HCI methods into the software life cycle processes of the time (see Mantei & Teorey, 1989). John Carroll (1997) attributes the fundamental idea of iterative cycles of prototyping and design to the industrial designer Henry Dreyfuss (1955). It was Fred Brooks (1975), referring to software programs, who said, "Plan to throw one away, you will anyhow" (p. 116).

Interface Builders

Once the interface had been disentangled from applications and the technology of GUIs began to mature, there was interest in simplifying the implementation of GUIs. In 1979–1985, Albert Stevens, Bruce Roberts, and their colleagues at BBN together with James Hollan at the Naval Personnel Research and Development Center (NPRDC) built a prototype graphics-based simulation system for training Navy crewmen to operate a steam propulsion plant (Stevens, Roberts, & Stead, 1983). The software demonstrated many of the ideas later incorporated into "Interface Builders," as they came to be called. It had a library of widgets corresponding to components in a steam plant (valves, turbines, pumps, gauges, switches, piping) that could be introduced into a diagram by simply dragging them to the proper location and dropping them in place, then adjusting their visual properties and connecting them to the underlying simulation through a menu interface. As the simulation ran, the widgets were automatically updated to reflect the changes in the simulation (gauges moved, pipes showed animated flow) and the user controlled the simulation by interacting with the widgets using a mouse (opening and closing valves). Brad Myers suggests that Stevens and Roberts' "Steamer" was possibly the first object-oriented graphics system (Myers, 1998). In 1981, Austin Henderson built Trillium (Henderson, 1986), which he referred to as a user interface design environment, building on and greatly extending the Steamer work. The real power of this approach became evident when the Macintosh was introduced. The Macintosh incorporated widgets directly on a computer chip, which allowed Apple to control their use by constraining software developers to use a well-defined and limited set. In addition, the Macintosh developer tools included a Resource Editor that further simplified the interface development task. Widgets-on-a-chip and the Resource Editor, together with carefully constructed guidelines for developers, were very successful in ensuring that each new application, particularly applications developed by a third party, would retain the look and feel intended by Apple's design staff. Users benefited because the Macintosh developer tools enforced a consistent look and feel across applications.

Donald A. Norman

Don Norman has identified with the field of HCI at least since 1981 when he authored an irreverent paper criticizing the UNIX interface (Norman, 1981a). Having first been a leader in the field of human information processing, then in cognitive science, at the University of California, San Diego, he brought to the applied world in general and the HCI world in particular a profound knowledge of the relevant cognitive psychology and behavioral science. The book he co-edited with Stephen Draper and contributed to in 1986 on "User-Centered System Design" focused specifically on HCI and set the tone for how behavioral science could impact HCI. In it, he used the term Cognitive Engineering, a term that, in the author's opinion, he coined (Norman, 1981b). He spoke of the Gulf of Execution and the Gulf of Evaluation, referring to the gulf between human goals and the physical interfaces through which humans must communicate with technology:

> The user of the systems starts off with goals expressed in psychological terms. The system, however, presents its current state in physical terms. Goals and system state differ significantly in form and content, creating the Gulfs that need to be bridged if the system can be used. The Gulfs can be bridged by starting in either direction. The designer can bridge the Gulfs by starting at the system side and moving closer to the person by constructing the input and output characteristics of the interface so as to make better matches to the psychological needs of the user. The user can bridge the Gulfs by creating plans, action sequences and interpretations that move the normal description of the goals and intentions closer to the description required by the physical system. (Norman & Draper, 1986, pp. 38–39)

Over the next several years, he became more committed to working on applied problems that could be informed by cognitive science. He worked on understanding typewriting skill. He undertook a deep analysis of human error and his landmark book, *The Psychology of Everyday Things*, later renamed *The Design of Everyday Things* (Norman, 1988), sensitized the public to the way people think and its implications for design. In 1993, he placed a foot solidly in the applied camp when he joined Apple Computer, Inc., and since then has been a consistent advocate for better HCI educational preparation for work in industry and better understanding of what cognitive science has to contribute to design.

Human–Computer Interaction as a Professional Field

It was during this period that specialists began to speak of a professional field of usability and HCI. The first textbook, (Shneiderman, 1980) was published, to be followed by many others, for example, Foley and Van Dam (1982) and Smith and Green (1980). In 1978, Ramsay and Attwood published a landmark bibliography abstracting the existing publications in the then nascent field (Ramsey & Atwood, 1979). Gary Perlman deserves credit for initiating a still more extensive bibliography project in 1988 that is now the primary reference source for HCI materials (Perlman, 2001).

Brian Shackel, quoting Brian Gaines, attributes the first HCI conference to the International Conference on Man–Machine Systems in 1969 at Cambridge, England, Shackel (1997), but that event was really a human systems conference at which HCI was highlighted. In 1981, the Association for Computing Machinery (ACM) Special Interest Group on Social and Behavioral Computing (SIGSOC) held a conference in Ann Arbor, MI, entitled, "Conference on Easier and More Productive Use of Computing." The first general conference devoted to HCI was held in Gaithersburg, MD, in 1982 and at that meeting it was announced that SICSOC would change its name to the Special

Interest Group on Computer–Human Interaction (ACM SIGCHI; http://www.acm.org/sigchi/). It was officially founded later that year (Borman, 1996). The series of CHI conferences officially began in 1983. The first international conference, IFIP Interact '84, was held in London. The annual SIGCHI Conference and the biannual INTERACT Conference remain the premiere HCI forms in the field.

The *International Journal of Man-Machine Systems*, founded in 1968, began publishing articles concerned with HCI in the 1970s and 1980s, and changed its name to *International Journal of Human-Computer Studies* in 1994. It was followed soon after by *Behaviour and Information Technology* in 1982, *Human-Computer Interaction* in 1985, *International Journal of Human-Computer Interaction*, and *Interacting with Computers* in 1989.

Usability Testing

Usability testing, as a part of the research on interactive technologies, was already going on in laboratories around the world, but it was during this period that the first application usability testing was initiated. IBM established formal usability labs. Considerable formative testing and evaluation went into the development of the Xerox Star Workstation (Bewley et al., 1983). In 1978, the Social Security Administration was undertaking a serious design exercise focused on the "Future Process," a major software development that was to replace Social Security hardware and software with a distributed interactive system that provided for source data entry and retrieval in Social Security Offices around the country. BBN set up a usability laboratory at Social Security headquarters in Baltimore in support of this project. In the course of the work, prototype interface designs were developed on Xerox Altos[4] and more than 60 Social Security Claims Representatives from around the country spent a week at headquarters participating as users in alternative design evaluations. A year later, management of Social Security changed and plans for the Future Process were abandoned. It took at least another 5 years before Social Security began the serious implementation of large-scale interactive systems in the field.

Naturalistic Observation and Ethnography

Converging forces derived from perceived limitations of cognitive task analysis based on interviews, and laboratory experimentation with alternative interface designs, led to new emphasis on field, naturalistic observation and contextual research as a means for better understanding the user, and the environment in which work is accomplished. John Whiteside led this movement, deriving his arguments from the philosophy of Heidegger (1962) (Whiteside, Bennett, & Holtzblatt, 1988). The argument was that truly useful applications depended on understanding in depth the social and environmental context in which the software would be used, and for that purpose there was no substitute for extensive in situ observation in the workplace. At about the same time, anthropologists and ethnographers whose specific expertise is in just this kind of field observation, albeit normally in more culturally remote field sites, also began supporting HCI requirements analysis. Suchman (1987) specifically refers to *situated action*, suggesting that activity is goal driven and is tied to the current knowledge state of the individual as well as the task-specific state of the workplace. It is now quite common to find anthropologists or ethnographers working side by side with HCI designers.

1989–1999

State-of-the-Art

The scene now shifts to the period that we can expect to be familiar to most readers. Without a doubt, this should be characterized as the era of the Internet. Although much of the preparation occurred earlier, it was in the 1990s that its major impact began to be felt. The complete timeline is documented by The Hobbes' Internet Timeline (Zakon, 2001). Only a few relevant highlights are presented here. Packet-switching technology was first introduced in 1969 with the building of the ARPANET by the ARPA. The TCP/IP protocol was adopted as the standard protocol for network transmissions in 1982. The first long-haul network e-mail was sent over the ARPANET in 1972 by Ray Tomlinson, a colleague of the author at BBN. The introduction of a Domain Name System, a common address space that eliminated the need to know the IP address of the recipient, took place in 1984. The NSFnet was created in 1984 to link the five major supercomputer centers that were created by the National Science Foundation, and this network became a national backbone to which other networks soon linked. At this point it could really be called an Internet.

In 1999, Tim Berners-Lee, a researcher at CERN, the European Organization for Nuclear Physics Research in France, first described the idea for a hypertext/hypermedia information management system that would be available at CERN for the management of myriad documents to communicate, share information, and share software across groups (Berners-Lee, 1989). It had to work across platforms. It had to move documents across networks. It needed a scheme for identifying local and remote documents (evolved into Uniform Resource Locators, (URLs)) and required a formatting language (Hypertext Markup Language, (HTML)) (Wilson, 1999). Berners-Lee imagined being able to summarize and analyze data across these very large hypertext distributed databases. He initially referred to the concept as *Mesh*, but changed the name to World Wide Web when he began to implement it. Quickly, there was a flurry of browser developments, but MOSAIC, the internet information browser and www client created by the University of Illinios National Center for Supercomputing Applications (NCSA) in 1993 quickly became the dominant technology (NCSA, 2000). Marc Andreessen and some colleagues left the University of Illinois and founded Netscape, thus privatizing the technology

[4]Special arrangements were made with the Xerox Corporation to lease Altos, because none were ever made available for sale.

developed at NCSA. Subsequently, the technology was licensed by many others wishing to get into the Internet game. The NCSA technology provided the foundation for Microsoft's Internet Explorer as well (NCSA, 2000).

There is no need to document here the impact the Internet has had on the business, commercial, and educational worlds. However, the jury may still be out on whether e-commerce will be the major economic force that was once forecast. The major strengths of the technology—(1) near-universal access, (2) relative ease of locating unique data, and (3) data that may need to be backed up, but never replicated—must be balanced against continuing concerns about privacy, security, and impersonalization.

Web Usability

Of course the advent of the Internet and the World Wide Web substantially changed the HCI game again, just when its specialists were beginning to think they had a handle on things. Those inexperienced users who first explored the web via an HTML hypertext browser were wildly enthusiastic. Military officers asked why all software couldn't be this easy to use? It truly reduced the task to point, click, and fill in the blanks. But, contrary to early popular belief, it soon became clear that it was rather easy to get lost in myriad hypertext layers and not realize how you reached that point. The sources of information were so vast that it became difficult to know how to reach a site of particular interest, giving birth to search engines, history lists, and bookmarks. In the drive to make economical use of page space, too many things were (and still are) presented on a single page, making it difficult to find and review material of interest. Efficiency of information retrieval (now often referred to as the quality of service) depended on the capacity of the user's computer, the capacity of the modem in use, the capacity of network communication links accessed, the load on the network and the complexity of the retrieval request and graphics of the to-be-downloaded page. User-system response time became an issue all over again.

Thus, *design* of web pages and the architecture of web sites has become a major occupation for specialists drawn from the technical writing community, the industrial design community, the advertising community, not to mention the usability and HCI communities.

The 1990s also signaled the mushrooming growth of interest and activity in HCI in general. The man on the street began to complain about how difficult it was to program a videorecorder (*Business Week*, 1991). Usability and ease of learning claims were found in advertisements. In some organizations, usability became an informal, but important product requirement. Other major software companies such as Oracle and Fidelity formally incorporated usability requirements into their product development process and created such positions as Vice President, User Interface Design, and Senior Director, Advanced User Interfaces. Promising dot-coms with extensive venture capital created an insatiable market for web pages. A small proportion of these companies actually became concerned about the usability of their pages.

Speech Technology

Speech has been a proverbial interface medium since first envisioned by Licklider in 1961. Unfortunately, the technology to support it only began to reach maturity in the late 1990s. The first serious research support of speech understanding was sponsored by ARPA in 1971. It brought together speech recognition and language specialists as a result of the insight that speech recognition was not going to advance without incorporating information derived from syntax and semantics. Limited progress was made, and after 3 years, ARPA discontinued financing because their management believed there was little real hope that it could ever be accomplished in real time. Serious DARPA research funding resumed in 1984, largely after the promise of Hidden Markoff Models (HMMs) for this application was identified (Baker, 1979). By 1987, virtually all the research was focused on the application of HMMs. It was only in the 1990s that the confluence of improved processor speeds and more efficient HMM pattern matching algorithms led to feasible, software-only (no requirement for special purpose hardware), real time, continuous, speaker independent, large vocabulary recognition on personal computers.

The first successful application of a speech recognition interface was by the United Parcel Service in the 1980s for hands-free package sorting, and at that time required speaking isolated words (digits). It was the 1990s breakthroughs that have opened the gates to more widespread application, and even today cost-effectiveness is limited to special domains, such as telephone call directors and information services.

Cost-Justifying Usability

Marketing the need for usability requirements has become an important part of the HCI job, and in the late 1980s, there was increased attention to the cost-effectiveness justification for including usability as an integral part of design cycle. The idea was first proposed in Mantei and Teorey (1988) and is well documented in Bias and Mayhew (1994). It can now be argued, sometimes quite convincingly, that attention to usability can result in significant cost savings. A frequently cited example is Project Ernestine, in which a GOMS model predicted that a new telephone toll assistance operator console the company was considering buying from an independent vendor would actually cost the company $2 million per year more to operate than the one it was replacing. This argument convinced the client not to purchase it. Subsequent user testing verified these results (Gray, John, & Attwood, 1993). One lesson learned from the early efforts to calculate cost benefits is the importance of documenting the usability and effectiveness of an application in its original form and as development progresses so that savings can be attributed to the changes.

Continued Professional Development

There was a flurry of new textbooks—Dix, Finlay, Abowd, and Beale (1993), Hix & Hartson (1993), Mayhew (1992), Preece et al. (1994), and Shneiderman (1992)—perhaps the most

influential being Jacob Nielsen's (Nielsen, 1993), Neilsen was a pioneer in advocating discount usability methods, by which he meant shortcuts from the more formal (and costly) usability testing designs derived from experimental psychology. There was some research underway at the time to support his recommendations. Methods he advocated include:

1. Prototypes on paper or in simple prototyping environments that only follow a single scenario thread at a time and allow for rapid revision and re-evaluation.
2. Simplified thinking aloud based on notes taken at the time of a user's walkthrough rather than detailed protocol analysis of videotaped sessions.
3. Heuristic evaluation that involves experts exercising the software and applying a limited set of broad user guidelines or heuristics to identify usability problems (Neilsen, 1993).

He made an important contribution by quantifying the results of heuristic evaluations in terms of the number of problems found as a function of the number of evaluators and, for actual usability tests, the recommended number of users required to achieve a given level of confidence in the results (Nielsen, 1993). While Nielsen did not directly advocate it, another useful technique championed during this period was the cognitive walkthrough, which is similar to Nielsen's heuristic evaluation (Lewis, Polson, & Rieman, 1991). In a cognitive walkthrough, the investigator steps through one or more typical tasks for which the interface would be used. At each step, the analysts ask whether, from the perspective of how the user thinks about it, the task is being supported. The current status of these approaches and others is presented and elaborated in other chapters of this handbook.

Computer-Supported Cooperative Work

When the ARPANET was first conceived, the motivation that sold the program was the idea that university scientists would be able, via the computer network, to transfer files back and forth and to collaborate at a distance. This was a part of Bush and Licklider's original vision that was carried forward by Englebart, Nelson, and a host of others. It came as a complete surprise that e-mail was the first widely accepted application of the ARPANET; however, it certainly supported the use of a computer network for collaboration independent of geography and identity (Hafner & Lyon, 1996).

Computer-to-computer communication on a packet-switched network began with the first ARPANET connections in 1969. The TELENET protocol was created soon thereafter and was widely used by specialists in the field to interconnect machines. The first e-mail on a packet-switched network was sent in 1971, and the first person-to-person real-time chat over a long-haul network took place in 1972 (Zacon, 2001). Files were exchanged with difficulty initially, but then more easily through the FTP protocol. However, it has only been with the introduction of e-mail attachments that file transfers have been widely accessible to the public.

Video teleconferencing, desktop video, even use of the telephone, should legitimately be considered under the aegis of computer-supported cooperative work (CSCW), but their discussion is beyond the scope of this chapter.

There were two software threads that could legitimately be called groupware: Lotus Notes Domino and GroupSystems. Lotus Notes originally supported mainly asynchronous collaboration. The original concept for notes was developed at the University of Illinois in the Computer-based Education Research Laboratory, in 1976 then called Plato Notes and later Plato Group Notes, and was used strictly for intragroup communication. The first commercial release of Lotus Notes was in 1989 and was the product of Ray Ozzie and his colleagues, several of whom had a history that included the Plato group. It included online discussion, e-mail, phone books, and document databases (Mathers, 2001). At first it was sold as an internal collaboration product, but as the Internet emerged, Notes quickly adapted to the broader application and has been widely successful. Lotus Notes has not won any awards for usability. Its success is a function of the extent to which it satisfies a corporate need, regardless of its ease of use. (See http://www.iarchitect.com/mtarget.htm for an extensive list of specific interface problems.)

At about the same time, Skip Ellis at Microelectronics and Computer Technology Corporation and Jay Nunamaker and his colleagues, working at the University of Arizona, were building and experimenting with synchronous face-to-face collaborative meeting room computer support for large group meetings. This work built on earlier efforts of Joyner and Tunstall (1970) and others (Kraemer & King, 1986; Jessup & Valacich, 1993). Nunamaker's developments achieved a critical mass in 1989 with the incorporation of the Ventana Corp., a company that offers the product, GroupSystems, initially to support collaborative decision making in a single meeting room setting, but now broadened to include remote collaboration (Nunamaker, Dennis, Valacich, Vogel, & George, 1991; Ventana, 2001).

Remote collaboration, which may still be regarded as a developing technology, creates a whole new set of HCI challenges. Productive work requires embedding the collaborators in the same context and minimizing the time spent in facilitating the communication itself (Are we on the same page? Who just wrote that? Who is controlling the pointer now?), as contrasted with accomplishing useful communication. Making turn-taking transparent and easily understood, and striving to capture the richness of face-to-face communication are challenges that continue to be addressed (Olson et al., 1993).

1999 AND BEYOND

State-of-the-Art

We appear to be heading into the much heralded epoch of ubiquitous computing. Mark Weiser, of PARC, who died in 1999 at the age of 46, coined the term, championed the vision, and led serious development efforts toward this goal. He said,

For thirty years most interface design, and most computer design, has been headed down the path of the "dramatic" machine. Its highest ideal is to make a computer so exciting, so wonderful, so interesting, that we never want to be without it. A less-traveled path I call the "invisible"; its highest ideal is to make a computer so imbedded, so fitting, so natural, that we use it without even thinking about it. (I have also called this notion "Ubiquitous Computing.") (Weiser, 1996)

The PARC developments were focused on ubiquitous computing in an office, including seamless connections among wearable, "deskable" and wall-mounted computers (Weiser, 1993). There is a natural extension to the consumer world in which more and more of our products actually contain computer chips. Donald Norman has gone so far as to predict transparently supplemented hearing, sight, memory, and reasoning via computer(s) implanted under the skin (Norman, 2001a).

The enabling technologies to accomplish much of this are already available. Many of them are listed in Table I.1. The three things that are missing are the multiindustry cooperation, the systems engineering required to make them all work together seamlessly, and, most importantly, from the point of view of this chapter, the human factors knowledge and experience to understand just what it would mean for them to be transparent to the human user. Research is still needed to further interpret this concept.

TABLE I.1. Technologies That Are Impacting Current and Future HCI Developments

Technologies
Global positioning systems (GPSs)
Personal digital assistants (PDAs)
Cellular telephones
Wireless application protocol (WAP)
Bluetooth short-range radio links
Miniature digital television cameras
Liveboards
Wearable computers

Note. HCI = human–computer interface.

Human–Computer Interaction Challenges and Ubiquitous Computing

On the one hand, we want to bury the complexity, but on the other, the users want to have control of the devices with which they are interacting. In the case of the automobile, the computer engine controls of today are transparent because they involve little or no modification of the existing controls. But, as our products become more sophisticated, for example, auto navigation computers and cruise and headway control sensors and computers, there will surely be additional controls. Perhaps an even better example, highlighted in a cogent critique by Donald Norman, is how to make home theater transparent to the user (Norman, 2001b). These two dimensions of future systems, level of sophistication and control, cannot be independently manipulated.

An important feature of such transparency is context sensitivity. Specialists already recognize that one of the keys to successful, natural, human interaction is detecting and accommodating the context in which the user is operating. A very significant part of that context is to know or be able to infer the user's intent. The GPS systems give us the location context. Clever system design can support some level of intent inferencing. Speech recognition has the potential to support more. Nevertheless, one of the challenges of the next several years will be to improve our methods and techniques for intent inferencing.

Portability goes hand in hand with small size. Even if the technical issues associated with seamless and timely downloading of megabits of information into pocket-sized devices are solved, an additional significant challenge is to understand how, indeed even whether, it is possible to provide, in the size of handheld devices, all the information management, display, and manipulation capabilities that a user would like to have. There is general agreement that the Apple Newton and the Palm Pilot have done an excellent job with the basic capabilities they provide, but the potential sophistication of such devices has not begun to be realized, and it is not clear that Palm's solutions will scale to these new requirements. We can expect to be overwhelmed with information and that it will be available to us anywhere via mobile connectivity. There is more work to be done to decide where to draw the line, in the range from wearable devices to liveboards; to decide what to make available in different scale devices (who wants to read a lengthy e-mail or order a new computer on their personal data unit?), and just how much of it should be available on mobile devices.

Interacting through speech has been the holy grail for many years. We can expect continued improvement in the performance of speech recognition systems and the naturalness of text-to-speech systems. However, making these systems more widely useful requires more than improved technology. Most specialists now realize that human-to-human communication is imperfect and is facilitated by a very wide range of sources of contextual cues that will need to be mimicked and interpreted in some way to make human-to-computer communication as facile.

In 2001, the final version of Section 508 of the Rehabilitation Act was approved, requiring equal access for disabled persons to electronic and information technology provided to federal employees and for public access to federal electronic and information technology. This requirement, although acknowledged for a number of years, is now the law. It represents significant challenges for HCI developers, but such efforts should be broadly rewarding. (See chapters 25 and 26 in this handbook.) The fact is that making information accessible to many types of disabled persons usually also enhances its accessibility for the rest of us. Alphonse Chapanis once noted that, when he revised some insurance regulations so that they met the standards for eighth grade readability, college graduates found them more accessible as well.

New Methodologies

As the scope and sophistication of digital systems become ubiquitous, the pressure for improved HCI methodologies will

continue to increase. On the design side, specialists and funding organizations have long sought the ultimate design methodology that would provide seamless integration of the elements of product usability into a single design support tool. Such a tool would include means for documenting the results of cognitive task analyses and naturalistic observation at the time they were collected. The analysis would directly support the generation of representative scenarios of use that, in turn, provide the data for user system requirement specification and system architecture design. The scenarios and requirements then could be converted easily into quantitative models that will be useful for evaluating alternative design concepts. Prototyping tools for formative evaluation by experts and by potential users would build from the scenarios and models, and make multiple levels of prototyping detail realizable as the design matures. In this idealized methodology, the models developed along the way would continue to be available for support of further exploration and evaluation of detailed design alternatives. The final results of the prototyping studies would be directly convertible into efficient application code.

On the usability evaluation side, the scenarios and prototypes would be readily converted into a simulation that would support human-in-the-loop studies of the effectiveness of the resultant designs. The tools would include online performance measures, fully indexed digital video recording, and spoken interaction automatically synchronized into a single visualization and analysis package for summary and reporting.

The Last Word

Over the last four decades, the profession of HCI has emerged. Although it began with relatively mundane human factors questions about the format and layout of programs and data, what has emerged is a collection of activities that can be regarded as central to the society of the 21st century. We are the gatekeepers to ensure that the technological capabilities that are thrust on us will be manageable and responsive to human requirements. This responsibility does not rest with social planners, sociologists, or politicians. It rests with us because we have the opportunity to design the interfaces that translate that technology into tools and products that are useful, usable, meaningful, accessible, rewarding, and fun.

ACKNOWLEDGMENTS

I thank Stuart Card, Richard Granda, Richard Hirsh, Bonnie John, Clayton Lewis, John Makhoul, Bruce Roberts, Marilyn Tremaine, and Ray Tomlinson for many helpful suggestions.

References

Baecker, R. M., & Buxton, W. A. S. (1987). *Readings in human-computer interaction: A multidisciplinary approach*. Los Altos, CA: Morgan Kaufmann.

Baecker, R. M., Grudin, J., Buxton, W. A. S., & Greenberg, S. (1995). *Readings in human-computer interaction: Toward the year 2000* (2nd ed.). San Francisco: Morgan Kaufmann.

Baker, J. (1979). Trainable grammars for speech recognition. In D. Klatt and J. Wolf (Eds.), *Speech Communication Papers for the 97th Meeting of the Acoustical Society of America* (pp. 547–550). Cambridge, MA: MIT Press.

Bewley, W. L., Roberts, T. L., Schroit, D., & Verplank, W. L. (1983). Human factors testing in the design of Xerox's 8010 "Star" Office Workstation. In *Proceedings of the CHI '83 Conference* (pp. 72–77). New York: Association of Computing Machinery.

Berners-Lee, T. (1989). Information management: A proposal. CERN. Available: http://www.w3.org/History/1989/proposal.html

Bias, R. G., & Mayhew, D. J. (1994). *Cost justifying usability*. Cambridge, MA: Academic Press.

Borman, L. (1996). SIGCHI: The early years. *SIGCHI Bulletin* [On-line], *28*(1), 4–6. Available: http://www.acm.org/sigchi/bulletin/1996.1

Brooks, F. P. (1975). *The mythical man-month: Essays on software engineering.* Reading, MA: Addison-Wesley.

Bush, V. (1945). As we may think. *Atlantic Monthly, 176*, 101–108.

Card, S. K., English, W. K., & Burr, B. J. (1978). Evaluation of mouse, rate-controlled isometric joystick, step keys and text keys for text selection on a CRT. *Ergonomics, 21*(8), 601–613.

Card, S. K., Moran, T. P., & Newell, A. (1980a). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM, 23*(7), 396–410.

Card, S. K., Moran, T. P., & Newell, A. (1980b). Computer text editing: An information-processing analysis of a routine cognitive skill. *Cognitive Psychology, 12*, 32–74.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*, Hillsdale, NJ: Erlbaum.

Carroll, J. M. (1997). Human-computer interaction: Psychology as a science of design. In *Annual review of psychology* (Vol. 48, pp. 61–83). Palo Alto, CA: Annual Reviews.

Dickason, E. (2001). Remembering Grace Murray Hopper: A legend in her own time. Available: http://www.norfolk.navy.mil/chips/grace_hopper/file2.htm

Dix, A., Finlay, J., Abowd, G., & Beale, R. (1993). *Human-computer interaction*. Hillsdale, NJ: Prentice Hall.

Dreyfuss, H. (1955). *Designing for people.* New York: Simon & Schuster.

Engle, S. E., & Granda, R. E. (1975). Guidelines for man/display interfaces. *IBM Technical Report TR 00.2720*. Poughkeepsie, NY: IBM.

Englebart, D. (1963). A conceptual framework for the augmentation of man's intellect. In P. W. Howerton, & D. C. Weeks (Eds.), *Vistas in information handling* (Vol. 1, pp. 1–29). Washington, DC: Spartan.

Ericsson, K. A., & Simon, H. A. (1980). Verbal reports as data. *Psychological Review, 3*, 215–251.

Foley, J. D., & Van Dam, A. (1982). *Fundamentals of interactive computer graphics*. Reading, MA: Addison-Wesley.

Gaines, B. R., & Shaw, M. L. G. (1986). From timesharing to the sixth generation: The development of human-computer interaction. Part I. *International Journal of Man-Machine Studies, 24*, 1–27.

Galitz, W. O. (1981). *Handbook of screen format design*. Wellesley, MA: Q. E. D. Information Sciences.

Gold, M. (1969). Time sharing and batch processing: An experimental comparison of their values in a problem-solving situation. *Communications of the ACM, 12*(5), 249–259.

Goldberg, A. (1998). *A history of personal workstations*. Reading, MA Addison-Wesley.

Grant, E. E., & Sackman, H. (1967). An exploratory investigation of programmer performance under on-line and off-line conditions. *IEEE Transactions on Human Factors in Electronics, HFE-8*(1), 33–48.

Gray, W. D., John, B. E., & Atwood, M. E. (1993). Project Ernestine: Validating a GOMS analysis for predicting and explaining real-work task performance. *Human-Computer Interaction, 8*, 237–309.

Hafner, K., & Lyon, M. (1996). *Where wizards stay up late: The origins of the Internet.* New York: Simon & Schuster.

Hanson, W. (1971). User engineering principles for interactive systems. In *AFIPS Conference Proceedings 39, Fall Joint Computer Conference* (pp. 523–532). Montvale, New Jersey: AFIPS Press.

Heidegger, M. (1962). *Being and time* (J. Macquarrie and E. Robinson, Trans.). New York: Harper Row.

Henderson, D. A., Jr. (1986). The trillium user interface design environment. In *Proceedings SIGCHI'86: Human Factors in Computing Systems* (pp. 221–227). New York: Association for Computing Machinery.

Hix, D., & Hartson, H. R. (1993). *Developing user interfaces: Ensuring usability through product and process*. New York: John Wiley & Sons, Inc.

Jessup, L., & Valacich, J. (Eds.). (1993). *Group support systems: New perspectives*. New York: Macmillan.

Joyner, R., & Tunstall, K. (1970). Computer augmented organizational problem solving. *Management Science, 17*(4), 212–225.

Kay, A. (1969). The reactive engine. PhD Thesis, University of Utah, Salt Lake City.

Kemeny, J. G., & Kurtz, T. E. (1966). *BASIC* (3rd ed.). Hanover, NH: Dartmouth College.

Kraemer, K. L., & King, J. L. (1986). Computer-based systems for group decision support: Status of use and problems in development. *Proceedings of the Conference on Computer-Supported Cooperative Work* (pp. 353–375). Austin, TX.

Lampson, B., & Taft, E. (1979). *Alto user's handbook*. Palo Alto, CA: Xerox Palo Alto Research Center.

Landauer, T. (1995). *The trouble with computers*. Cambridge, MA: The MIT Press.

LegacyJ Corp. (2001). COBOL milestones. Available: http://www.legacyj.com/cobol/cobol_history.html

Lewis, C. (1982). Using the "thinking-aloud" method in cognitive interface design. *IBM Research Report RC 9265*. Yorktown Heights, NY: IBM Thomas J. Watson Research Center.

Lewis, C., Polson, P. G., & Rieman, J. (1991). Cognitive walkthrough forms and instructions. *Institute of Cognitive Science Technical Report #ICS 91–14*. Boulder, CO: University of Colorado.

Licklider, J. C. R. (1960). Man-computer symbiosis. *IRE Transaction on Human Factors in Electronics, HFE-1*, 4–11.

Licklider, J. C. R. (1965). *Libraries of the future*. Cambridge, MA: MIT Press.

Mantei, M., & Teorey, T. J. (1988). Cost/benefit analysis for incorporating human factors in the software lifecycle. *Communications of the ACM, 31*(4), 428–439.

Mantei, M., & Teorey, T. J. (1989). Incorporating behavioral techniques into the system development life cycle. *Management Information Systems Quarterly, 13*(3), 257–276.

Marks, S. L., & Armerding, G. W. (1967). The JOSS primer *Memorandum RM-5220–PR*. Santa Monica, CA: The Rand Corp.

Martin, J. (1973). *Design of man-computer dialogues*. Edgewood Cliffs, NJ: Prentice-Hall.

Mathers, B. (2001). Westford, MA: Iris Associates. Available: http://www.notes.net/history.nsf/text?OpenPage

Mayhew, D. (1992). *Principles and guidelines in software user interface design*. Edgewood Cliffs, NJ: Prentice Hall.

Miller, R. B. (1968). Response time in man-computer conversational transactions. *AFIPS Conference Proceedings, 33*, 267–277.

Morfield, M. A., Wiesen, R. A., Grossberg, M., & Yntema, D. B. (1969). Initial experiments on the effects of system delay on on-line problem-solving. *MIT Lincoln Laboratory Technical Note 1969-5*. Lexington, MA: MIT.

Myer, T. H. (1967). *TELCOMP manual*. Cambridge, MA: Bolt Beranek and Newman.

Myers, B. A. (1998). A brief history of human computer interaction technology. *ACM Interactions, 5*(2), 44–54.

Myers, B., Hollan, J., & Cruz I. (1996). Strategic directions in human-computer interaction. *ACM Computing Surveys, 28*(4), 794–809.

National Center for Supercomputing Applications [NCSA]. (2000). Board of Trustees, University of Illinois Associates. Available: http://www.ncsa.uiuc.edu/Divisions/Communications/MosaicHistory/impact.html

Nelson, T. H. (1965). A file structure for the complex, the changing and the indeterminate. *Proceedings of the ACM National Conference* (pp. 84–100). New York: Association for Computing Machinery.

Nelson, T. H. (1973). A conceptual framework for man-machine everything. *Proceedings of the ACM National Conference* (pp. m21–m26). New York: Association for Computing Machinery.

Nelson, T. H. (1974a). *Computer lib*. South Bend, IN: The Distributors.

Nelson, T. H. (1974b). *Dream machines*. South Bend, IN: The Distributors.

Nelson, T. H. (1981). *Literary machines*. San Antonio, TX: Project Xanadu.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

Nickerson, R. S., & Landauer, T. K. (1997). Human-computer interaction: Background and issues. In M. Helender, T. K. Landauer, & P. Prabhu (Eds.), *Handbook of human-computer interaction: Second, completely revised edition* (pp. 3–31). Amsterdam: Elsevier Science B.V.

Nielsen, J. (1993). *Usability engineering*. Cambridge, MA: Academic Press.

Norman, D. (2001a). Cyborgs. *Communications of the ACM, 44*(3), 36–37.

Norman, D. (2001b). The perils of home theater. Available: http://www.jnd.org/dn.mss/ProblemsOfHomeTheater.html

Norman, D. A. (1981a). The trouble with UNIX: The user interface is horrid. *Datamation, 27*(12), 139–150.

Norman, D. A. (1981b). Comments on cognitive engineering, the need for clear "system images," and the safety of nuclear power plants. *Conference Record for 1981 IEEE Standards Workshop on Human Factors and Nuclear Safety* (pp. 91–92, 211–212). New York: IEEE.

Norman, D. A., & Draper, S. (Eds.). (1986). User-centered system design: New perspectives on human-computer interaction. Hillsdale, NJ: Erlbaum.

Norman, D. A. (1988). The psychology of everyday things. New York: Basic Books.

Nunamaker, J. F., Dennis, A. R., Valacich, J. S., Vogel, D. R., & George, A. F. (1991). Electronic meeting systems to support group work. *Communications of the ACM, 34*(7), 40–61.

Nussbaum, B., & Neff, R. (1991). I can't work this thing. *Business Week*, 58–66.