside the corporate citadel by brash, young hackers — especially in California, where the socially divergent types had gathered along that strip of the San Francisco peninsula which was coming to be called Silicon Valley. By the mid-1970s, small groups of these hackers had begun to meet in informal rap sessions where computer lore was freely swapped like gossip over the cracker barrel in a country store.[8]

These hackers were so intensely interested in playing with computers that the fact they would have to build one was no obstacle. As a result, the development of microcomputer technology began outside the corporate computer structure by individuals who were interested in playing with technology.

The Homebrew Computer Club meetings were a focal point for hackers to discuss and develop the emerging microcomputer. One club member, Steven Wozniak, started to build his own microcomputer after attending club meetings. Wozniak was something of a prodigy. His father was an electronics engineer and Wozniak had been building computing devices since he was in the eighth grade. He lived a freewheeling lifestyle that revolved around working for Hewlett-Packard, hacking on his own, and playing computer games. When Wozniak first build his computer, the Apple I, he had no intention of starting a business. He just wanted to go down to the Homebrew Club and show off and play with it. However, Steve Jobs persuaded him to sell his design as a product.

Wozniak's first microcomputer led to the development of the Apple II and the creation of Apple Computer, Inc. The Apple II became a huge commercial success, making these two college dropouts multi-millionaires. According to Steven Levy, the Apple II moved the personal computer out of the play-driven stage of hardware hacking into a product-driven stage of development.[9] Graphical user interface technology was soon to follow.

Apple Computer's phenomenal success in marketing and selling personal computers is legendary. In 1979, Jobs took a tour of Xerox PARC and saw an immediate commercial application for Kay's interface technology. He realized that putting a graphical screen on Apple's personal computers would make them easier to operate. He applied Kay's graphical features to the Macintosh computer and created the ``Desktop Finder'' interface software. However, Jobs appropriated only Kay's visual screen design, not the object-oriented Smalltalk programming language.

The Macintosh, as conceived by Jef Raskin, its original designer, was to be a home ``appliance,'' not a programmable computer. As a result, Kay's programming language was not deemed necessary and was abandoned. Raskin's concept was to create an easy-to-use computer with preprogrammed software applications. By adding a graphical screen to this easy-to-use computer, it became ``user-friendly.'' This term would later become the basis for a wildly successful marketing strategy to sell the Macintosh.

The Macintosh became the bridge into the fourth stage, the market-driven stage. Bill Gates took Macintosh's Desktop Finder interface and, with minor modifications, marketed it as Microsoft Windows. To date, Microsoft Windows has been the single most powerful influence in the market-driven stage of interface development. Windows brought the Macintosh-style interface to MS-DOS personal computer users. It created a new marketplace by making the PC visually resemble the user-friendly Macintosh. Gates intentionally developed visual interfaces to make complex computer technology easier for non-technological people to operate and Windows quickly became a dominant influence in the PC market because of Gates's previous success in setting industry software standards with MS-DOS. As a result, the Windows graphical user interface is currently being used by millions of people, ranging from office workers to school children.

In four short years, Windows has become the leading graphical user interface in the personal computer market. But IBM and Apple are battling with Microsoft in the marketplace to keep Gates from controlling the future of graphical interface standards. Gates's talent for tapping new markets is remarkable. He now wants to expand Windows into the realm of interactive TV and digital devices. His goal is to make Windows the interface standard for these emerging technologies. ``Windows everywhere'' is his corporate battle cry in the interface wars. His intention is to make Windows the graphical user interface standard for all types of computer-based systems.

The emergence of Windows may be the embodiment of Williams' intention theory: that technology is developed with certain purposes and practices already in mind — or rather, in a *particular* mind. And in this case, the particular mind behind the intentional development of the ``user-friendly'' graphical interface is that of Bill Gates. Gates is an entrepreneur, a practical man: the Thomas Alva Edison who invents the light bulb and then General Electric to capitalize on it. He first controlled the MS-DOS operating system software and now he is controlling Windows, the graphical interface technology that runs on top of his operating system. Gates describes how he capitalizes on Windows as follows:

> In a free-market economy, businesses are not required to share their innovative work with competitors. But some industry observers say that, as creator of Windows, the world's most popular PC operating system, we should be required to do so. In fact, for good business reasons, we do so voluntarily.
>
> We make available information to allow software publishers to develop terrific applications for Microsoft operating systems because these applications fuel demand for our operating system products.[10]

In each of the four stages of technological development, the graphical user interface was intentionally developed. However, the intentions in each stage were different. In the first stage, the intentions were ideals-driven. Engelbart aspired to invent better decision-making tools for a post-war society. During the second, play-driven stage, the purpose behind developing the technology remained essentially the same, but the inventive process was fueled primarily by the sheer excitement and fun of tinkering with a new technology. In the third, product-driven stage, however, the intention of developing graphical user interface technology changed. Here the technology became part of a strategy to sell personal computers to computer-illiterate users. Now, in the final market-driven stage, Gates is developing graphical interface technology with the intent of controlling the graphical interface standard for all types of computer devices, and capturing the largest share of the market for such devices.

During the development process the goals of inventors and the practical use of graphical interface technology altered several times. Thus, my study does not entirely support Williams' perspective on how technologies develop because the intentions of the original inventors have not in fact guided the way in which the technology has come to be used. While Williams' idea is a useful one, it does not account for the shift in intention that occurred between the second and third development stages of the graphical user interface. The results of my study suggest that a pivotal moment in the history of graphical interfaces was Steve Jobs's decision to apply the visual screen elements to Apple computers without the underlying programming language. Jobs's intention was primarily to sell computers, and in the interest of that objective, he largely ignored the social and cognitive ideals underlying the earlier designs. Today, Jobs's decision can be viewed as a historical turning point that created paradoxical situations for the future development of graphical user interfaces. These paradoxes were created because Jobs considered the computer to be a tool rather than a medium of communication.

In the first two stages of development, Engelbart and Kay viewed the computer as a medium of communication. They were interested in developing computer literacy skills to enable people to learn how to read and write with a computer. In contrast, the Macintosh was developed as an appliance or tool. Jobs saw the interface as way to make a machine easier to operate. Neil Postman describes the difference between a medium and a machine as follows: ``a technology . . . is merely a machine.'' It ``becomes a medium as it employs a symbolic code, as it finds its place in a particular social setting.'' Thus, ``a medium is the social and intellectual environment a machine creates.''[11]

Engelbart's and Kay's models of computer interaction paid particular attention to the social and intellectual environments created by the computer. For example, Engelbart argued that computer technology must not be developed haphazardly. As a result, his research included methods for studying the effects of computer interaction on people in social settings. Kay was also aware of the influence computers would have on culture. His interface model followed a logical cognitive progression to develop computer literacy skills. His intention was to develop the computer as a medium of communication by making the learning process accessible to children. However, the social awareness of these early inventors was abandoned in the third stage of development.

Jobs did not view the computer as a medium of communication. He saw it as a machine. Consequently, he did not see the need for the underlying programming language in Kay's design. Jobs only saw the surface visual representation displayed on the computer screen. As a result, Apple used the visual icons as a marketing strategy to sell ``user-friendly'' computers. Kay refers to this as putting ``training wheels'' on a computer. The problem with the Macintosh model is that the training wheels can't be taken off. There is no path to learn programming.

Today, Apple's policy of ``isolating and insulating" its users from the internal operations of a computer creates two paradoxical situations. The first paradox relates to access to knowledge. While ``user-friendly" interface designs make computers accessible to inexperienced users, they also make the user dependent on the software programs from Apple and its developers. This creates what Postman refers to as a ``knowledge monopoly."[12] Only people with a high level of programming skills and access to Apple developer information can program the machine. Thus, while Apple brings computers to the masses, it also forces the masses to buy pre-programmed software packages that Apple controls. Currently, this trend is being implemented on a larger level by Microsoft with Windows.

In 1984, the original idea of ``user-friendly" software helped people to easily operate word processing, simple drawing, and spreadsheet software programs. By focusing on ease of operation as a strategy to increase sales to non-computer-literate markets, however, commercial interfaces hide the machine operations from the user. This creates a group of people who can operate the machine, but who are completely illiterate in terms of how it works or how to program it. Thus today, most computer users ``can do some jobs, without understanding why or how. There is a whole industry dedicated to making it possible to compute without knowing about computing."[13]

In today's market-driven stage of graphical interface development, ``entrepreneurs exploit consumer incompetence. Mac, Windows, OS2, NeXT Operating systems, the mouse and other devices that simplify the use of the computer are still arcane to most users"[14] As a result, the production of computer manuals, computer commentaries, and computer books for ``dummies" approaches Talmudic proportions: ``Publishers get rich publishing books that purport to make it easy to use complex programs. The net effect is that it becomes more difficult to train the people who will design the next generation of computers."[15] Thus, the ``user friendly" strategy sews the seeds of its own destruction. Here is the second paradox emerging from the ``user-friendly" approach: by not supporting the development of computer literacy skills, graphical interfaces fail to develop the programming skills required by the next generation to develop and maintain computer technology. Currently, a gap is being created between computer users and highly trained software engineers.

This social paradox is an unintentional consequence of stage three in the development of graphical interfaces, the stage in which the social intentions of the original developers were reversed. This reversal suggests that technology does not develop according to the intentions of its originators, but that the social and technical variables influencing technology development are so varied that it is impossible to determine how a technology will develop. Moreover, it is impossible to predict the social impact of a new technology from the intentions of its originators. Jobs's decision to add a graphical screen to the Macintosh was a twist of fate that changed the direction of interface development. As a result, the social consequences of this technology are now moving in the opposite direction from that intended by the original inventors.

Today, graphical interface technology is still in its beginning stages. Whether or not the current intentions of developers and distributors will shift again is a subject for future research.

Notes

[1]Graphical User Interface is a method that allows computer users to see and directly manipulate representations of objects on the computer screen, rather than addressing the objects through an intervening command language code. For example, Microsoft Windows allows users to point and click with a mouse on visual representations of documents to directly open the document without typing in a keyboard command.

[2]Jan Jaben, ``Eighth Wonder of the World: Microsoft Markets Worldwide, Opening Windows to Criticism," *Advertising Age International*, September 18, 1995, section I, p. 3.

[3]The present essay is based on the much fuller discussion of the topic in Susan B. Barnes, *The Development of Graphical User Interfaces from 1970 to 1993, and Some of its Social Implications in Offices, Schools, and the Graphic Arts*. Diss. New York University (1995).

[4]Raymond Williams, *Television Technology and Cultural Form* (New York: Schocken Books, 1974).

[5]Vannevar Bush, ``As We May Think," in *From Memex to Hypertext: Vannevar Bush and the Mind's New Machine*, ed. James M. Nyce and Paul Kahn (Boston: Academic Press, 1991), pp. 197-216.

[6]Douglas C. Engelbart, ``A Conceptual Framework for the Augmentation

Sacred Heart University Review, Vol. 16 [1996], Iss. 1, Art. 4

of Man's Intellect," in *Vistas in Information Handling*, ed. Paul W. Howerton and David C. Weeks (Washington, DC: Spartan Books, 1963), 1: 1-29.

[7]Alan C. Kay, ``User Interface: A Personal View," in *The Art of Human-Computer Interface Design*, ed. Brenda Laurel (Reading, MA: Addison-Wesley Publishing, 1990), pp. 191-207.

[8]Theodore Roszak, *The Cult of Information* (New York: Pantheon Books, 1986), p. 141.

[9]Steven Levy, *Hackers* (New York: Doubleday, 1984).

[10]Bill Gates, ``It's Feast, Not Famine, At the Software Table," *New York Times*, March 26, 1995, p. F13.

[11]Neil Postman, *Amusing Ourselves to Death* (New York: Penguin Books, 1985), p. 84.

[12]Neil Postman, *Technopoly: The Surrender of Culture to Technology* (New York: Alfred A. Knopf, 1992).

[13]Gerald M. Phillips, ``A Nightmare Scenario: Literacy & Technology," *Interpersonal Computing and Technology: An Electronic Journal for the 21st Century*, 2, No. 2 (April 1994), 51-73. (Internet location for this journal: LISTSERV@GUVM.GEORGETOWN.EDU)

[14]Phillips, p. 64.

[15]Phillips, p. 64.