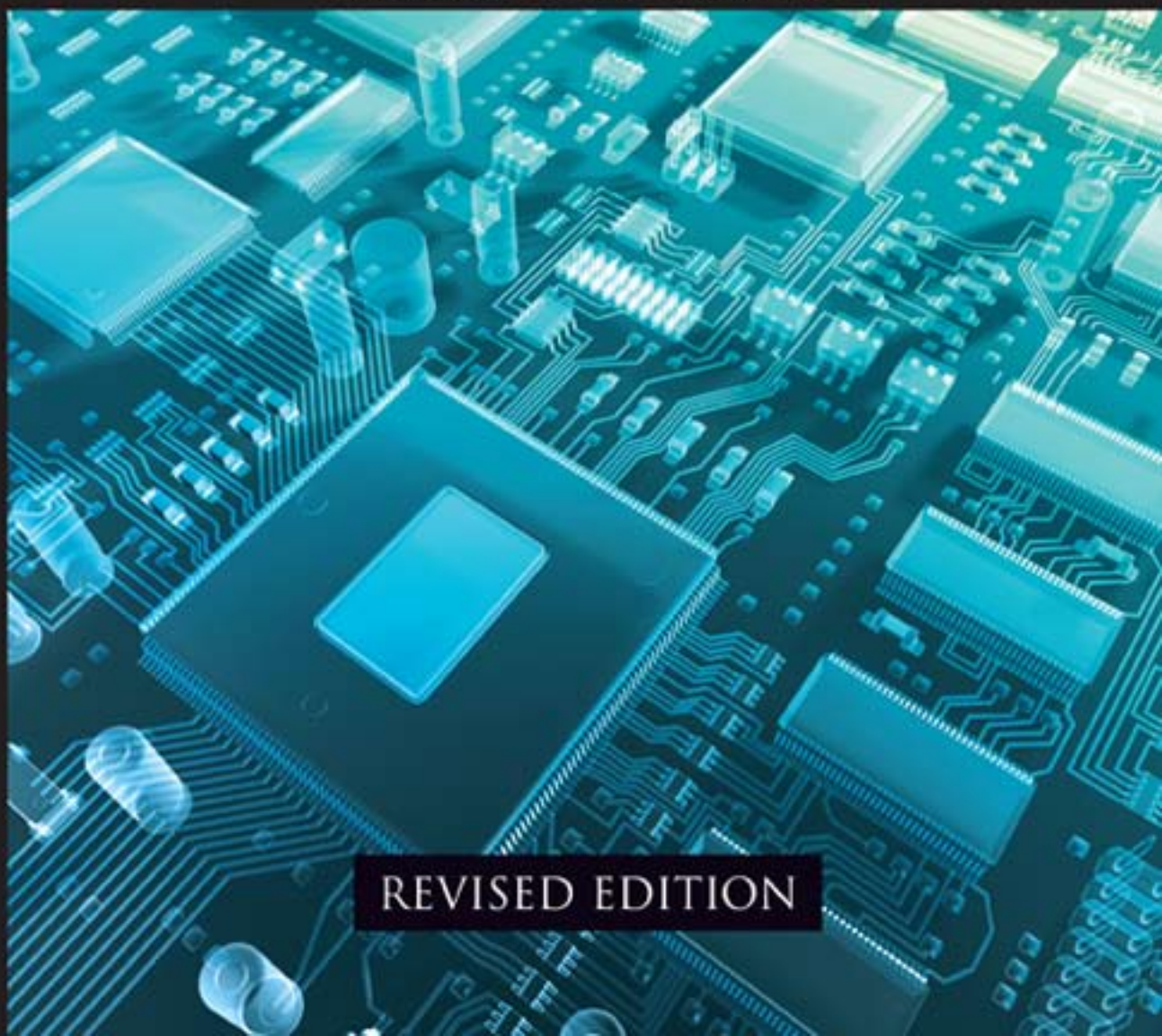
FACTS ON FILE SCIENCE LIBRARY

ENCYCLOPEDIA OF COMPUTER SCIENCE AND TECHNOLOGY

HARRY HENDERSON



REVISED EDITION

ENCYCLOPEDIA OF COMPUTER SCIENCE AND TECHNOLOGY

REVISED EDITION

HARRY HENDERSON

Facts On File
An imprint of Infobase Publishing

In memory of my brother,
Bruce Henderson,
who gave me my first opportunity to explore
personal computing almost 30 years ago.

ENCYCLOPEDIA OF COMPUTER SCIENCE AND TECHNOLOGY, Revised Edition

Facts On File, Inc.
An imprint of Infobase Publishing
132 West 31st Street
New York NY 10001

Library of Congress Cataloging-in-Publication Data

Henderson, Harry, 1951–
Encyclopedia of computer science and technology / Harry Henderson.—Rev. ed.
p. cm.
Includes bibliographical references and index.
ISBN-13: 978-0-8160-6382-6
ISBN-10: 0-8160-6382-6
1. Computer science—Encyclopedias. 2. Computers—Encyclopedias. I. Title.
QA76.15.H43 2008
004.03—dc22 2008029156

Facts On File books are available at special discounts when purchased in bulk quantities for businesses, associations, institutions, or sales promotions. Please call our Special Sales Department in New York at (212) 967-8800 or (800) 322-8755.

You can find Facts On File on the World Wide Web at http://www.factsonfile.com

Text design by Erika K. Arroyo
Cover design by Salvatore Luongo
lllustrations by Sholto Ainslie
Photo research by Tobi Zausner, Ph.D.

Printed in the United States of America

VB Hermitage 10 9 8 7 6 5 4 3 2 1

This book is printed on acid-free paper and contains
30 percent postconsumer recycled content.

CONTENTS

ACKNOWLEDGMENTS
iv

INTRODUCTION TO THE REVISED EDITION
v

A–Z ENTRIES
1

APPENDIX I
Bibliographies and Web Resources
527

APPENDIX II
A Chronology of Computing
529

APPENDIX III
Some Significant Awards
542

APPENDIX IV
Computer-Related Organizations
553

INDEX
555

ACKNOWLEDGMENTS

I wish to acknowledge with gratitude the patient and thorough management of this project by my editor, Frank K. Darmstadt. I can scarcely count the times he has given me encouragement and nudges as needed. I also wish to thank Tobi Zausner, Ph.D., for her ability and efficiency in obtaining many of the photos for this book.

INTRODUCTION TO THE REVISED EDITION

Chances are that you use at least one computer or computer-related device on a daily basis. Some are obvious: for example, the personal computer on your desk or at your school, the laptop, the PDA that may be in your briefcase. Other devices may be a bit less obvious: the "smart" cell phone, the iPod, a digital camera, and other essentially specialized computers, communications systems, and data storage systems. Finally, there are the "hidden" computers found in so many of today's consumer products—such as the ones that provide stability control, braking assistance, and navigation in newer cars.

Computers not only seem to be everywhere, but also are part of so many activities of daily life. They bring together willing sellers and buyers on eBay, allow you to buy a book with a click on the Amazon.com Web site, and of course put a vast library of information (of varying quality) at your fingertips via the World Wide Web. Behind the scenes, inventory and payroll systems keep businesses running, track shipments, and more problematically, keep track of where people go and what they buy. Indeed, the infrastructure of modern society, from water treatment plants to power grids to air-traffic control, depends on complex software and systems.

Modern science would be inconceivable without computers to gather data and run models and simulations. Whether bringing back pictures of the surface of Mars or detailed images to guide brain surgeons, computers have greatly extended our knowledge of the world around us and our ability to turn ideas into engineering reality.

The revised edition of the *Facts On File Encyclopedia of Computer Science and Technology* provides overviews and important facts about these and dozens of other applications of computer technology. There are also many entries dealing with the fundamental concepts underlying computer design and programming, the Internet, and other topics such as the economic and social impacts of the information society.

The book's philosophy is that because computer technology is now inextricably woven into our everyday lives, anyone seeking to understand its impact must not only know how the bits flow, but also how the industry works and where it may be going in the years to come.

NEW AND ENHANCED COVERAGE

The need for a revised edition of this encyclopedia becomes clear when one considers the new products, technologies, and issues that have appeared in just a few years. (Consider that at the start of the 2000 decade, Ajax was still only a cleaning product and *blog* was not even a word.)

The revised edition includes almost 180 new entries, including new programming languages (such as C# and Ruby), software development and Web design technologies (such as the aforementioned Ajax, and Web services), and expanded coverage of Linux and other open-source software. There are also entries for key companies in software, hardware, and Web commerce and services.

Many other new entries reflect new ways of using information technology and important social issues that arise from such use, including the following:

- blogging and newer forms of online communication that are influencing journalism and political campaigns
- other ways for users to create and share content, such as file-sharing networks and YouTube
- new ways to share and access information, such as the popular Wikipedia
- the ongoing debate over who should pay for Internet access, and whether service providers or governments should be able to control the Web's content
- the impact of surveillance and data mining on privacy and civil liberties

- threats to data security, ranging from identity thieves and "phishers" to stalkers and potential "cyberterrorists"
- the benefits and risks of social networking sites (such as MySpace)
- the impact of new technology on women and minorities, young people, the disabled, and other groups

Other entries feature new or emerging technology, such as

- portable media devices (the iPod and its coming successors)
- home media centers and the gradual coming of the long-promised "smart house"
- navigation and mapping systems (and their integration with e-commerce)
- how computers are changing the way cars, appliances, and even telephones work
- "Web 2.0"—and beyond

Finally, we look at the farther reaches of the imagination, considering such topics as

- nanotechnology
- quantum computing
- science fiction and computing
- philosophical and spiritual aspects of computing
- the ultimate "technological singularity"

In addition to the many new entries, all existing entries have been carefully reviewed and updated to include the latest facts and trends.

Getting the Most Out of This Book

This encyclopedia can be used in several ways: for example, you can look up specific entries by referring from topics in the index, or simply by browsing. The nearly 600 entries in this book are intended to read like "mini-essays," giving not just the bare definition of a topic, but also developing its significance for the use of computers and its relationship to other topics. Related topics are indicated by SMALL CAPITAL LETTERS. At the end of each entry is a list of books, articles, and/or Web sites for further exploration of the topic.

Every effort has been made to make the writing accessible to a wide range of readers: high school and college students, computer science students, working computer professionals, and adults who wish to be better informed about computer-related topics and issues.

The appendices provide further information for reference and exploration. They include a chronology of significant events in computing; a listing of achievements in computing as recognized in major awards; an additional bibliography to supplement that given with the entries; and finally, brief descriptions and contact information for some important organizations in the computer field.

This book can also be useful to obtain an overview of particular areas in computing by reading groups of related entries. The following listing groups the entries by category.

AI and Robotics

artificial intelligence
artificial life
Bayesian analysis
Breazeal, Cynthia
Brooks, Rodney
cellular automata
chess and computers
cognitive science
computer vision
Dreyfus, Hubert L.
Engelberger, Joseph
expert systems
Feigenbaum, Edward
fuzzy logic
genetic algorithms
handwriting recognition
iRobot Corporation
knowledge representation
Kurzweil, Raymond C.
Lanier, Jaron
Maes, Pattie
McCarthy, John
Minsky, Marvin Lee
MIT Media Lab
natural language processing
neural interfaces
neural network
Papert, Seymour
pattern recognition
robotics
singularity, technological
software agent
speech recognition and synthesis
telepresence
Weizenbaum, Joseph

Business and E-Commerce Applications

Amazon.com
America Online (AOL)
application service provider (ASP)
application software
application suite
auctions, online
auditing in data processing
banking and computers
Bezos, Jeffrey P.
Brin, Sergey
business applications of computers
Craigslist
customer relationship management (CRM)
decision support system
desktop publishing (DTP)

enterprise computing
Google
groupware
home office
management information system (MIS)
middleware
office automation
Omidyar, Pierre
online advertising
online investing
online job searching and recruiting
optical character recognition (OCR)
Page, Larry
PDF (Portable Document Format)
personal health information management
personal information manager (PIM)
presentation software
project management software
smart card
spreadsheet
supply chain management
systems analyst
telecommuting
text editor
transaction processing
trust and reputation systems
word processing
Yahoo!

Computer Architecture

addressing
arithmetic logic unit (ALU)
bits and bytes
buffering
bus
cache
computer engineering
concurrent programming
cooperative processing
Cray, Seymour
device driver
distributed computing
embedded system
grid computing
parallel port
reduced instruction set computer (RISC)
serial port
supercomputer
USB (Universal Serial Bus)

Computer Industry

Adobe Systems
Advanced Micro Devices (AMD)
Amdahl, Gene Myron
Apple Corporation
Bell, C. Gordon
Bell Laboratories
benchmark
certification of computer professionals
Cisco Systems
compatibility and portability
computer industry
Dell, Inc.
education in the computer field
employment in the computer field
entrepreneurs in computing
Gates, William III (Bill)
Grove, Andrew
IBM
Intel Corporation
journalism and the computer industry
marketing of software
Microsoft Corporation
Moore, Gordon E.
Motorola Corporation
research laboratories in computing
standards in computing
Sun Microsystems
Wozniak, Steven

Computer Science Fundamentals

Church, Alonzo
computer science
computability and complexity
cybernetics
hexadecimal system
information theory
mathematics of computing
measurement units used in computing
Turing, Alan Mathison
von Neumann, John
Wiener, Norbert

Computer Security and Risks

authentication
backup and archive systems
biometrics
computer crime and security
computer forensics
computer virus
copy protection
counterterrorism and computers
cyberstalking and harassment
cyberterrorism
Diffie, Bailey Whitfield
disaster planning and recovery
encryption
fault tolerance
firewall
hackers and hacking
identity theft
information warfare
Mitnick, Kevin D.
online frauds and scams
phishing and spoofing
RFID (radio frequency identification)

risks of computing
Spafford, Eugene H.
spam
spyware and adware
Y2K Problem

Databases

CORBA (Common Object Request Broker Architecture)
data conversion
data dictionary
data mining
data security
data warehouse
database administration
database management system (DBMS)
database
hashing
information retrieval
Oracle Corporation
SAP
SOAP (Simple Object Access Protocol)
SQL

Data Communications and Networking (General)

bandwidth
Bluetooth
broadband
cable modem
client-server computing
data acquisition
data communications
data compression
DSL (digital subscriber line)
error correction
fiber optics
file server
file transfer protocols
FireWire
local area network (LAN)
modem
network
satellite Internet service
Shannon, Claude E
synchronous/asynchronous operation
telecommunications
terminal
Wifi
wireless computing

Data Types and Algorithms

algorithm
array
binding
bitwise operations
Boolean operators
branching statements
characters and strings
class
constants and literals
data
data abstraction
data structures
data types
enumerations and sets
heap (data structure)
Knuth, Donald
list processing
numeric data
operators and expressions
sorting and searching
stack
tree
variable

Development of Computers

Aiken, Howard
analog and digital
analog computer
Atanasoff, John Vincent
Babbage, Charles
calculator
Eckert, J. Presper
history of computing
Hollerith, Hermann
Mauchly, John William
mainframe
minicomputer
Zuse, Konrad

Future Computing

bioinformation
Dertouzos, Michael
Joy, Bill
molecular computing
nanotechnology
quantum computing
trends and emerging technologies
ubiquitous computing

Games, Graphics, and Media

animation, computer
art and the computer
bitmapped image
codec
color in computing
computer games
computer graphics
digital rights management (DRM)
DVR (digital video recording)
Electronic Arts
film industry and computing
font
fractals in computing
game consoles
graphics card

graphics formats
graphics tablet
image processing
media center, home
multimedia
music and video distribution, online
music and video players, digital
music, computer
online gambling
online games
photography, digital
podcasting
PostScript
RSS (real simple syndication)
RTF (Rich Text Format)
sound file formats
streaming (video or audio)
Sutherland, Ivan Edward
video editing, digital
YouTube

Hardware Components

CD-ROM and DVD-ROM
flash drive
flat-panel display
floppy disk
hard disk
keyboard
monitor
motherboard
networked storage
optical computing
printers
punched cards and paper tape
RAID (redundant array of inexpensive disks)
scanner
tape drives

Internet and World Wide Web

active server pages (ASP)
Ajax (Asynchronous JavaScript and XML)
Andreessen, Marc
Berners-Lee, Tim
blogs and blogging
bulletin board systems (BBS)
Bush, Vannevar
cascading style sheets (CSS)
Cerf, Vinton G.
certificate, digital
CGI (common gateway interface)
chat, online
chatterbots
conferencing systems
content management
cookies
Cunningham, Howard (Ward)
cyberspace and cyber culture
digital cash (e-commerce)
digital convergence
domain name system (DNS)
eBay
e-books and digital libraries
e-commerce
e-mail
file-sharing and P2P networks
flash and smart mob
HTML, DHTML, and XHTML
hypertext and hypermedia
Internet
Internet applications programming
Internet cafes and "hot spots"
Internet organization and governance
Internet radio
Internet service provider (ISP)
Kleinrock, Leonard
Licklider, J. C. R.
mashups
Netiquette
netnews and newsgroups
online research
online services
portal
Rheingold, Howard
search engine
semantic Web
social networking
TCP/IP
texting and instant messaging
user-created content
videoconferencing
virtual community
Wales, Jimmy
Web 2.0 and beyond
Web browser
Web cam
Web filter
Webmaster
Web page design
Web server
Web services
wikis and Wikipedia
World Wide Web
XML

Operating Systems

demon
emulation
file
input/output (I/O)
job control language
kernel
Linux
memory
memory management
message passing
microsoft windows
MS-DOS
multiprocessing

multitasking
operating system
OS X
system administrator
regular expression
Ritchie, Dennis
shell
Stallman, Richard
Torvalds, Linus
UNIX

Other Applications

bioinformatics
cars and computing
computer-aided design and manufacturing (CAD/CAM)
computer-aided instruction (CAI)
distance education
education and computers
financial software
geographical information systems (GIS)
journalism and computers
language translation software
law enforcement and computers
legal software
libraries and computing
linguistics and computing
map information and navigation systems
mathematics software
medical applications of computers
military applications of computers
scientific computing applications
smart buildings and homes
social sciences and computing
space exploration and computers
statistics and computing
typography, computerized
workstation

Personal Computer Components

BIOS (Basic Input-Output System)
boot sequence
chip
chipset
clock speed
CPU (central processing unit)
green PC
IBM PC
laptop computer
microprocessor
personal computer (PC)
PDA (personal digital assistant)
plug and play
smartphone
tablet PC

Program Language Concepts

authoring systems
automatic programming
assembler
Backus-Naur Form (BNF)
compiler
encapsulation
finite state machine
flag
functional languages
interpreter
loop
modeling languages
nonprocedural languages
ontologies and data models
operators and expressions
parsing
pointers and indirection
procedures and functions
programming languages
queue
random number generation
real-time processing
recursion
scheduling and prioritization
scripting languages
Stroustrup, Bjarne
template
Wirth, Niklaus

Programming Languages

Ada
Algol
APL
awk
BASIC
C
C#
C++
Cobol
Eiffel
Forth
FORTRAN
Java
JavaScript
LISP
LOGO
Lua
Pascal
Perl
PHP
PL/1
Prolog
Python
RPG
Ruby
Simula
Tcl
Smalltalk
VBScript

Social, Political, and Legal Issues

anonymity and the Internet
censorship and the Internet

computer literacy
cyberlaw
developing nations and computing
digital divide
disabled persons and computing
e-government
electronic voting systems
globalization and the computer industry
government funding of computer research
identity in the online world
intellectual property and computing
Lessig, Lawerence
net neutrality
philosophical and spiritual aspects of computing
political activism and the Internet
popular culture and computing
privacy in the digital age
science fiction and computing
senior citizens and computing
service-oriented architecture (SOA)
social impact of computing
Stoll, Clifford
technology policy
women and minorities in computing
young people and computing

Software Development and Engineering

applet
application program interface (API)
bugs and debugging
CASE (computer-aided software engineering)
design patterns
Dijkstra, Edsger
documentation of program code
documentation, user
document model
DOM (document Object Model)
error handling
flowchart
Hopper, Grace Murray
information design
internationalization and localization
library, program
macro
Microsoft .NET
object-oriented programming (OOP)
open source movement
plug-in
programming as a profession
programming environment
pseudocode
quality assurance, software
reverse engineering
shareware
Simonyi, Charles
simulation
software engineering
structured programming
systems programming
virtualization

User Interface and Support

digital dashboard
Engelbart, Doug
ergonomics of computing
haptic interface
help systems
installation of software
Jobs, Steven Paul
Kay, Alan
Macintosh
mouse
Negroponte, Nicholas
psychology of computing
technical support
technical writing
touchscreen
Turkle, Sherry
ser groups
user interface
virtual reality
wearable computers

A

abstract data type

See DATA ABSTRACTION.

active server pages (ASP)

Many users think of Web pages as being like pages in a book, stored intact on the server, ready to be flipped through with the mouse. Increasingly, however, Web pages are dynamic—they do not actually exist until the user requests them, and their content is determined largely by what the user requests. This demand for greater interactivity and customization of Web content tends to fall first on the server (see CLIENT-SERVER COMPUTING and WEB SERVER) and on "server side" programs to provide such functions as database access. One major platform for developing Web services is Microsoft's Active Server Pages (ASP).

In ASP programmers work with built-in objects that represent basic Web page functions. The RecordSet object can provide access to a variety of databases; the Response object can be invoked to display text in response to a user action; and the Session object provides variables that can be used to store information about previous user actions such as adding items to a shopping cart (see also COOKIES).

Control of the behavior of the objects within the Web page and session was originally handled by code written in a scripting language such as VBScript and embedded within the HTML text (see HTML and VBSCRIPT). However, ASP .NET, based on Microsoft's latest Windows class libraries (see MICROSOFT .NET) and introduced in 2002, allows Web services to be written in full-fledged programming languages such as Visual Basic .NET and C#, although in-page scripting can still be used. This can provide several advantages: access to software development tools and methodologies available for established programming languages, better separation of program code from the "presentational" (formatting) elements of HTML, and the speed and security associated with compiled code. ASP .NET also emphasizes the increasingly prevalent Extensible Markup Language (see XML) for organizing data and sending those data between objects using Simple Object Access Protocol (see SOAP).

Although ASP .NET was designed to be used with Microsoft's Internet Information Server (IIS) under Windows, the open-source Mono project (sponsored by Novell) implements a growing subset of the .NET classes for use on UNIX and Linux platforms using a C# compiler with appropriate user interface, graphics, and database libraries.

An alternative (or complementary) approach that has become popular in recent years reduces the load on the Web server by avoiding having to resend an entire Web page when only a small part actually needs to be changed. See AJAX (asynchronous JavaScript and XML).

Further Reading

Bellinaso, Marco. *ASP .NET 2.0 Website Programming: Problem—Design—Solution*. Indianapolis: Wiley Publishing, 2006.

Liberty, Jesse, and Dan Hurwitz. *Programming ASP .NET*. 3rd ed. Sebastapol, Calif.: O'Reilly, 2005.

McClure, Wallace B., et al. *Beginning Ajax with ASP .NET*. Indianapolis: Wiley Publishing, 2006.

Mono Project. Available online. URL: http://www.mono-project.com/Main_Page. Accessed April 10, 2007.

Ada

Starting in the 1960s, the U.S. Department of Defense (DOD) began to confront the growing unmanageability of its software development efforts. Whenever a new application such as a communications controller (see EMBEDDED SYSTEM) was developed, it typically had its own specialized programming language. With more than 2,000 such languages in use, it had become increasingly costly and difficult to maintain and upgrade such a wide variety of incompatible systems. In 1977, a DOD working group began to formally solicit proposals for a new general-purpose programming language that could be used for all applications ranging from weapons control and guidance systems to barcode scanners for inventory management. The winning language proposal eventually became known as Ada, named for 19th-century computer pioneer Ada Lovelace see also BABBAGE, CHARLES). After a series of reviews and revisions of specifications, the American National Standards Institute officially standardized Ada in 1983, and this first version of the language is sometimes called Ada-83.

Language Features

In designing Ada, the developers adopted basic language elements based on emerging principles (see STRUCTURED PROGRAMMING) that had been implemented in languages developed during the 1960s and 1970s (see ALGOL and PASCAL). These elements include well-defined control structures (see BRANCHING STATEMENTS and LOOP) and the avoidance of the haphazard jump or "goto" directive.

Ada combines standard structured language features (including control structures and the use of subprograms) with user-definable data type "packages" similar to the classes used later in C++ and other languages (see CLASS and OBJECT-ORIENTED PROGRAMMING). As shown in this simple example, an Ada program has a general form similar to that used in Pascal. (Note that words in boldface type are language keywords.)

```
with Ada.Text_IO; use Ada.Text_IO;
procedure Get_Name is
Name : String (1..80);
Length : Integer;

begin
Put ("What is your first name?");
Get_Line (Name, Length);
New_Line;
Put ("Nice to meet you,");
Put (Name (1..Length));
end Get_Name;
```

The first line of the program specifies what "packages" will be used. Packages are structures that combine data types and associated functions, such as those needed for getting and displaying text. The Ada.Text.IO package, for example, has a specification that includes the following:

```
package Text_IO is
type File_Type is limited private;
type File_Mode is (In_File, Out_File, Append_File);
procedure Create (File : in out File_Type;
Mode : in File_Mode := Out_File;
Name : in String := "");
procedure Close (File : in out File_Type);
procedure Put_Line (File : in File_Type;
Item : in String);
procedure Put_Line (Item : in String);
end Text_IO;
```

The package specification begins by setting up a data type for files, and then defines functions for creating and closing a file and for putting text in files. As with C++ classes, more specialized packages can be derived from more general ones.

In the main program **Begin** starts the actual data processing, which in this case involves displaying a message using the Put function from the Ada.Text.IO function and getting the user response with Get_Line, then using Put again to display the text just entered.

Ada is particularly well suited to large, complex software projects because the use of packages hides and protects the details of implementing and working with a data type. A programmer whose program uses a package is restricted to using the visible interface, which specifies what parameters are to be used with each function. Ada compilers are carefully validated to ensure that they meet the exact specifications for the processing of various types of data (see DATA TYPES), and the language is "strongly typed," meaning that types must be explicitly declared, unlike the case with C, where subtle bugs can be introduced when types are automatically converted to make them compatible.

Because of its application to embedded systems and real-time operations, Ada includes a number of features designed to create efficient object (machine) code, and the language also makes provision for easy incorporation of routines written in assembly or other high-level languages. The latest official version, Ada 95, also emphasizes support for parallel programming (see MULTIPROCESSING). The future of Ada is unclear, however, because the Department of Defense no longer requires use of the language in government contracts.

Ada development has continued, particularly in areas including expanded object-oriented features (including support for interfaces with multiple inheritance); improved handling of strings, other data types, and files; and refinements in real-time processing and numeric processing.

Further Reading

"Ada 95 Lovelace Tutorial." Available online. URL: http://www.adahome.com/Tutorials/Lovelace/lovelace.htm. Accessed April 18, 2008.

Ada 95 On-line Reference Manual (hypertext) Available online. URL: http://www.adahome.com/Resources/refs/rm95.html. Accessed April 18, 2008.

Barnes, John. *Programming in Ada 2005 with CD.* New York: Pearson Education, 2006.

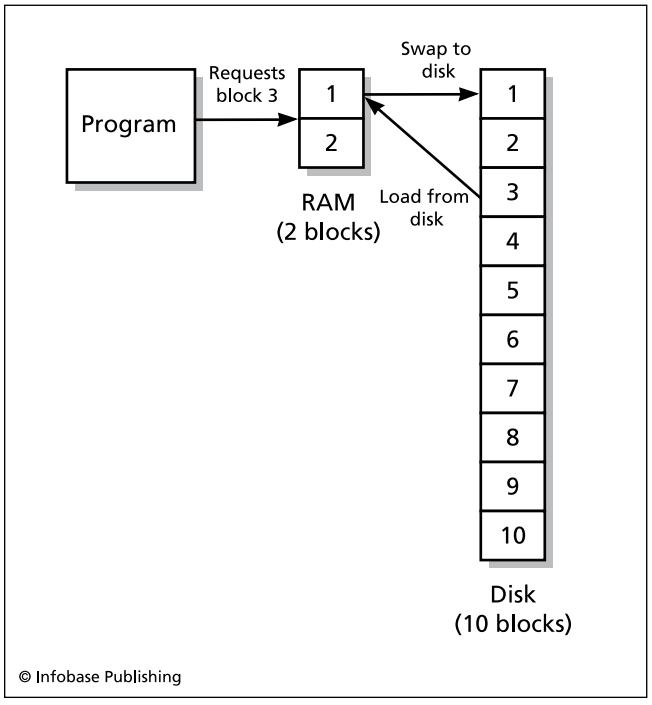
Dale, Nell, and John W. McCormick. *Ada Plus Data Structures: An Object-Oriented Approach.* 2nd ed. Sudbury, Mass.: Jones and Bartlett, 2006.

addressing

In order for computers to manipulate data, they must be able to store and retrieve it on demand. This requires a way to specify the location and extent of a data item in memory. These locations are represented by sequential numbers, or addresses.

Physically, a modern RAM (random access memory) can be visualized as a grid of address lines that crisscross with data lines. Each line carries one bit of the address, and together, they specify a particular location in memory (see MEMORY). Thus a machine with 32 address lines can handle up to 32 bits, or 4 gigabytes (billions of bytes) worth of addresses. However the amount of memory that can be addressed can be extended through indirect addressing, where the data stored at an address is itself the address of another location where the actual data can be found. This allows a limited amount of fast memory to be used to point to data stored in auxiliary memory or mass storage thus extending addressing to the space on a hard disk drive.

Some of the data stored in memory contains the actual program instructions to be executed. As the processor executes program instructions, an instruction pointer accesses the location of the next instruction. An instruction can also specify that if a certain condition is met the processor will jump over intervening locations to fetch the next instruction. This implements such control structures as branching statements and loops.



Virtual memory uses indirect addressing. When a program requests data from memory, the address is looked up in a table that keeps track of each block's actual location. If the block is not in RAM, one or more blocks in RAM are copied to the swap file on disk, and the needed blocks are copied from disk into the vacated area in RAM.

ADDRESSING IN PROGRAMS

A variable name in a program language actually references an address (or often, a range of successive addresses, since most data items require more than one byte of storage). For example, if a program includes the declaration

```
Int Old_Total, New_Total;
```

when the program is compiled, storage for the variables Old_Total and New_Total is set aside at the next available addresses. A statement such as

```
New_Total = 0;
```

is compiled as an instruction to store the value 0 in the address represented by New_Total. When the program later performs a calculation such as:

```
New_Total = Old_Total + 1;
```

the data is retrieved from the memory location designated by Old_Total and stored in a register in the CPU, where 1 is added to it, and the result is stored in the memory location designated by New_Total.

Although programmers don't have to work directly with address locations, programs can also use a special type of variable to hold and manipulate memory addresses for more efficient access to data (see POINTERS AND INDIRECTION).

Further Reading

"Computer Architecture Tutorial." Available online. URL: http://www.cs.iastate.edu/~prabhu/Tutorial/title.html. Accessed April 10, 2007.

Murdocca, Miles J., and Vincent P. Heuring. *Principles of Computer Architecture*. Upper Saddle River, N.J.: Prentice Hall, 2000.

Adobe Systems

Adobe Systems (NASDAQ symbol ADBE) is best known for products relating to the formatting, printing, and display of documents. Founded in 1982 by John Warnock and Charles Geschke, the company is named for a creek near one of their homes.

Adobe's first major product was a language that describes the font sizes, styles, and other formatting needed to print pages in near-typeset quality (see POSTSCRIPT). This was a significant contribution to the development of software for document creation (see DESKTOP PUBLISHING), particularly on the Apple Macintosh, starting in the later 1980s. Building on this foundation, Adobe developed high-quality digital fonts (called Type 1). However, Apple's TrueType fonts proved to be superior in scaling to different sizes and in the precise control over the pixels used to display them. With the licensing of TrueType to Microsoft for use in Windows, TrueType fonts took over the desktop, although Adobe Type 1 remained popular in commercial typesetting applications. Finally, in the late 1990s Adobe, together with Microsoft, established a new font format called OpenType, and by 2003 Adobe had converted all of its Type 1 fonts to the new format.

Adobe's Portable Document Format (see PDF) has become a ubiquitous standard for displaying print documents. Adobe greatly contributed to this development by making a free Adobe Acrobat PDF reader available for download.

Image Processing Software

In the mid-1980s Adobe's founders realized that they could further exploit the knowledge of graphics rendition that they had gained in developing their fonts. They began to create software that would make these capabilities available to illustrators and artists as well as desktop publishers. Their first such product was Adobe Illustrator for the Macintosh, a vector-based drawing program that built upon the graphics capabilities of their PostScript language.

In 1989 Adobe introduced Adobe Photoshop for the Macintosh. With its tremendous variety of features, the program soon became a standard tool for graphic artists. However, Adobe seemed to have difficulty at first in anticipating the growth of desktop publishing and graphic arts on the Microsoft Windows platform. Much of that market was seized by competitors such as Aldus PageMaker and QuarkXPress. By the mid-1990s, however, Adobe, fueled by the continuing revenue from its PostScript technology, had acquired both Aldus and Frame Technologies, maker of the popular FrameMaker document design program. Meanwhile PhotoShop continued to develop on both the Macintosh and Windows platforms, aided by its ability to accept add-ons from hundreds of third-party developers (see PLUG-INS).

Multimedia and the Web

Adobe made a significant expansion beyond document processing into multimedia with its acquisition of Macromedia (with its popular Flash animation software) in 2005 at a cost of about $3.4 billion. The company has integrated Macromedia's Flash and Dreamweaver Web-design software into its Creative Suite 3 (CS3). Another recent Adobe product that targets Web-based publishing is Digital Editions, which integrated the existing Dreamweaver and Flash software into a powerful but easy-to-use tool for delivering text content and multimedia to Web browsers. Buoyed by these developments, Adobe earned nearly $2 billion in revenue in 2005, about $2.5 billion in 2006, and $3.16 billion in 2007.

Today Adobe has over 6,600 employees, with its headquarters in San Jose and offices in Seattle and San Francisco as well as Bangalore, India; Ottawa, Canada; and other locations. In recent years the company has been regarded as a superior place to work, being ranked by *Fortune* magazine as the fifth best in America in 2003 and sixth best in 2004.

Further Reading

"Adobe Advances on Stronger Profit." *Business Week Online,* December 18, 2006. Available online. URL: http://www.businessweek.com/investor/content/dec2006/pi20061215_986588.htm. Accessed April 10, 2007.

Adobe Systems Incorporated home page. Available online. URL: http://www.adobe.com. Accessed April 10, 2007.

"Happy Birthday Acrobat: Adobe's Acrobat Turns 10 Years Old." *Print Media* 18 (July–August 2003): 21.

Advanced Micro Devices (AMD)

Sunnyvale, California-based Advanced Micro Devices, Inc., (NYSE symbol AMD) is a major competitor in the market for integrated circuits, particularly the processors that are at the heart of today's desktop and laptop computers (see MICROPROCESSOR). The company was founded in 1969 by a group of executives who had left Fairchild Semiconductor. In 1975 the company began to produce both RAM (memory) chips and a clone of the Intel 8080 microprocessor.

When IBM adopted the Intel 8080 for its first personal computer in 1982 (see INTEL CORPORATION and IBM PC), it required that there be a second source for the chip. Intel therefore signed an agreement with AMD to allow the latter to manufacture the Intel 9806 and 8088 processors. AMD also produced the 80286, the second generation of PC-compatible processors, but when Intel developed the 80386 it canceled the agreement with AMD.

A lengthy legal dispute ensued, with the California Supreme Court finally siding with AMD in 1991. However, as disputes continued over the use by AMD of "microcode" (internal programming) from Intel chips, AMD eventually used a "clean room" process to independently create functionally equivalent code (see REVERSE ENGINEERING). However, the speed with which new generations of chips was being produced rendered this approach impracticable by the mid-1980s, and Intel and AMD concluded a (largely secret) agreement allowing AMD to use Intel code and providing for cross-licensing of patents.

In the early and mid-1990s AMD had trouble keeping up with Intel's new Pentium line, but the AMD K6 (introduced in 1997) was widely viewed as a superior implementation of the microcode in the Intel Pentium—and it was "pin compatible," making it easy for manufacturers to include it on their motherboards.

Today AMD remains second in market share to Intel. AMD's Athlon, Opteron, Turion, and Sempron processors are comparable to corresponding Intel Pentium processors, and the two companies compete fiercely as each introduces new architectural features to provide greater speed or processing capacity.

In the early 2000s AMD seized the opportunity to beat Intel to market with chips that could double the data bandwidth from 32 bits to 64 bits. The new specification standard, called AMD64, was adopted for upcoming operating systems by Microsoft, Sun Microsystems, and the developers of Linux and UNIX kernels. AMD has also matched Intel in the latest generation of dual-core chips that essentially provide two processors on one chip. Meanwhile, AMD strengthened its position in the high-end server market when, in May 2006, Dell Computer announced that it would market servers containing AMD Opteron processors. In 2006 AMD also moved into the graphics-processing field by merging with ATI, a leading maker of video cards, at a cost of $5.4 billion. Meanwhile AMD also continues to be a leading maker of flash memory, closely collaborating with Japan's Fujitsu Corporation (see FLASH DRIVE). In 2008 AMD continued its aggressive pursuit of market share, announcing a variety of products, including a quad-core Opteron chip that it expects to catch up to if not surpass similar chips from Intel.

Further Reading
AMD Web site. Available online. URL: http://www.amd.com/us-en/. Accessed April 10, 2007.
Rodengen, Jeffrey L. *The Spirit of AMD: Advanced Micro Devices*. Ft. Lauderdale, Fla.: Write Stuff Enterprises, 1998.
Tom's Hardware [CPU articles and charts]. Available online. URL: http://www.tomshardware.com/find_by_topic/cpu.html. Accessed April 10, 2007.

advertising, online

See ONLINE ADVERTISING.

agent software

See SOFTWARE AGENT.

AI

See ARTIFICIAL INTELLIGENCE.

Aiken, Howard

(1900–1973)
American
Electrical Engineer

Howard Hathaway Aiken was a pioneer in the development of automatic calculating machines. Born on March 8, 1900, in Hoboken, New Jersey, he grew up in Indianapolis, Indiana, where he pursued his interest in electrical engineering by working at a utility company while in high school. He earned a B.A. in electrical engineering in 1923 at the University of Wisconsin.

By 1935, Aiken was involved in theoretical work on electrical conduction that required laborious calculation. Inspired by work a hundred years earlier (see BABBAGE, CHARLES), Aiken began to investigate the possibility of building a large-scale, programmable, automatic computing device (see CALCULATOR). As a doctoral student at Harvard, Aiken aroused interest in his project, particularly from Thomas Watson, Sr., head of International Business Machines (IBM). In 1939, IBM agreed to underwrite the building of Aiken's first calculator, the Automatic Sequence Controlled Calculator, which became known as the Harvard Mark I.

MARK I AND ITS PROGENY

Like Babbage, Aiken aimed for a general-purpose programmable machine rather than an assembly of special-purpose arithmetic units. Unlike Babbage, Aiken had access to a variety of tested, reliable components, including card punches, readers, and electric typewriters from IBM and the mechanical electromagnetic relays used for automatic switching in the telephone industry. His machine used decimal numbers (23 digits and a sign) rather than the binary numbers of the majority of later computers. Sixty registers held whatever constant data numbers were needed to solve a particular problem. The operator turned a rotary dial to enter each digit of each number. Variable data and program instructions were entered via punched paper tape. Calculations had to be broken down into specific instructions similar to those in later low-level programming languages such as "store this number in this register" or "add this number to the number in that register" (see ASSEMBLER). The results (usually tables of mathematical function values) could be printed by an electric typewriter or output on punched cards. Huge (about 8 feet [2.4 m] high by 51 feet [15.5 m] long), slow, but reliable, the Mark I worked on a variety of problems during World War II, ranging from equations used in lens design and radar to the designing of the implosive core of an atomic bomb.

Aiken completed an improved model, the Mark II, in 1947. The Mark III of 1950 and Mark IV of 1952, however, were electronic rather than electromechanical, replacing relays with vacuum tubes.

Compared to later computers such as the ENIAC and UNIVAC, the sequential calculator, as its name suggests, could only perform operations in the order specified. Any looping had to be done by physically creating a repetitive tape of instructions. (After all, the program as a whole was not stored in any sort of memory, and so previous instructions could not be reaccessed.) Although Aiken's machines soon slipped out of the mainstream of computer development, they did include the modern feature of parallel processing, because different calculation units could work on different instructions at the same time. Further, Aiken recognized the value of maintaining a library of frequently needed routines that could be reused in new programs—another fundamental of modern software engineering.

Aiken's work demonstrated the value of large-scale automatic computation and the use of reliable, available technology. Computer pioneers from around the world came to Aiken's Harvard computation lab to debate many issues that would become staples of the new discipline of computer science. The recipient of many awards including the Edison Medal of the IEEE and the Franklin Institute's John Price Award, Howard Aiken died on March 14, 1973, in St. Louis, Missouri.

Further Reading
Cohen, I. B. *Howard Aiken: Portrait of a Computer Pioneer.* Cambridge, Mass.: MIT Press, 1999.
Cohen, I. B., R. V. D. Campbell, and G. Welch, eds. *Makin' Numbers: Howard Aiken and the Computer.* Cambridge, Mass.: MIT Press, 1999.

Ajax (Asynchronous JavaScript and XML)

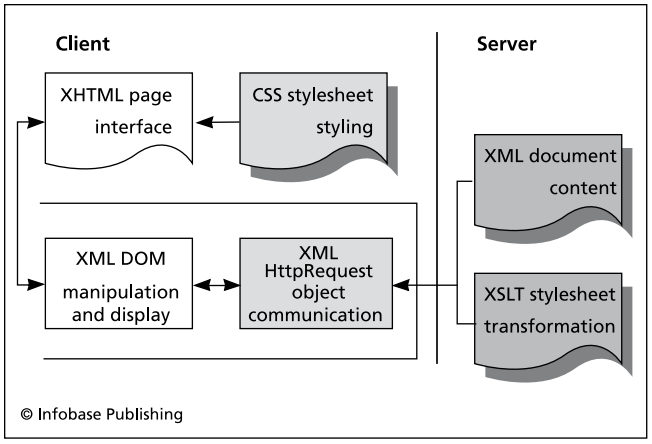
With the tremendous growth in Web usage comes a challenge to deliver Web-page content more efficiently and with greater flexibility. This is desirable to serve adequately the many users who still rely on relatively low-speed dial-up Internet connections and to reduce the demand on Web servers. Ajax (asynchronous JavaScript and XML) takes advantage of several emerging Web-development technologies to allow Web pages to interact with users while keeping the amount of data to be transmitted to a minimum.

In keeping with modern Web-design principles, the organization of the Web page is managed by coding in XHTML, a dialect of HTML that uses the stricter rules and

grammar of the data-description markup language XML (see HTML, DHTML, AND XHTML and XML). Alternatively, data can be stored directly in XML. A structure called the DOM (Document Object Model; see DOM) is used to request data from the server, which is accessed through an object called httpRequest. The "presentational" information (regarding such matters as fonts, font sizes and styles, justification of paragraphs, and so on) is generally incorporated in an associated cascading style sheet (see CASCADING STYLE SHEETS). Behavior such as the presentation and processing of forms or user controls is usually handled by a scripting language (for example, see JAVASCRIPT). Ajax techniques tie these forms of processing together so that only the part of the Web page affected by current user activity needs to be updated. Only a small amount of data needs to be received from the server, while most of the HTML code needed to update the page is generated on the client side—that is, in the Web browser. Besides making Web pages more flexible and interactive, Ajax also makes it much easier to develop more elaborate applications, even delivering fully functional applications such as word processing and spreadsheets over the Web (see APPLICATION SERVICE PROVIDER).

Some critics of Ajax have decried its reliance on JavaScript, arguing that the language has a hard-to-use syntax similar to the C language and poorly implements objects (see OBJECT-ORIENTED PROGRAMMING). There is also a need to standardize behavior across the popular Web browsers. Nevertheless, Ajax has rapidly caught on in the Web development community, filling bookstore shelves with books on applying Ajax techniques to a variety of other languages (*see*, for example, PHP).

Ajax can be simplified by providing a framework of objects and methods that the programmer can use to set up and manage the connections between server and browser. Some frameworks simply provide a set of data structures and functions (see APPLICATION PROGRAM INTERFACE), while others include Ajax-enabled user interface components such as buttons or window tabs. Ajax frameworks also vary in how much of the processing is done on the server and how much is done on the client (browser) side. Ajax frameworks are most commonly used with JavaScript, but also exist for Java (Google Web Toolkit), PHP, C++, and Python as well as other scripting languages. An interesting example is Flapjax, a project developed by researchers at Brown University. Flapjax is a complete high-level programming language that uses the same syntax as the popular JavaScript but hides the messy details of sharing and updating data between client and server.



Ajax is a way to quickly and efficiently update dynamic Web pages—formatting is separate from content, making it easy to revise the latter.

DRAWBACKS AND CHALLENGES

By their very nature, Ajax-delivered pages behave differently from conventional Web pages. Because the updated page is not downloaded as such from the server, the browser cannot record it in its "history" and allow the user to click the "back" button to return to a previous page. Mechanisms for counting the number of page views can also fail. As a workaround, programmers have sometimes created "invisible" pages that are used to make the desired history entries. Another problem is that since content manipulated using Ajax is not stored in discrete pages with identifiable URLs, conventional search engines cannot read and index it, so a copy of the data must be provided on a conventional page for indexing. The extent to which XML should be used in place of more compact data representations is also a concern for many developers. Finally, accessibility tools (see DISABLED PERSONS AND COMPUTERS) often do not work with Ajax-delivered content, so an alternative form must often be provided to comply with accessibility guidelines or regulations.

Despite these concerns, Ajax is in widespread use and can be seen in action in many popular Web sites, including Google Maps and the photo-sharing site Flickr.com.

Further Reading

Ajaxian [news and resources for Ajax developers]. Available online. URL: http://ajaxian.com/. Accessed April 10, 2007.
Crane, David, Eric Pascarello, and Darren James. *Ajax in Action*. Greenwich, Conn.: Manning Publications, 2006.
"Google Web Toolkit: Build AJAX Apps in the Java Language." Available online. URL: http://code.google.com/webtoolkit/. Accessed April 10, 2007.
Holzner, Steve. *Ajax for Dummies*. Hoboken, N.J.: Wiley, 2006.
Jacobs, Sas. *Beginning XML with DOM and Ajax: From Novice to Professional*. Berkeley, Calif.: Apress, 2006.

Algol

The 1950s and early 1960s saw the emergence of two high-level computer languages into widespread use. The first was designed to be an efficient language for performing scientific calculations (see FORTRAN). The second was designed for business applications, with an emphasis on data processing (see COBOL). However many programs continued to be coded in low-level languages (see ASSEMBLER) designed to take advantages of the hardware features of particular machines.

In order to be able to easily express and share methods of calculation (see ALGORITHM), leading programmers

began to seek a "universal" programming language that was not designed for a particular application or hardware platform. By 1957, the German GAMM (Gesellschaft für angewandte Mathematik und Mechanik) and the American ACM (Association for Computing Machinery) had joined forces to develop the specifications for such a language. The result became known as the Zurich Report or Algol-58, and it was refined into the first widespread implementation of the language, Algol-60.

Language Features

Algol is a block-structured, procedural language. Each variable is declared to belong to one of a small number of kinds of data including integer, real number (see DATA TYPES), or a series of values of either type (see ARRAY). While the number of types is limited and there is no facility for defining new types, the compiler's type checking (making sure a data item matches the variable's declared type) introduced a level of security not found in most earlier languages.

An Algol program can contain a number of separate procedures or incorporate externally defined procedures (see LIBRARY, PROGRAM), and the variables with the same name in different procedure blocks do not interfere with one another. A procedure can call itself (see RECURSION). Standard control structures (see BRANCHING STATEMENTS and LOOP) were provided.

The following simple Algol program stores the numbers from 1 to 10 in an array while adding them up, then prints the total:

```
begin
  integer array ints[1:10];
  integer counter, total;
  total := 0;
  for counter :=1 step 1 until counter > 10
  do
    begin
    ints [counter] := counter;
    total := total + ints[counter];
  end;
  printstring "The total is:";
  printint (total);
  end
```

Algol's Legacy

The revision that became known as Algol-68 expanded the variety of data types (including the addition of boolean, or true/false values) and added user-defined types and "structs" (records containing fields of different types of data). Pointers (references to values) were also implemented, and flexibility was added to the parameters that could be passed to and from procedures.

Although Algol was used as a production language in some computer centers (particularly in Europe), its relative complexity and unfamiliarity impeded its acceptance, as did the widespread corporate backing for the rival languages FORTRAN and especially COBOL. Algol achieved its greatest success in two respects: for a time it became the language of choice for describing new algorithms for computer scientists, and its structural features would be adopted in the new procedural languages that emerged in the 1970s (see PASCAL and C).

Further Reading

"Algol 68 Home Page." URL: http://www.algol68.org. Accessed April 10, 2007.

Backus, J. W., and others. "Revised Report on the Algorithmic Language Algol 60." Originally published in *Numerische Mathematik*, the *Communications of the ACM,* and the *Journal of the British Computer Society.* Available online. URL: http://www.masswerk.at/algol60/report.htm. Accessed April 10, 2007.

algorithm

When people think of computers, they usually think of silicon chips and circuit boards. Moving from relays to vacuum tubes to transistors to integrated circuits has vastly increased the power and speed of computers, but the essential idea behind the work computers do remains the algorithm. An algorithm is a reliable, definable procedure for solving a problem. The idea of the algorithm goes back to the beginnings of mathematics and elementary school students are usually taught a variety of algorithms. For example, the procedure for long division by successive division, subtraction, and attaching the next digit is an algorithm. Since a bona fide algorithm is guaranteed to work given the specified type of data and the rote following of a series of steps, the algorithmic approach is naturally suited to mechanical computation.

Algorithms in Computer Science

Just as a cook learns both general techniques such as how to sauté or how to reduce a sauce and a repertoire of specific recipes, a student of computer science learns both general problem-solving principles and the details of common algorithms. These include a variety of algorithms for organizing data (see SORTING AND SEARCHING), for numeric problems (such as generating random numbers or finding primes), and for the manipulation of data structures (see LIST PROCESSING and QUEUE).

A working programmer faced with a new task first tries to think of familiar algorithms that might be applicable to the current problem, perhaps with some adaptation. For example, since a variety of well-tested and well-understood sorting algorithms have been developed, a programmer is likely to apply an existing algorithm to a sorting problem rather than attempt to come up with something entirely new. Indeed, for most widely used programming languages there are packages of modules or procedures that implement commonly needed data structures and algorithms (see LIBRARY, PROGRAM).

If a problem requires the development of a new algorithm, the designer will first attempt to determine whether the problem can, at least in theory, be solved (see COMPUTABILITY AND COMPLEXITY). Some kinds of problems have been shown to have no guaranteed answer. If a new algorithm seems feasible, principles found to be effective in the past will be employed, such as breaking complex problems

down into component parts or building up from the simplest case to generate a solution (see RECURSION). For example, the merge-sort algorithm divides the data to be sorted into successively smaller portions until they are sorted, and then merges the sorted portions back together.

Another important aspect of algorithm design is choosing an appropriate way to organize the data (see DATA STRUCTURES). For example, a sorting algorithm that uses a branching (tree) structure would probably use a data structure that implements the nodes of a tree and the operations for adding, deleting, or moving them (see CLASS).

Once the new algorithm has been outlined (see PSEUDOCODE), it is often desirable to demonstrate that it will work for any suitable data. Mathematical techniques such as the finding and proving of loop invariants (where a true assertion remains true after the loop terminates) can be used to demonstrate the correctness of the implementation of the algorithm.

Practical Considerations

It is not enough that an algorithm be reliable and correct, it must also be accurate and efficient enough for its intended use. A numerical algorithm that accumulates too much error through rounding or truncation of intermediate results may not be accurate enough for a scientific application. An algorithm that works by successive approximation or convergence on an answer may require too many iterations even for today's fast computers, or may consume too much of other computing resources such as memory. On the other hand, as computers become more and more powerful and processors are combined to create more powerful supercomputers (see SUPERCOMPUTER and CONCURRENT PROGRAMMING), algorithms that were previously considered impracticable might be reconsidered. Code profiling (analysis of which program statements are being executed the most frequently) and techniques for creating more efficient code can help in some cases. It is also necessary to keep in mind special cases where an otherwise efficient algorithm becomes much less efficient (for example, a tree sort may work well for random data but will become badly unbalanced and slow when dealing with data that is already sorted or mostly sorted).

Sometimes an exact solution cannot be mathematically guaranteed or would take too much time and resources to calculate, but an approximate solution is acceptable. A so-called "greedy algorithm" can proceed in stages, testing at each stage whether the solution is "good enough." Another approach is to use an algorithm that can produce a reasonable if not optimal solution. For example, if a group of tasks must be apportioned among several people (or computers) so that all tasks are completed in the shortest possible time, the time needed to find an exact solution rises exponentially with the number of workers and tasks. But an algorithm that first sorts the tasks by decreasing length and then distributes them among the workers by "dealing" them one at a time like cards at a bridge table will, as demonstrated by Ron Graham, give an allocation guaranteed to be within 4/3 of the optimal result—quite suitable for most applications. (A procedure that can produce a practical, though not perfect solution is actually not an algorithm but a heuristic.)

An interesting approach to optimizing the solution to a problem is allowing a number of separate programs to "compete," with those showing the best performance surviving and exchanging pieces of code ("genetic material") with other successful programs (see GENETIC ALGORITHMS). This of course mimics evolution by natural selection in the biological world.

Further Reading

Berlinksi, David. *The Advent of the Algorithm: The Idea That Rules the World.* New York: Harcourt, 2000.

Cormen, T. H., C. E. Leiserson, R. L. Rivest, and Clifford Stein. *Introduction to Algorithms.* 2nd ed. Cambridge, Mass.: MIT Press, 2001.

Knuth, Donald E. *The Art of Computer Programming.* Vol. 1: *Fundamental Algorithms.* 3rd ed. Reading, Mass.: Addison-Wesley, 1997. Vol. 2: *Seminumerical Algorithms.* 3rd ed. Reading, Mass.: Addison-Wesley, 1997. Vol. 3: *Searching and Sorting.* 2nd ed. Reading, Mass.: Addison-Wesley, 1998.

ALU

See ARITHMETIC LOGIC UNIT.

Amazon.com

Beginning modestly in 1995 as an online bookstore, Amazon.com became one of the first success stories of the early Internet economy (see also E-COMMERCE).

Named for the world's largest river, Amazon.com was the brainchild of entrepreneur Jeffrey Bezos (see BEZOS, JEFFREY P.). Like a number of other entrepreneurs of the early 1990s, Bezos had been searching for a way to market to the growing number of people who were going online. He soon decided that books were a good first product, since they were popular, nonperishable, relatively compact, and easy to ship.

Several million books are in print at any one time, with about 275,000 titles or editions added in 2007 in the United States alone. Traditional "brick and mortar" (physical) bookstores might carry a few thousand titles up to perhaps 200,000 for the largest chains. Bookstores in turn stock their shelves mainly through major book distributors that serve as intermediaries between publishers and the public.

For an online bookstore such as Amazon.com, however, the number of titles that can be made available is limited only by the amount of warehouse space the store is willing to maintain—and no intermediary between publisher and bookseller is needed. From the start, Amazon.com's business model has capitalized on this potential for variety and the ability to serve almost any niche interest. Over the years the company's offerings have expanded beyond books to 34 different categories of merchandise, including software, music, video, electronics, apparel, home furnishings, and even nonperishable gourmet food and groceries. (Amazon.com also entered the online auction market, but remains a distant runner-up to market leader eBay).

Expansion and Profitability

Because of its desire to build a very diverse product line, Amazon.com, unusually for a business startup, did not expect to become profitable for about five years. The growing revenues were largely poured back into expansion. In the heated atmosphere of the Internet boom of the late 1990s, many other Internet-based businesses echoed that philosophy, and many went out of business following the bursting of the so-called dot-com bubble of the early 2000s. Some analysts questioned whether even the hugely popular Amazon.com would ever be able to convert its business volume into an operating profit. However, the company achieved its first profitable year in 2003 (with a modest $35 million surplus). Since then growth has remained steady and generally impressive: In 2005, Amazon.com earned $8.49 billion revenues with a net income of $359 million. By then the company had about 12,000 employees and had been added to the S&P 500 stock index.

In 2006 the company maintained its strategy of investing in innovation rather than focusing on short-term profits. Its latest initiatives include selling digital versions of books (e-books) and magazine articles, new arrangements to sell video content, and even a venture into moviemaking. By year end, annual revenue had increased to $10.7 billion.

In November 2007 Amazon announced the Kindle, a book reader (see E-BOOKS AND DIGITAL LIBRARIES) with a sharp "paper-like" display. In addition to books, the Kindle can also subscribe to and download magazines, content from newspaper Web sites, and even blogs.

As part of its expansion strategy, Amazon.com has acquired other online bookstore sites including Borders.com and Waldenbooks.com. The company has also expanded geographically with retail operations in Canada, the United Kingdom, France, Germany, Japan, and China.

Amazon.com has kept a tight rein on its operations even while continually expanding. The company's leading market position enables it to get favorable terms from publishers and manufacturers. A high degree of warehouse automation and an efficient procurement system keep stock moving quickly rather than taking up space on the shelves.

Information-Based Strategies

Amazon.com has skillfully taken advantage of information technology to expand its capabilities and offerings. Examples of such efforts include new search mechanisms, cultivation of customer relationships, and the development of new ways for users to sell their own goods.

Amazon's "Search Inside the Book" feature is a good example of leveraging search technology to take advantage of having a growing amount of text online. If the publisher of a book cooperates, its actual text is made available for online searching. (The amount of text that can be displayed is limited to prevent users from being able to read entire books for free.) Further, one can see a list of books citing (or being cited by) the current book, providing yet another way to explore connections between ideas as used by different authors. Obviously for Amazon.com, the ultimate reason for offering all these useful features is that more potential customers may be able to find and purchase books on even the most obscure topics.

Amazon.com's use of information about customers' buying histories is based on the idea that the more one knows about what customers have wanted in the past, the more effectively they can be marketed to in the future through customizing their view of the site. Users receive automatically generated recommendations for books or other items based on their previous purchases (see also CUSTOMER RELATIONSHIP MANAGEMENT). There is even a "plog" or customized Web log that offers postings related to the user's interests and allows the user to respond.

There are other ways in which Amazon.com tries to involve users actively in the marketing process. For example, users are encouraged to review books and other products and to create lists that can be shared with other users. The inclusion of both user and professional reviews in turn makes it easier for prospective purchasers to determine whether a given book or other item is suitable. Authors are given the opportunity through "Amazon Connect" to provide additional information about their books. Finally, in late 2005 Amazon replaced an earlier "discussion board" facility with a wiki system that allows purchasers to create or edit an information page for any product (see WIKIS AND WIKIPEDIA).

The company's third major means of expansion is to facilitate small businesses and even individual users in the marketing of their own goods. Amazon Marketplace, a service launched in 2001, allows users to sell a variety of items, with no fees charged unless the item is sold. There are also many provisions for merchants to set up online "storefronts" and take advantage of online payment and other services.

Another aspect of Amazon's marketing is its referral network. Amazon's "associates" are independent businesses that provide links from their own sites to products on Amazon. For example, a seller of crafts supplies might include on its site links to books on crafting on the Amazon site. In return, the referring business receives a commission from Amazon.com.

Although often admired for its successful business plan, Amazon.com has received criticism from several quarters. Some users have found the company's customer service (which is handled almost entirely by e-mail) to be unresponsive. Meanwhile local and specialized bookstores, already suffering in recent years from the competition of large chains such as Borders and Barnes and Noble, have seen in Amazon.com another potent threat to the survival of their business. (The company's size and economic power have elicited occasional comparisons with Wal-Mart.) Finally, Amazon.com has been criticized by some labor advocates for paying low wages and threatening to terminate workers who sought to unionize.

Further Reading

Amazon.com Web site. Available online. URL: http://www.amazon.com. Accessed August 28, 2007.
Daisey, Mike. *21 Dog Years: Doing Time @ Amazon.com.* New York: The Free Press, 2002.
Marcus, James. *Amazonia.* New York: New Press, 2005.

Shanahan, Francis. *Amazon.com Mashups*. New York: Wrox/Wiley, 2007.
Spector, Robert. *Amazon.com: Get Big Fast: Inside the Revolutionary Business Model That Changed the World*. New York: HarperBusiness, 2002.

Amdahl, Gene Myron

(1922–)
American
Inventor, Entrepreneur

Gene Amdahl played a major role in designing and developing the mainframe computer that dominated data processing through the 1970s (see MAINFRAME). Amdahl was born on November 16, 1922, in Flandreau, South Dakota. After having his education interrupted by World War II, Amdahl received a B.S. from South Dakota State University in 1948 and a Ph.D. in physics at the University of Wisconsin in 1952.

As a graduate student Amdahl had realized that further progress in physics and other sciences required better, faster tools for computing. At the time there were only a few computers, and the best approach to getting access to significant computing power seemed to be to design one's own machine. Amdahl designed a computer called the WISC (Wisconsin Integrally Synchronized Computer). This computer used a sophisticated procedure to break calculations into parts that could be carried out on separate processors, making it one of the earliest examples of the parallel computing techniques found in today's computer architectures.

Designer for IBM

In 1952 Amdahl went to work for IBM, which had committed itself to dominating the new data processing industry. Amdahl worked with the team that eventually designed the IBM 704. The 704 improved upon the 701, the company's first successful mainframe, by adding many new internal programming instructions, including the ability to perform floating point calculations (involving numbers that have decimal points). The machine also included a fast, high-capacity magnetic core memory that let the machine retrieve data more quickly during calculations. In November 1953 Amdahl became the chief project engineer for the 704 and then helped design the IBM 709, which was designed especially for scientific applications.

When IBM proposed extending the technology by building a powerful new scientific computer called STRETCH, Amdahl eagerly applied to head the new project. However, he ended up on the losing side of a corporate power struggle, and did not receive the post. He left IBM at the end of 1955.

In 1960 Amdahl rejoined IBM, where he was soon involved in several design projects. The one with the most lasting importance was the IBM System/360, which would become the most ubiquitous and successful mainframe computer of all time. In this project Amdahl further refined his ideas about making a computer's central processing unit more efficient. He designed logic circuits that enabled the processor to analyze the instructions waiting to be executed (the "pipeline") and determine which instructions could be executed immediately and which would have to wait for the results of other instructions. He also used a cache, or special memory area, in which the instructions that would be needed next could be stored ahead of time so they could be retrieved immediately when needed. Today's desktop PCs use these same ideas to get the most out of their chips' capabilities.

Amdahl also made important contributions to the further development of parallel processing. Amdahl created a formula called Amdahl's law that basically says that the advantage gained from using more processors gradually declines as more processor are added. The amount of improvement is also proportional to how much of the calculation can be broken down into parts that can be run in parallel. As a result, some kinds of programs can run much faster with several processors being used simultaneously, while other programs may show little improvement.

In the mid-1960s Amdahl helped establish IBM's Advanced Computing Systems Laboratory in Menlo Park, California, which he directed. However, he became increasingly frustrated with what he thought was IBM's too rigid approach to designing and marketing computers. He decided to leave IBM again and, this time, challenge it in the marketplace.

Creator of "Clones"

Amdahl resolved to make computers that were more powerful than IBM's machines, but that would be "plug compatible" with them, allowing them to use existing hardware and software. To gain an edge over the computer giant, Amdahl was able to take advantage of the early developments in integrated electronics to put more circuits on a chip without making the chips too small, and thus too crowded for placing the transistors.

Thanks to the use of larger scale circuit integration, Amdahl could sell machines with superior technology to that of the IBM 360 or even the new IBM 370, and at a lower price. IBM responded belatedly to the competition, making more compact and faster processors, but Amdahl met each new IBM product with a faster, cheaper alternative. However, IBM also countered by using a sales technique that opponents called FUD (fear, uncertainty, and doubt). IBM salespersons promised customers that IBM would soon be coming out with much more powerful and economical alternatives to Amdahl's machines. As a result, many would-be customers were persuaded to postpone purchasing decisions and stay with IBM. Amdahl Corporation began to falter, and Gene Amdahl gradually sold his stock and left the company in 1980.

Amdahl then tried to repeat his success by starting a new company called Trilogy. The company promised to build much faster and cheaper computers than those offered by IBM or Amdahl. He believed he could accomplish this by using the new, very-large-scale integrated silicon wafer technology in which circuits were deposited in layers on a single chip rather than being distributed on separate chips on a printed circuit board. But the problem of dealing with the electrical characteristics of such dense circuitry,

as well as some design errors, somewhat crippled the new computer design. Amdahl was forced to repeatedly delay the introduction of the new machine, and Trilogy failed in the marketplace.

Amdahl's achievements could not be overshadowed by the failures of his later career. He has received many industry awards, including Data Processing Man of the Year by the Data Processing Management Association (1976), the Harry Goode Memorial Award from the American Federation of Information Processing Societies, and the SIGDA Pioneering Achievement Award (2007).

Further Reading

"Gene Amdahl." Available online. URL: http://www.thocp.net/biographies/amdahl_gene.htm. Accessed April 10, 2007.
Slater, Robert. *Portraits in Silicon*. Cambridge, Mass.: MIT Press, 1987.

America Online (AOL)

For millions of PC users in the 1990s, "going online" meant connecting to America Online. However, this once dominant service provider has had difficulty adapting to the changing world of the Internet.

By the mid-1980s a growing number of PC users were starting to go online, mainly dialing up small bulletin board services. Generally these were run by individuals from their homes, offering a forum for discussion and a way for users to upload and download games and other free software and shareware (see BULLETIN BOARD SYSTEMS). However, some entrepreneurs saw the possibility of creating a commercial information service that would be interesting and useful enough that users would pay a monthly subscription fee for access. Perhaps the first such enterprise to be successful was Quantum Computer Services, founded by Jim Kimsey in 1985 and soon joined by another young entrepreneur, Steve Case. Their strategy was to team up with personal computer makers such as Commodore, Apple, and IBM to provide special online services for their users.

In 1989 Quantum Link changed its name to America Online (AOL). In 1991 Steve Case became CEO, taking over from the retiring Kimsey. Case's approach to marketing AOL was to aim the service at novice PC users who had trouble mastering arcane DOS (disk operating system) commands and interacting with text-based bulletin boards and primitive terminal programs. As an alternative, AOL provided a complete software package that managed the user's connection, presented "friendly" graphics, and offered point-and-click access to features.

Chat rooms and discussion boards were also expanded and offered in a variety of formats for casual and more formal use. Gaming, too, was a major emphasis of the early AOL, with some of the first online multiplayer fantasy role-playing games such as a version of Dungeons and Dragons called *Neverwinter Nights* (see ONLINE GAMES). A third popular application has been instant messaging (IM), including a feature that allowed users to set up "buddy lists" of their friends and keep track of when they were online (see also TEXTING AND INSTANT MESSAGING).

Internet Challenge

By 1996 the World Wide Web was becoming popular (see WORLD WIDE WEB). Rather than signing up with a proprietary service such as AOL, users could simply get an account with a lower-cost direct-connection service (see INTERNET SERVICE PROVIDER) and then use a Web browser such as Netscape to access information and services. AOL was slow in adapting to the growing use of the Internet. At first, the service provided only limited access to the Web (and only through its proprietary software). Gradually, however, AOL offered a more seamless Web experience, allowing users to run their own browsers and other software together with the proprietary interface. Also, responding to competition, AOL replaced its hourly rates with a flat monthly fee ($19.95 at first).

Overall, AOL increasingly struggled with trying to fulfill two distinct roles: Internet access provider and content provider. By the late 1990s AOL's monthly rates were higher than those of "no frills" access providers such as NetZero. AOL tried to compensate for this by offering integration of services (such as e-mail, chat, and instant messaging) and news and other content not available on the open Internet.

AOL also tried to shore up its user base with aggressive marketing to users who wanted to go online but were not sure how to do so. Especially during the late 1990s, AOL was able to swell its user rolls to nearly 30 million, largely by providing millions of free CDs (such as in magazine inserts) that included a setup program and up to a month of free service. But while it was easy to get started with AOL, some users began to complain that the service would keep billing them even after they had repeatedly attempted to cancel it. Meanwhile, AOL users got little respect from the more sophisticated inhabitants of cyberspace, who often complained that the clueless "newbies" were cluttering newsgroups and chat rooms.

In 2000 AOL and Time Warner merged. At the time, the deal was hailed as one of the greatest mergers in corporate

America Online (AOL) was a major online portal in the 1990s, but has faced challenges adapting to the modern world of the Web. (SCREEN IMAGE CREDIT: AOL)

history, bringing together one of the foremost Internet companies with one of the biggest traditional media companies. The hope was that the new $350 billion company would be able to leverage its huge subscriber base and rich media resources to dominate the online world.

From Service to Content Provider

By the 2000s, however, an increasing number of people were switching from dial-up to high-speed broadband Internet access (see BROADBAND) rather than subscribing to services such as AOL simply to get online. This trend and the overall decline in the Internet economy early in the decade (the "dot-bust") contributed to a record loss of $99 billion for the combined company in 2002. In a shakeup, Time-Warner dropped "AOL" from its name, and Steve Case was replaced as executive chairman. The company increasingly began to shift its focus to providing content and services that would attract people who were already online, with revenue coming from advertising instead of subscriptions.

In October 2006 the AOL division of Time-Warner (which by then had dropped the full name America Online) announced that it would provide a new interface and software optimized for broadband users. AOL's OpenRide desktop presents users with multiple windows for e-mail, instant messaging, Web browsing, and media (video and music), with other free services available as well. These offerings are designed to compete in a marketplace where the company faces stiff competition from other major Internet presences who have been using the advertising-based model for years (see YAHOO! and GOOGLE).

Further Reading

AOL Web site. Available online. URL: http://www.aol.com. Accessed August 28, 2007.

Kaufeld, John. *AOL for Dummies*. 2nd ed. Hoboken, N.J.: Wiley, 2004.

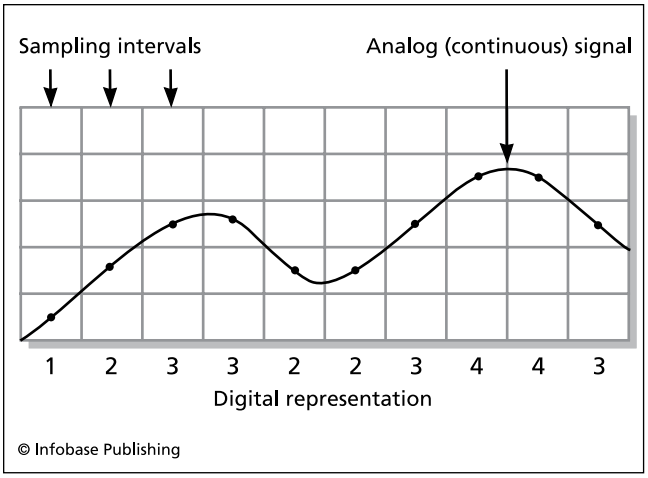
Klein, Alec. *Stealing Time: Steve Case, Jerry Levin, and the Collapse of AOL Time Warner*. New York: Simon & Schuster, 2003.

Mehta, Stephanie N. "Can AOL Keep Pace?" *Fortune,* August 21, 2006, p. 29.

Swisher, Kara. *AOL.COM: How Steve Case Beat Bill Gates, Nailed the Netheads, and Made Millions in the War for the Web*. New York: Times Books, 1998.

analog and digital

The word *analog* (derived from Greek words meaning "by ratio") denotes a phenomenon that is continuously variable, such as a sound wave. The word *digital,* on the other hand, implies a discrete, exactly countable value that can be represented as a series of digits (numbers). Sound recording provides familiar examples of both approaches. Recording a phonograph record involves electromechanically transferring a physical signal (the sound wave) into an "analogous" physical representation (the continuously varying peaks and dips in the record's surface). Recording a CD, on the other hand, involves sampling (measuring) the sound level at thousands of discrete instances and storing the results in a physical representation of a numeric format that can in turn be used to drive the playback device.



Most natural phenomena such as light or sound intensity are analog values that vary continuously. To convert such measurements to a digital representation, "snapshots" or sample readings must be taken at regular intervals. Sampling more frequently gives a more accurate representation of the original analog data, but at a cost in memory and processor resources.

Virtually all modern computers depend on the manipulation of discrete signals in one of two states denoted by the numbers 1 and 0. Whether the 1 indicates the presence of an electrical charge, a voltage level, a magnetic state, a pulse of light, or some other phenomenon, at a given point there is either "something" (1) or "nothing" (0). This is the most natural way to represent a series of such states.

Digital representation has several advantages over analog. Since computer circuits based on binary logic can be driven to perform calculations electronically at ever-increasing speeds, even problems where an analog computer better modeled nature can now be done more efficiently with digital machines (see ANALOG COMPUTER). Data stored in digitized form is not subject to the gradual wear or distortion of the medium that plagues analog representations such as the phonograph record. Perhaps most important, because digital representations are at base simply numbers, an infinite variety of digital representations can be stored in files and manipulated, regardless of whether they started as pictures, music, or text (see DIGITAL CONVERGENCE).

Converting between Analog and Digital Representations

Because digital devices (particularly computers) are the mechanism of choice for working with representations of text, graphics, and sound, a variety of devices are used to digitize analog inputs so the data can be stored and manipulated. Conceptually, each digitizing device can be thought of as having three parts: a component that scans the input and generates an analog signal, a circuit that converts the analog signal from the input to a digital format, and a component that stores the resulting digital data for later use. For example, in the ubiquitous flatbed scanner a moving head reads varying light levels on the paper and converts them to

a varying level of current (see SCANNER). This analog signal is in turn converted into a digital reading by an analog-to-digital converter, which creates numeric information that represents discrete spots (pixels) representing either levels of gray or of particular colors. This information is then written to disk using the formats supported by the operating system and the software that will manipulate them.

Further Reading

Chalmers, David J. "Analog vs. Digital Computation." Available online. URL: http://www.u.arizona.edu/~chalmers/notes/analog.html. Accessed April 10, 2007.

Hoeschele, David F. *Analog-to-Digital and Digital-to-Analog Conversion Techniques.* 2nd ed. New York: Wiley-Interscience, 1994.

analog computer

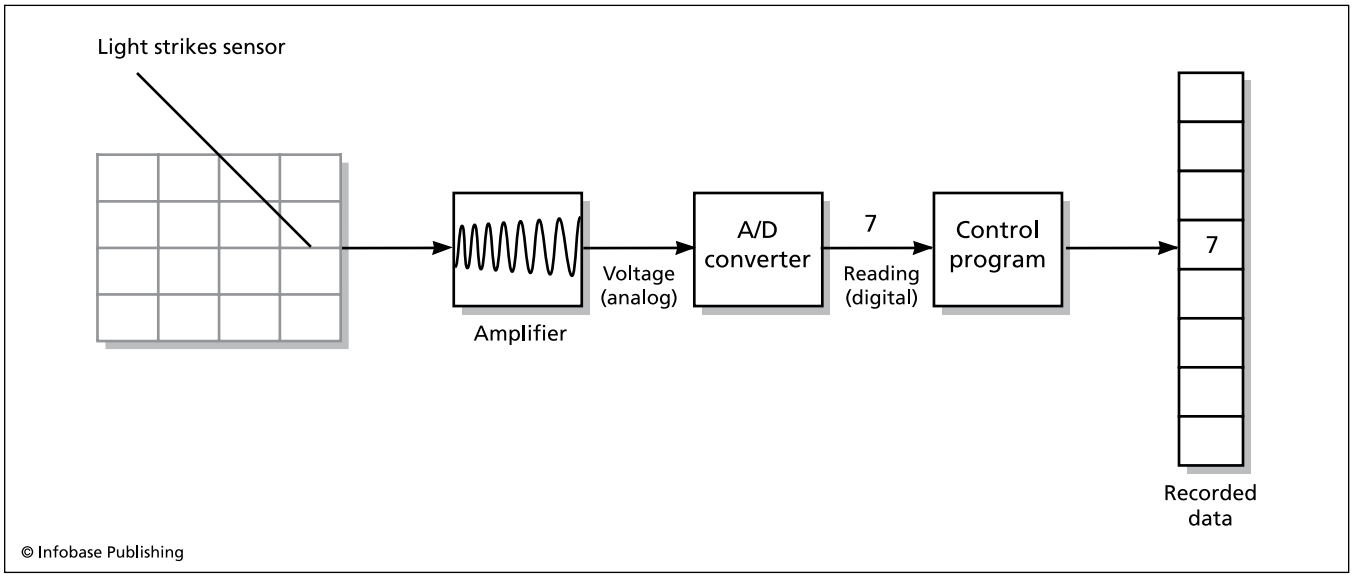
Most natural phenomena are analog rather than digital in nature (see ANALOG AND DIGITAL). But just as mathematical laws can describe relationships in nature, these relationships in turn can be used to construct a model in which natural forces generate mathematical solutions. This is the key insight that leads to the analog computer.

The simplest analog computers use physical components that model geometric ratios. The earliest known analog computing device is the Antikythera Mechanism. Constructed by an unknown scientist on the island of Rhodes around 87 B.C., this device used a precisely crafted differential gear mechanism to mechanically calculate the interval between new moons (the synodic month). (Interestingly, the differential gear would not be rediscovered until 1877.)

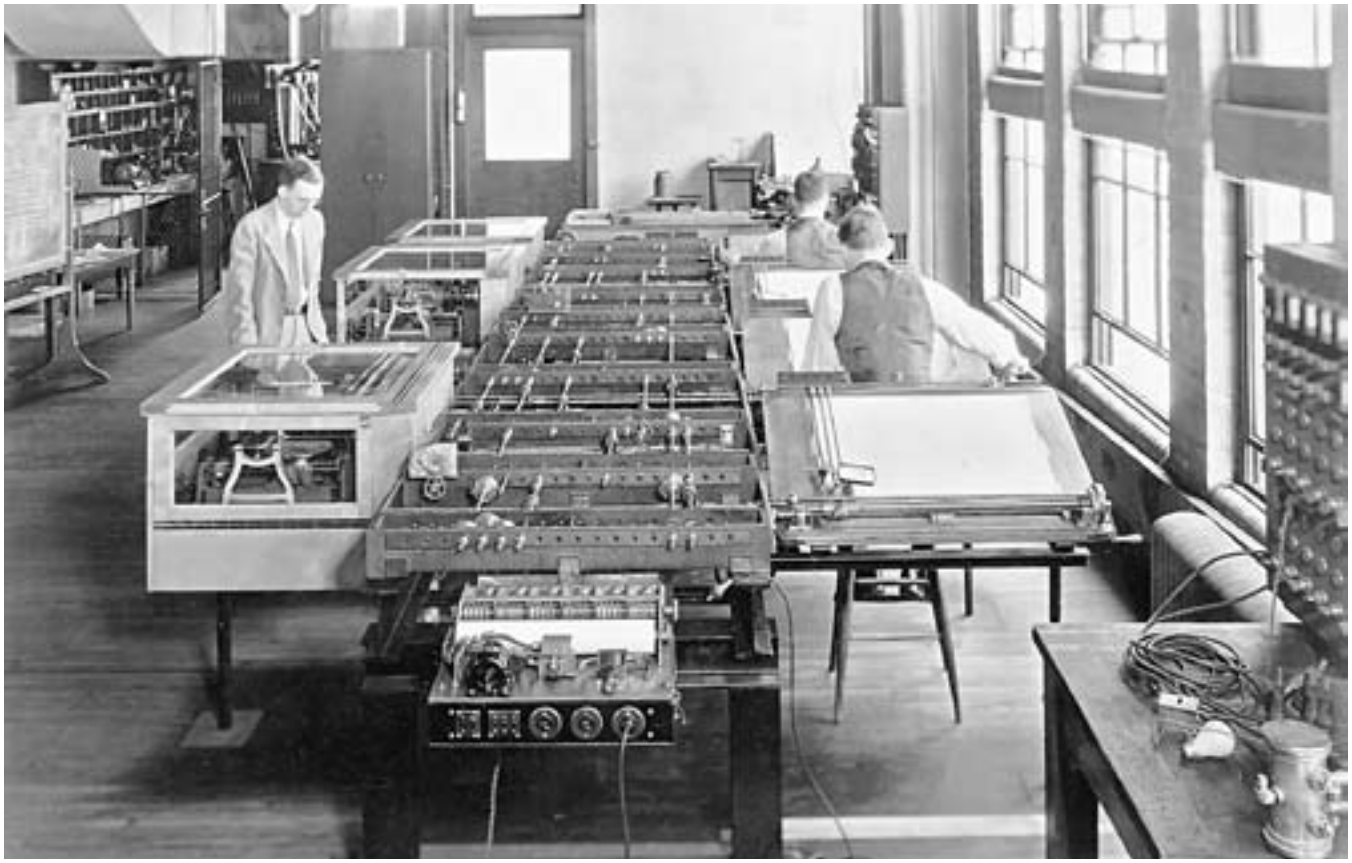
Another analog computer, the slide rule, became the constant companion of scientists, engineers, and students until it was replaced by electronic calculators in the 1970s. Invented in simple form in the 17th century, the slide rule's movable parts are marked in logarithmic proportions, allowing for quick multiplication, division, the extraction of square roots, and sometimes the calculation of trigonometric functions.

The next insight involved building analog devices that set up dynamic relationships between mechanical movements. In the late 19th century two British scientists, James Thomson and his brother Sir William Thomson (later Lord Kelvin) developed the mechanical integrator, a device that could solve differential equations. An important new principle used in this device is the closed feedback loop, where the output of the integrator is fed back as a new set of inputs. This allowed for the gradual summation or integration of an equation's variables. In 1931, VANNEVAR BUSH completed a more complex machine that he called a "differential analyzer." Consisting of six mechanical integrators using specially shaped wheels, disks, and servomechanisms, the differential analyzer could solve equations in up to six independent variables. As the usefulness and applicability of the device became known, it was quickly replicated in various forms in scientific, engineering, and military institutions.

These early forms of analog computer are based on fixed geometrical ratios. However, most phenomena that scientists and engineers are concerned with, such as aerodynamics, fluid dynamics, or the flow of electrons in a circuit, involve a mathematical relationship between forces where the output changes smoothly as the inputs are changed. The "dynamic" analog computer of the mid-20th century took advantage of such force relationships to construct devices where input forces represent variables in the equation, and



Converting analog data to digital involves several steps. A sensor (such as the CCD, or charge-coupled device in a digital camera) creates a varying electrical current. An amplifier can strengthen this signal to make it easier to process, and filters can eliminate spurious spikes or "noise." The "conditioned" signal is then fed to the analog-to-digital (A/D) converter, which produces numeric data that is usually stored in a memory buffer from which it can be processed and stored by the controlling program.



Completed in 1931, Vannevar Bush's Differential Analyzer was a triumph of analog computing. The device could solve equations with up to six independent values. (MIT MUSEUM)

nature itself "solves" the equation by producing a resulting output force.

In the 1930s, the growing use of electronic circuits encouraged the use of the flow of electrons rather than mechanical force as a source for analog computation. The key circuit is called an operational amplifier. It generates a highly amplified output signal of opposite polarity to the input, over a wide range of frequencies. By using components such as potentiometers and feedback capacitors, an analog computer can be programmed to set up a circuit in which the laws of electronics manipulate the input voltages in the same way the equation to be solved manipulates its variables. The results of the calculation are then read as a series of voltage values in the final output.

Starting in the 1950s, a number of companies marketed large electronic analog computers that contained many separate computing units that could be harnessed together to provide "real time" calculations in which the results could be generated at the same rate as the actual phenomena being simulated. In the early 1960s, NASA set up training simulations for astronauts using analog real-time simulations that were still beyond the capability of digital computers.

Gradually, however, the use of faster processors and larger amounts of memory enabled the digital computer to surpass its analog counterpart even in the scientific programming and simulations arena. In the 1970s, some hybrid machines combined the easy programmability of a digital "front end" with analog computation, but by the end of that decade the digital computer had rendered analog computers obsolete.

Further Reading

"Analog Computers." Computer Museum, University of Amsterdam. Available online. URL: http://www.science.uva.n/museum/AnalogComputers.html. Accessed April 18, 2007.

Hoeschele, David F., Jr. *Analog-to-Digital and Digital-to-Analog Conversion Techniques*. 2nd ed. New York: John Wiley, 1994.

Vassos, Basil H., and Galen Ewing, eds. *Analog and Computer Electronics for Scientists*. 4th ed. New York: John Wiley, 1993.

Andreessen, Marc

(1971–)
American
Entrepreneur, Programmer

Marc Andreessen brought the World Wide Web and its wealth of information, graphics, and services to the desktop, setting the stage for the first "e-commerce" revolution of the later 1990s. As founder of Netscape, Andreessen also

created the first big "dot-com," or company doing business on the Internet.

Born on July 9, 1971, in New Lisbon, Wisconsin, Andreessen grew up as part of a generation that would become familiar with personal computers, computer games, and graphics. By seventh grade Andreessen had his own PC and was programming furiously. He then studied computer science at the University of Illinois at Urbana-Champaign, where his focus on computing was complemented by a wide-ranging interest in music, history, literature, and business.

By the early 1990s the World Wide Web (see WORLD WIDE WEB and BERNERS-LEE, TIM) was poised to change the way information and services were delivered to users. However, early Web pages generally consisted only of linked pages of text, without point-and-click navigation or the graphics and interactive features that adorn Web pages today.

Andreessen learned about the World Wide Web shortly after Berners-Lee introduced it in 1991. Andreessen thought it had great potential, but also believed that there needed to be better ways for ordinary people to access the new medium. In 1993, Andreessen, together with colleague Eric Bina and other helpers at the National Center for Supercomputing Applications (NCSA), set to work on what became known as the Mosaic Web browser. Since their work was paid for by the government, Mosaic was offered free to users over the Internet. Mosaic could show pictures as well as text, and users could follow Web links simply by clicking on them with the mouse. The user-friendly program became immensely popular, with more than 10 million users by 1995.

Marc Andreessen, Chairman of Loudcloud, Inc., speaks at Fortune magazine's "Leadership in Turbulent Times" conference on November 8, 2001, in New York City. (PHOTO BY MARIO TAMA/GETTY IMAGES)

After earning a B.S. in computer science, Andreessen left Mosaic, having battled with its managers over the future of Web-browsing software. He then met Jim Clark, an older entrepreneur who had been CEO of Silicon Graphics. They founded Netscape Corporation in 1994, using $4 million seed capital provided by Clark.

Andreessen recruited many of his former colleagues at NCSA to help him write a new Web browser, which became known as Netscape Navigator. Navigator was faster and more graphically attractive than Mosaic. Most important, Netscape added a secure encrypted facility that people could use to send their credit card numbers to online merchants. This was part of a two-pronged strategy: First, attract the lion's share of Web users to the new browser, and then sell businesses the software they would need to create effective Web pages for selling products and services to users.

By the end of 1994 Navigator had gained 70 percent of the Web browser market. *Time* magazine named the browser one of the 10 best products of the year, and Netscape was soon selling custom software to companies that wanted a presence on the Web. The e-commerce boom of the later 1990s had begun, and Marc Andreessen was one of its brightest stars. When Netscape offered its stock to the public in summer 1995, the company gained a total worth of $2.3 billion, more than that of many traditional blue-chip industrial companies. Andreessen's own shares were worth $55 million.

BATTLE WITH MICROSOFT

Microsoft (see MICROSOFT and GATES, BILL) had been slow to recognize the growing importance of the Web, but by the mid-1990s Gates had decided that the software giant had to have a comprehensive "Internet strategy." In particular, the company had to win control of the browser market so users would not turn to "platform independent" software that could deliver not only information but applications, without requiring the use of Windows at all.

Microsoft responded by creating its own Web browser, called Internet Explorer. Although technical reviewers generally considered the Microsoft product to be inferior to Netscape, it gradually improved. Most significantly, Microsoft included Explorer with its new Windows 95 operating system. This "bundling" meant that PC makers and consumers had little interest in paying for Navigator when they already had a "free" browser from Microsoft. In response to this move, Netscape and other Microsoft competitors helped promote the antitrust case against Microsoft that would result in 2001 in some of the company's practices being declared an unlawful use of monopoly power.

Andreessen tried to respond to Microsoft by focusing on the added value of his software for Web servers while making Navigator "open source," meaning that anyone was allowed to access and modify the program's code (see OPEN SOURCE). He hoped that a vigorous community of programmers might help keep Navigator technically superior to Internet Explorer. However, Netscape's revenues began to decline steadily. In 1999 America Online (AOL) bought the company, seeking to add its technical assets and Webcenter online portal to its own offerings (see AMERICA ONLINE).

After a brief stint with AOL as its "principal technical visionary," Andreessen decided to start his own company, called LoudCloud. The company provided Web-site development, management, and custom software (including e-commerce "shopping basket" systems) for corporations that had large, complex Web sites. However, the company was not successful; Andreessen sold its Web-site-management component to Texas-based Electronic Data Systems (EDS) while retaining its software division under the new name Opsware. In 2007 Andreessen scored another coup, selling Opsware to Hewlett-Packard (HP) for $1.6 billion.

In 2007 Andreessen launched Ning, a company that offers users the ability to add blogs, discussion forums, and other features to their Web sites, but facing established competitors such as MySpace (see also SOCIAL NETWORKING). In July 2008 Andresseen joined the board of Facebook.

While the future of his recent ventures remains uncertain, Marc Andreessen's place as one of the key pioneers of the Web and e-commerce revolution is assured. His inventiveness, technical insight, and business acumen made him a model for a new generation of Internet entrepreneurs. Andreessen was named one of the Top 50 People under the Age of 40 by *Time* magazine (1994) and has received the Computerworld/Smithsonian Award for Leadership (1995) and the W. Wallace McDowell Award of the IEEE Computer Society (1997).

Further Reading

Clark, Jim. *Netscape Time: The Making of the Billion-Dollar Startup That Took on Microsoft*. New York: St. Martin's Press, 1999.

Guynn, Jessica. "Andreessen Betting Name on New Ning." *San Francisco Chronicle,* February 27, 2006, p. D1, D4.

Payment, Simone. *Marc Andreessen and Jim Clark: The Founders of Netscape*. New York: Rosen Pub. Group, 2006.

Quittner, Joshua, and Michelle Slatala. *Speeding the Net: The Inside Story of Netscape and How It Challenged Microsoft*. New York: Atlantic Monthly Press, 1998.

animation, computer

Ever since the first hand-drawn cartoon features entertained moviegoers in the 1930s, animation has been an important part of the popular culture. Traditional animation uses a series of hand-drawn frames that, when shown in rapid succession, create the illusion of lifelike movement.

Computer Animation Techniques

The simplest form of computer animation (illustrated in games such as *Pong*) involves drawing an object, then erasing it and redrawing it in a different location. A somewhat more sophisticated approach can create motion in a scene by displaying a series of pre-drawn images called *sprites*—for example, there could be a series of sprites showing a sword-wielding troll in different positions.

Since there are only a few intermediate images, the use of sprites doesn't convey truly lifelike motion. Modern animation uses a modern version of the traditional drawn animation technique. The drawings are "keyframes" that capture significant movements by the characters. The keyframes are later filled in with transitional frames in a process called *tweening*. Since it is possible to create algorithms that describe the optimal in-between frames, the advent of sufficiently powerful computers has made computer animation both possible and desirable. Today computer animation is used not only for cartoons but also for video games and movies. The most striking use of this technique is morphing, where the creation of plausible intermediate images between two strikingly different faces creates the illusion of one face being transformed into the other.

Algorithms that can realistically animate people, animals, and other complex objects require the ability to create a model that includes the parts of the object that can move separately (such as a person's arms and legs). Because the movement of one part of the model often affects the positions of other parts, a treelike structure is often used to describe these relationships. (For example, an elbow moves an arm, the arm in turn moves the hand, which in turn moves the fingers). Alternatively, live actors performing a repertoire of actions or poses can be digitized using wearable sensors and then combined to portray situations, such as in a video game.

Less complex objects (such as clouds or rainfall) can be treated in a simpler way, as a collection of "particles" that move together following basic laws of motion and gravity. Of course when different models come into contact (for example, a person walking in the rain), the interaction between the two must also be taken into consideration.

While realism is always desirable, there is inevitably a tradeoff between the resources available. Computationally intensive physics models might portray a very realistic spray of water using a high-end graphics workstation, but simplified models have to be used for a program that runs on a game console or desktop PC. The key variables are the frame rate (higher is smoother) and the display resolution. The amount of available video memory is also a consideration: many desktop PCs sold today have 256MB or more of video memory.

Applications

Computer animation is used extensively in many feature films, such as for creating realistic dinosaurs (*Jurassic Park*) or buglike aliens (*Starship Troopers*). Computer games combine animation techniques with other techniques (see COMPUTER GRAPHICS) to provide smooth action within a vivid 3D landscape. Simpler forms of animation are now a staple of Web site design, often written in Java or with the aid of animation scripting programs such as Adobe Flash.

The intensive effort that goes into contemporary computer animation suggests that the ability to fascinate the human eye that allowed Walt Disney to build an empire is just as compelling today.

Further Reading

"3-D Animation Workshop." Available online. URL: http://www.webreference.com/3d/indexa.html. Accessed April 12, 2007.
Comet, Michael B. "Character Animation: Principles and Practice." Available online. URL: http://www.comet-cartoons.com/toons/3ddocs/charanim. Accessed April 12, 2007.
Hamlin, J. Scott. *Effective Web Animation: Advanced Techniques for the Web*. Reading, Mass.: Addison-Wesley, 1999.
O'Rourke, Michael. *Principles of Three-Dimensional Computer Animation: Modeling, Rendering, and Animating with 3D Computer Graphics*. New York: Norton, 1998.
Parent, Rick. *Computer Animation: Algorithms and Techniques*. San Francisco: Morgan Kaufmann, 2002.
Shupe, Richard, and Robert Hoekman. *Flash 8: Projects for Learning Animation and Interactivity*. Sebastapol, Calif.: O'Reilly Media, 2006.

anonymity and the Internet

Anonymity, or the ability to communicate without disclosing a verifiable identity, is a consequence of the way most Internet-based e-mail, chat, or news services were designed (see E-MAIL, CHAT, TEXTING AND INSTANT MESSAGING, and NETNEWS AND NEWGROUPS). This does not mean that messages do not have names attached. Rather, the names can be arbitrarily chosen or pseudonymous, whether reflecting development of an online persona *or* the desire to avoid having to take responsibility for unwanted communications (see SPAM).

Advantages

If a person uses a fixed Internet address (see TCP/IP), it may be possible to eventually discover the person's location and even identity. However, messages can be sent through anonymous remailing services where the originating address is removed. Web browsing can also be done "at arm's length" through a proxy server. Such means of anonymity can arguably serve important values, such as allowing persons living under repressive governments (or who belong to minority groups) to express themselves more freely precisely because they cannot be identified. However, such techniques require some sophistication on the part of the user. With ordinary users using their service provider accounts directly, governments (notably China) have simply demanded that the user's identity be turned over when a crime is alleged.

Pseudonymity (the ability to choose names separate from one's primary identity) in such venues as chat rooms or online games can also allow people to experiment with different identities or roles, perhaps getting a taste of how members of a different gender or ethnic group are perceived (see IDENTITY IN THE ONLINE WORLD).

Anonymity can also help protect privacy, especially in commercial transactions. For example, purchasing something with cash normally requires no disclosure of the purchaser's identity, address, or other personal information. Various systems can use secure encryption to create a cash equivalent in the online world that assures the merchant of valid payment without disclosing unnecessary information about the purchaser (see DIGITAL CASH). There are also facilities that allow for essentially anonymous Web browsing, preventing the aggregation or tracking of information (see COOKIES).

Problems

The principal problem with anonymity is that it can allow the user to engage in socially undesirable or even criminal activity with less fear of being held accountable. The combination of anonymity (or the use of a pseudonym) and the lack of physical presence seems to embolden some people to engage in insult or "flaming," where they might be inhibited in an ordinary social setting. A few services (notably The WELL) insist that the real identity of all participants be available even if postings use a pseudonym.

Spam or deceptive e-mail (see PHISHING AND SPOOFING) takes advantage both of anonymity (making it hard for authorities to trace) and pseudonymity (the ability to disguise the site by mimicking a legitimate business). Anonymity makes downloading or sharing files easier (see FILE-SHARING AND P2P NETWORKS), but also makes it harder for owners of videos, music, or other content to pursue copyright violations. Because of the prevalence of fraud and other criminal activity on the Internet, there have been calls to restrict the ability of online users to remain anonymous, and some nations such as South Korea have enacted legislation to that effect. However, civil libertarians and privacy advocates believe that the impact on freedom and privacy outweighs any benefits for security and law enforcement.

The database of Web-site registrants (called Whois) provides contact information intended to ensure that someone will be responsible for a given site and be willing to cooperate to fix technical or administrative problems. At present, Whois information is publicly available. However, the Internet Corporation for Assigned Names and Numbers (ICANN) is considering making the contact information available only to persons who can show a legitimate need.

Further Reading

Lessig, Lawrence. *Code: Version 2.0*. New York: Basic Books, 2006.
Rogers, Michael. "Let's See Some ID, Please: The End of Anonymity on the Internet?" *The Practical Futurist* (MSNBC), December 13, 2005. Available online. URL: http://www.msnbc.msn.com/ID/10441443/. Accessed April 10, 2007.
Wallace, Jonathan D. "Nameless in Cyberspace: Anonymity on the Internet." CATO Institute Briefing Papers, no. 54, December 8, 1999. Available online. URL: http://www.cato.org/pubs/briefs/bp54.pdf. Accessed April 10, 2007.

AOL

See AMERICA ONLINE.

API

See APPLICATIONS PROGRAM INTERFACE.