Computing in the Web Age

A Web-Interactive Introduction

Robert J. Dilligan

Computing in the Web Age

A Web- Interactive Introduction

Computing in the Web Age

A Web-Interactive Introduction

Robert J. Dilligan
University of Southern California
Los Angeles, California

KLUWER ACADEMIC PUBLISHERS
New York, Boston, Dordrecht, London, Moscow

eBook ISBN: 0-306-46965-0
Print ISBN: 0-306-45972-8

Created in the United States of America

Visit Kluwer Online at: http://kluweronline.com
and Kluwer's eBookstore at: http://ebooks.kluweronline.com

For Patricia

... più che la stella

Contents

Acknowledgments

I would like to express my gratitude to students, friends, and colleagues who have helped and encouraged me in this project. Special thanks are due to my colleagues Moshe Lazar, Max Schulz, Ross Winterowd, and James Kincaid, all of whom read portions and drafts of this manuscript and made numerous helpful suggestions and comments. I also want to thank my friend Robert Allen for his invaluable assistance in getting the manuscript in final form. All mistakes, blunders, and errors of omission and commission are my own.

Permissions

Screen shots from the Oak Software Repository are reproduced here with permission from Oakland University.

Screen shots and images showing Tintern Abbey on pages 237 and 238 are copyrighted material of Professor Alison Stones of the University of Pittsburgh and are reproduced here with her permission.

Screen shots and images showing Charles Babbage's Calculating Engines on pages 202, 203, and 216 are copyrighted material of the British Museum of Science and Industry and are reproduced here with the Museum's permission. Any reproduction of this material without the Museum's express written permission is hereby prohibited.

Windows 95, Visual Studio 97, and Internet Explorer are all products and registered trademarks of Microsoft Corporation. Screen shots reprinted by permission from Microsoft Corporation.

Screen shots of WS_FTP program are reproduced with permission from Ipswitch, Incorporated.

Screen shots of the ZipFolders Self-Extracting Archive (v 2.03) are reproduced with permission from Mijenix Corporation.

Screen shots from HotBot are reproduced with permission from Wired Digital, Incorporated. Copyright © 1994–1998 Wired Digital, Incorporated. All rights reserved.

"Where to Browse for Art on the Internet" Copyright © 1996 by The New York Times. Reprinted with permission.

Screen shots from Netscape are Copyright © 1998 Netscape Communications Corporation. Used with permission. All rights reserved. These electronic files or pages may not be reprinted or copied without the express written permission of Netscape.

Screen shots of Yahoo! Text and artwork Copyright © 1998 by Yahoo! Incorporated. All rights reserved. Yahoo! and Yahoo! Logo are trademarks of YAHOO! Incorporated.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Incorporated in the United States and other countries. Screen shots from the JAVA Web site are reprinted with permission from Sun Microsystems, Incorporated.

Introduction

This book reflects thirty years of experience in the applications of computer technology to literary research and instruction and in consulting work in office automation and system integration. In that time I have again and again found myself in the position of having to introduce students, both undergraduate and graduate, colleagues and clients to the fundamentals of computer hardware and software. Over the years, as computers became both central and commonplace in professional life, I have been aware of changing attitudes toward this technology. From attitudes that ranged from the disdain of platonic dialecticians for mere technology to intimidation bordering at times almost on terror, people have moved to incorporate this new technology into their frame of reference (*humani nil a me alienum*). The development of the microprocessor and its subsequent use for word processing marked one important watershed. The widespread use of word processors made it more likely than not that people would own their own computers, at least at work, and use them as part of their work-a-day activities. But while word processing provided some increased familiarity with computers, it did not lead most individuals much beyond a knowledge of the usual incantations needed to control the *MultiMate* or *Nota Bene* or *Word Perfect* golam and, as a result of unhappy experience, a begrudging acceptance of the need to make back up copies of important files. For most of us, word processing, presentation systems, E-mail, spreadsheets, and the like are useful computer applications, but none of them is a "killer app," a computer application that not only gets useful work done but also opens possibilities unthinkable save for the application's existence. (The MAC became and remains today—who knows for how long— the "killer app" for graphics design professionals and computer-based instruction.)

By all accounts, the universal "killer app" is the World Wide Web. While individuals worked at their word processors, a number of developments in the fields of networking, libraries, and publishing stole a march on them. Networks developed a worldwide reach. Libraries moved from computerized cataloguing to computerized catalogues accessible through dedicated networks like OCLC. At the same time, the expansion of the number of specialized journals, combined with the rising cost of traditional printing, soon outstripped the capacity of even the largest library serials budgets to keep up with current publications. The Internet, a product of the Department of Defense during the Cold War, developed a mechanism whereby scholars and researchers at widely separated institutions could collaborate using shared computer files. But the Internet, with its UNIX-based character interface, was much too arcane and awkward to find general use among professionals for much the same reasons that early character-based text formatting programs like SCRIPT and TROFF never really found general acceptance. But once the physicists at CERN and the programmers at the University of Illinois developed the World Wide Web with a graphical user interface that was as user-friendly as a MAC, the elements were in place for the expansion of the portion of the Internet called the World Wide Web into many areas of our lives.

We are all being swept along into using the World Wide Web. There are already journals, databases, and bibliographies that are accessible only through the Web. In the future, the Web will increasingly provide immediate access to the most recent work in most fields. This book is intended as a general introduction to computer technology as it relates to the development of the World Wide Web. The book combines discussions of computer technology with exercises in the use of computer applications and programming. I think of this book as providing a studio course whose goal is to provide a unified and integrated understanding of the basics of computer technology. By working through this book and performing the exercises it sets, readers should become able to understand what they are doing from a computational point of view as well as from a pragmatic one. With this dual perspective, they should be able to exploit the possibilities provided by the Web with a thoroughness and depth not possible without it. Rather than covering any particular aspect of this technology exhaustively, the chapters that follow move in a wide-ranging way across the field of computer technology and pause to examine in depth the technologies crucial to the Web.

Chapter 1, "From ENIAC to the World Wide Web to . . . ," offers a brief history of the technologies that lead to the development of the Web and some speculation about the future of computers. This covers early work on theory and hardware during the 1930s; the development of ENIAC and EDVAC during and immediately after World War II; the rise and fall of mainframe computers from the 1950s through the 1970s; the development of microprocessors and networks in the 1980s and 1990s; and the development of the Internet and the Web from 1969 to the present. I then review current thinking about the future of computers, especially as it relates to the possibilities of artificial intelligence, and offer some speculations relating to this topic.

Chapter 2, "Binary Numbers, ASCII Code, and the Turing Machine," introduces the fundamentals of computers and Computer Science. Readers are made familiar with binary numbers, an understanding of which is indispensable in any discussion of computer technology, through a pocket calculator program that is accessible on the Web. Next I discuss the way information is encoded for a computer by examining ASCII code, which is the most widely used method of encoding textual information on a computer. Readers are shown how to use an encryption device that operates in the same way as a Lone Ranger code ring. With the code ring, which is available on the Web, they are able to see how binary numbers underlie the textual material they are accustomed to read on their computer screens. The last part of this chapter explains the working of a Turing machine, a hypothetical, programmable device first described by the computer pioneer Alan Turing. This device provides the mathematical model for all modern computers. The way this hypothetical machine works is demonstrated by a program, available on the Web, that takes readers step by step through the execution of a Turing machine program. The Turing machine explains its operations step by step as it carries them out. By the end of this chapter, readers should be equipped to understand the logical principals of programming and the binary foundation of the Web.

Chapter 3, "The CPU as a Turing Machine," illustrates the principle that any computer is a Turing machine. Building on the understanding of binary numbers and code in Chapter 2, the goal of this chapter is to give readers a real view of the shifting binary patterns of cyberspace. This chapter shows how the ideas of computer software may be embodied in both real and virtual machines. Computers are made up of collections of switching circuits called bits. Understanding how a computer works and how a program carries out its task means understanding how and in what sequences these bit switches are thrown on and off. Readers observe the basic execution cycle of a computer as it moves data between memory and its central processing unit (CPU), counts, makes logical decisions, and computes tables of numbers. The computer they observe doing these things is a hypothetical, virtual computer, specially designed to illustrate the general principles of operation of a computer. They also get an introduction to the basics of programming as they see how a computer program is translated from words and symbols that humans can understand into the binary code that the computer uses, how the binary code is linked to the operating system and loaded into memory, and how it is executed. The virtual computer, like the Turing machine, gives an explanation of each step in its operation as it is performed.

Chapter 4, "The World According to Programmers," applies the principles of programming developed in Chapter 3 with a virtual computer to actual programming for the World Wide Web. First, the chapter explains how a programmer develops the interactive control windows and icons that are used to control programs. These control windows are known as graphical user interfaces (GUI) and are, from the user's point of view, the single most important part of a computer program. Anything they do on the Web must be accomplished through a GUI. The Internet was transformed to the World Wide Web through the provision of a GUI. The chapter explains the principles of design that programmers use in developing a GUI and the programming tools used in implementing one. This behind-the-scenes look at a GUI as it takes shape will clarify for users what they are doing when using one in a way that will enable them to make maximum use of the Web and of all GUI programs. Understanding how a GUI works is the key to developing sophisticated computer skills. After learning about the development of the GUI, users see how it operates as part of a text retrieval program of the sort used to obtain information from the Web. The technique of keyword searching that is explained for this program is the basic technique for all Web searching. The discussion enables readers to understand the principles of information retrieval from the Web. Readers are provided with the opportunity to .download three JAVA programs for text retrieval and a JAVA operating environment, the JAVA Runtime Environment (JRE) from the Web. This exercise introduces readers to downloading from the Web and to the use of compressed data and self-extracting archives, skills that enhance the use of the Web as a source of information. Readers are also shown how to deal with information that is distributed across a number of computer files, as it frequently is on the Web.

Chapter 5, “Connections to the World Wide Web,” begins with a discussion of the reasons for using computer networks like the Web and the issues of privacy, encryption, and governmental intrusion on Web users. It then explains how network software and hardware provide the interconnections of users to each other through Web applications. With this background understanding, the chapter then focuses on using the Netscape browser to interface with and search the Web. This chapter shows readers how to use the Web as a source of information and of programs with which to analyze information from the Web or other sources. The techniques of Web searching and information retrieval are explained with emphasis on the issue of information literacy as a way of understanding the scope and usefulness of information from the Web. The topic of artificial intelligence is used to illustrate the various search engines and options provided by the Web. This topic is searched in various ways and the implications of the information retrieved from each search is evaluated and compared. The chapter then explains how to save for future use information found through Web searches. The types of data that are found integrated on web sites are described. The different techniques used to save information in different formats are illustrated as well. Finally, readers are introduced to the software archives available on the Web. They are shown how to obtain from these archives programs for editing text and graphical information, for compressing data, and for publishing information on the Web.

Chapter 6, “HTML Programming and Web Publishing,” takes the next step in understanding and using the Web by explaining to readers the basics of how they may create their own web sites and place on the Web information they create themselves. Readers are shown how to take an ordinary text file and transform it into a hypertext document suitable for publication on the Web. Drawing on the work with programming in earlier chapters, readers learn how to program using Hyper Text Markup Language (HTML). The basic “tags” and conventions for laying out Web documents are explained as well as the way in which these documents are organized on the Web. The concept of the Uniform Resource Locator (URL) and the use of relative and absolute addressing on web sites is covered in detail. The use and placement of images on files and the methods HTML provides for displaying them on Web pages are detailed. The use of helper applications for multimedia information is discussed. Readers are shown how to add helper applications to their Web browser. Both video and audio applications are covered. Finally, readers are introduced to the UNIX operating system so that they can understand how and where to place hypertext material on a Web server for access on the Web.

Studio courses require studios. In taking this course, readers are encouraged to access the Web with a suitably equipped computer. Before explaining the relationship of this book to the web site that has been designed to offer material supplementary to its text, I’d like to make a general point about the manner of change of computer technology. I would describe this change as punctuated equilibrium, using the term coined by Stephen Gould to describe his version of Darwinism and to distinguish it from the traditional gradualist description of evolutionary change. Rather than viewing evolution as a slow process of cumulative change, Gould argues that it is more accurate to think of evolutionary change as long periods of stability alternating with short periods of rapid adaptation by isolated populations subjected to unusual environmental stress. So too with the development of computers. The main evolutionary lines run from mainframes to minicomputers to microcomputers; from stand-alone computers to networks of similar machines to networks of heterogeneous machines; from proprietary standards to open standards; and from numerical to character to graphical interfaces. The generations of computers are numbered by the technology they use. The first generation, a product of World War II engineering research, used vacuum tube technology. The second generation used transistors. The third generation used integrated circuits that combined a number of transistors on a single piece of silicon. The fourth generation uses microprocessors, an extension of integrated circuits that allowed an entire computer to be placed on a single chip.

Computer capacity measured in terms of the size of memory and the speed of processing doubles about every eighteen months. With this rapid development of capacity, there emerges a dominant technological paradigm that seems to remain stable and unchallenged for about three to five years. The dominant technological paradigm today is characterized by the graphical user interface to the World Wide Web. Most developments today focus on filling out the implications of this technology, in particular with interest in providing live video and sound and on distributing JAVA computer applications on the Web. This book is written in the context of this current, particular period of stability. It is organized to give readers a solid background in the basic of computer technology and then to discuss things like graphical user interfaces, Java and multimedia because these are the things that seem to me to characterize the current state of computing for most users. It discusses the current state of computing in the context of its unchanging fundamentals.

Now for some technical details. To use the web site associated with this book, readers must have access to a suitable Power MAC or Pentium PC connected to the Internet by a telephone modem or through an Ethernet card. Such machines come equipped with at least eight and preferably sixteen megabits of random access memory, a pointing device or mouse, a color monitor, a hard disk, a floppy disk, a CD-ROM drive, and a sound card. (The sound card and CD-ROM are optional for purposes of this book.) The MAC and the PC are the two machines most-widely used in the United States and most of the western world. All exercises and examples in the text of this book will be presented on the PC. The text of the book describes the exercises as carried out with a Pentium PC. The differences between the PC and MAC, though real, are not so substantial as partisans in the MAC/Windows wars would lead one to believe. The sample programs and exercises in this book work for the most part about as well on either machine, though the programs may appear in slightly different form on the MAC. Where the MAC and PC diverge too widely for use of common examples and exercises—as, for example, in Chapters 4 and 5—alternate material is provided for the MAC via the World Wide Web.

The application software most frequently used in this book is either Netscape 3.0 or above and Internet Explorer 4.0. Both are browser programs and may be used interchangeably for purposes of this book. They provide a connection or interface between a MAC or PC and the Web. It is through one of these programs that readers search the Web, read text, run JAVA programs, view images, and listen to sounds. Netscape is the dominant network browser used on the Web. The current versions of both Netscape and Internet Explorer for MAC OS and Windows 95 support JAVA applications.

Instructions for downloading material from the Web site and installing it on a computer are given in Appendix C.

CHAPTER 1

From ENIAC to the World Wide Web to . . .

COMPUTING ON THE WORLD WIDE WEB

As a visit to the computer section of any book store will confirm, there is a plethora of books on the market that offer their readers introductions to just about any topic having to do with computers, computer applications, or computer programming. If their titles are any guide to their contents, these books promise their readers everything from quick and easy introductions (*Learn Windows 95 in 48 Hours*), to crash courses for career advancement (*Become a C++ Programmer in 21 Days*), to new canons (*The Microsoft C Bible*), to cabalistic initiations (*Mastering Paradox 4.5*), to satori (*Zen and the Art ofthe Internet*), to tabloid revelations (*Inside Visual Basic*), to self-assured self-deprecation (*The Internet for Dummies*). This book is a general introduction to computer technology. The readers it addresses are professionals who do not have a technical background. They may use a word processor to write their memos, reports, articles and books, reference library catalogues over a network, occasionally browse on the World Wide Web and use Email to communicate with colleagues. But they do these things with only the most general notion of the workings of computer technology. To them, using a computer is like evoking a magic spell with a series of incomprehensible but powerful incantations.

This book does not attempt to provide its readers with a quick yet thorough introduction to any particular topic, to open new career paths for them, or to initiate them into mysteries sacred or profane. It singles out for discussion aspects of computer technology that form the basis of modern computer applications. By combining theoretical and technical discussion of computer technology with hands-on exercises using the computer, it attempts not only to provide an understanding of the characteristic perspective, sensibility, and the habits of thought of those who design computer software and hardware but also to combine that new technical perspective with the readers' own. The ultimate aim of this book is to provide the opportunity for a true experience of computer technology that will encourage readers to explore its usefulness for their own professions.

This book, then, hopes to be generally useful to readers whose inclinations are not, first of all, technical. My own profession, I admit at the outset, is literary studies and so readers in other professions will sense, no doubt, a tinge of the literary in this text. This tinge is present if only because in writing this book I have followed Sir Phillip Sidney's prescription for overcoming writer's block: "Fool,...look in thy heart and write." The present conjunction of the Internet and Graphical User Interfaces (GUIs)[1] that has produced the World Wide Web (hereinafter the Web) is a development in computer technology with important cultural implications. As such it deserves to be studied and recorded and that is one aim of this book. This book is like a studio course in drawing that combines the study of anatomy with an exposition of the techniques of sketching. It combines discussion of computer technology with exercises on the computer and illustrative examples. At the end of the course, I hope that readers will be able to look at a computer system or application with an appreciation of the art and science concealed there and feel confident enough to embark on their own explorations of the medium. I will begin by reviewing those aspects of the development of the modern computer that shed some light on basic concepts underlying its operation.

ENIAC AND THE DEVELOPMENT OF THE MODERN COMPUTER

The modern computer emerged into public consciousness at the end of World War II as an important advance in our capacity to carry out scientific calculations. There is a finite limit not only to the speed at which human beings, even idiot savants, can make complex calculations requiring many steps but also to the speed at which organized teams of individuals can carry out these calculations. The speed at which results can be calculated is not proportional to the number of individuals set to the task.[2] At some point, a form of entropy sets in as the time and complexity needed for team members to share results overcomes the advantages of adding members to the team. Increasing the size of the team may, in some circumstances, actually increase the time it takes to complete the task. As military technology advanced during World War II, the need for rapid calculations to solve problems associated with the design and operation of advanced weapons increased enormously. The limits of human calculation were reached in that war, during which it proved impossible to keep up with the demand for calculations for mundane and esoteric problems. The War Department and its scientific agencies were overwhelmed with demands for calculations as mundane as those needed to produce artillery firing tables and as esoteric as those needed to design atomic weapons.

In its early history, the computer was regarded as a number cruncher. It provided a quick and accurate way to produce arithmetical results that were too tedious to generate by error-prone humans.[3] Although its use during the war for the encryption and decryption of codes suggested the potential of the computer as a general-purpose manipulator of logical symbols, its usefulness was generally regarded as limited to scientific and engineering calculations. The fact that its use in encryption and decryption remained top secret until long after World War II[4] may have contributed to an emphasis on the publicly acknowledged view of the computer as scientific calculator. The market for computers was seen early on as limited to government bureaucracies such as the military and the weather service and to scientific centers like the Institute for Advanced Studies at Princeton. One 1946 marketing survey estimated that there was a worldwide market for about one dozen computers. As things developed, the market for scientific computers was served by manufacturers, like Cray Research, that specialized in the design and production of supercomputers. The need for this type of computing is met today at the various national computer centers funded through the National Science Foundation. It amounts to a tiny fraction of current computer usage. Had their main use been for scientific calculation, computers would today be regarded as laboratory curiosities. The emergence of computers as what David Bolter has termed the "defining technology" of our culture[5] rests more on their prowess as manipulators of logical symbols and information retrieval systems than on their arithmetic speed and accuracy.

ENIAC, which became known as the first modern electronic computer, was constructed at the University of Pennsylvania under the direction of John Mauchly and J. Presper Eckert. ENIAC is an acronym for Electronic Numeric Integrator And Calculator. As the name suggests, it was from the first regarded as a device for scientific calculation. The important things about ENIAC were that it was digital, reliable, and programmable and could obtain results about five hundred times faster than was possible with earlier human or machine-assisted methods.[6] Its ability to exceed the human capacity for rapid calculation assured its place as an important technological advance. Among the first tasks to which ENIAC was set were calculations needed for the design of atomic weapons.

A number of advances in both theoretical and applied twentieth-century science contributed to the development of the modern computer. It has often been the case historically with major scientific advances that the intellectual climate of the time seems ripe for the advance that emerges more or less simultaneously in more than one place. It is often difficult to say who exactly should get credit for an advance or when exactly it occurred. Leibnitz and Newton arrived at the calculus independently. Evolutionary theory is called Darwinism, but Wallace in the jungles of Asia formulated a principle of natural selection identical with Darwin's formulation from the voyage of the Beagle. To Crick and Watson at Cambridge goes the credit for the discovery of the structure of DNA. But as Watson's *The Double Helix* makes clear, Rosalind Franklin or Maurice Wilkins at King's College, London, or Linus Pauling at Cal Tech would have come up with the structure of DNA sooner or later. The modem computer did not exist in 1936 and clearly exists by the end of 1945. A consideration of how it developed is another reminder, if one is needed, about the difference between instant replay and history.

For the development of computers, the giants of modem Physics, people like Plank, Einstein, and Heisenberg, did the fundamental theoretical work that made possible modem electronics, especially transistors and integrated circuits. (One of the problems that limited the success of nineteenth-century computer pioneer Charles Babbage was his dependence on mechanical technology.) But above all, one must acknowledge the work of Alan Turing. His work during the nineteen thirties on computable numbers laid the theoretical foundation of Computer Science, before modem computers were invented. Turing developed a mathematical model that underlies all modem computers. This model is called a "Universal Turing Machine," a hypothetical device that demonstrates the possibility of a powerful programmable machine. During World War II, he was engaged in highly classified work on a British computer known as Colossus whose purpose was to decode military codes generated by the German Enigma coding machine. After the War, in a famous 1950 article, "Computing Machinery and Intelligence,"[7] he proposed a conversational game the outcome of which was to answer the question of whether machines could be programmed to be intelligent. The set-up for the game was that a judge would be seated at computer terminal into which he would type questions and from which he would receive answers. After five minutes of questions and answers, he was to decide whether the answers were being supplied by an out-of-sight human or by a computer program. If, at least seventy percent of the time, the program succeeded in fooling the judge into thinking its answers were supplied by a human, the program was to be regarded as intelligent. Turing predicted that such a program would be devised by the year 2000. This game has become known as the "Turing Test." Turing, then, is the founder of both Computer Science and the field of Artificial Intelligence. There is today an annual contest, the Loebner Prize Competition in Artificial Intelligence, to determine which of the program entries most successfully passes the test. The dialogs of winning programs are available on the Web.[8] Turing's life took a tragic turn in 1952, when he was arrested for homosexuality. He refused to defend himself against the charge, asserting that he saw nothing wrong in his actions. Unlike Oscar Wilde who, a half century before, had been jailed for his sexual orientation, Turing was forced to undergo hormonal treatments. His life and health were shattered by the consequences of his arrest. In 1954, he committed suicide by eating an apple soaked in cyanide. I imagine that Turing would have appreciated today's efforts to insure personal privacy and prevent governmental intrusion on the Web.

In addition to Turing's theoretical work on the mathematics of computing, there were many advances in applied technology that contributed to the development of