SOFTWARE - AN OVERVIEW

Table of Contents

Digital

Digital describes electronic technology that generates, stores, and processes data in terms of two states: positive and non-positive. Positive is expressed or represented by the number 1 and non-positive by the number 0. Thus, data transmitted or stored with digital technology is expressed as a string of 0's and 1's. Each of these state digits is referred to as a bit (and a string of bits that a computer can address individually as a group is a byte).

Prior to digital technology, electronic transmission was limited to analog technology, which conveys data as electronic signals of varying frequency or amplitude that are added to carrier waves of a given frequency. Broadcast and phone transmission has conventionally used analog technology.

Digital technology is primarily used with new physical communications media, such as satellite and fiber optic transmission. A modem is used to convert the digital information in your computer to analog signals for your phone line and to convert analog phone signals to digital information for your computer.

Binary

Binary is the base two number system that computers use to represent data. It consists of only two numbers: "0" and "1".

Bit (binary digit)

A bit is the smallest unit of data in a computer. A bit has a single binary value, either 0 or 1. Although computers usually provide instructions that can test and manipulate bits, they generally are designed to store data and execute instructions in bit multiples called bytes. In most computer systems, there are eight bits in a byte. The value of a bit is usually stored as either above or below a designated level of electrical charge in a single capacitor within a memory device.

Half a byte (four bits) is called a nibble. In some systems, the term octet is used for an eight-bit unit instead of byte. In many systems, four eight-bit bytes or octets form a 32-bit word. In such systems, instruction lengths are sometimes expressed as full-word (32 bits in length) or half-word (16 bits in length).

Kilobit

In data communications, a kilobit is a thousand bits, or 1,000 (103) bits. It's commonly used for measuring the amount of data that is transferred in a second between two telecommunication points. Kilobits per second is usually shortened to Kbps.*

Some sources define a kilobit to mean 1,024 (that is, 210) bits. Although the bit is a unit of the binary number system, bits in data communications are discrete signal pulses and have historically been counted using the decimal number system. For example, 28.8 kilobits per second (Kbps) is 28,800 bits per second. Because of computer architecture and memory address boundaries, bytes are always some multiple or exponent of two.

Nibble

In computers and digital technology, a nibble (pronounced NIHB-uhl) is four bits or half of an eight-bit byte. A nibble can be conveniently represented by one hexadecimal digit.

According to Microsoft's dictionary, a nibble is sometimes spelled "nybble."

In communications, a nibble is sometimes referred to as a "quadbit." or one of 16 possible four-bit combinations. A signal may be encoded in quadbits rather than one bit at a time. According to Harry Newton, nibble interleaving or multiplexing takes a quadbit or nibble from a lower-speed channel as input for a multiplexed signal on a higher-speed channel.

Byte

In most computer systems, a byte is a unit of information that is eight bits long. A byte is the unit most computers use to represent a character such as a letter, number, or typographic symbol (for example, "g", "5", or "?"). A byte can also hold a string of bits that need to be used in some larger unit for application purposes (for example, the stream of bits that constitute a visual image for a program that displays images).

In some computer systems, four bytes constitute a word, a unit that a computer processor can be designed to handle efficiently as it reads and processes each instruction. Some computer processors can handle two-byte or single-byte instructions.

A byte is abbreviated with a "B". (A bit is abbreviated with a small "b".) Computer storage is usually measured in byte multiples (for example, an 820 MB hard drive holds a nominal 820 million bytes (megabytes) of information. (The number is actually somewhat larger since byte multiples are calculated in powers of 2 and we express them as decimal numbers .)

A 28.8 Kbps modem is one that operates at 28.8 thousand bits (kilobits) per second. (Storage is measured in bytes; transmission capacity in bits per second.)

Some language scripts require two bytes to represent a character. These are called double-byte character sets (DBCS).

Word

In computers, a word is a contiguous string of bits that can be manipulated in specified sections based on how the computer is designed. A word is usually some multiple number of bytes or eight-bit units of data. Thus, a 32-bit word contains four bytes.

A doubleword contains two words. A word is called a "fullword" to distinguish it from a "halfword."

In IBM's OS/370 architecture, a word was four bytes or 32 bits in length. It was the basic unit of data that the processor handled as an instruction. Some instructions could be halfwords or 16 bits in length. Instructions and storage addresses needed to be specified on word-related boundaries.

Kilobyte (KB or Kbyte)

As a measure of computer memory or storage, a kilobyte (KB or Kbyte*) is approximately a thousand bytes (actually, 2 to the 10th power, or decimal 1,024 bytes).

Megabyte (MB)

1) As a measure of computer processor storage and real and virtual memory, a megabyte (abbreviated MB) is 2 to the 20th power bytes, or 1,048,576 bytes in decimal notation.

2) According to the IBM Dictionary of Computing, when used to describe disk storage capacity and transmission rates, a megabyte is 1,000,000 bytes in decimal notation.

According to the Microsoft Press Computer Dictionary, a megabyte means either 1,000,000 bytes or 1,048,576 bytes.

According to Eric S. Raymond in The New Hacker's Dictionary, a megabyte is always 1,048,576 bytes on the argument that bytes should naturally be computed in powers of two.

Iomega Corporation uses the decimal megabyte in calling the Zip drive disk a "100MB disk" when it actually holds 100,431,872 bytes. If Iomega used the powers-of-two megabyte, the disk could be said to hold only 95.8 megabytes (if you divide 100,431,872 by 1,048,576).

Gigabyte (GB)

A gigabyte (pronounced GIG-a-bite with hard G's) is a measure of computer data storage capacity and is "roughly" a billion bytes. A gigabyte is two to the 30th power, or 1,073,741,824 in decimal notation.

Terabyte

A terabyte is a measure of computer storage capacity and is 2 to the 40th power or, in decimal, approximately a thousand billion bytes (that is, a thousand gigabytes).

Petabyte

A petabyte is a measure of memory or storage capacity and is 2 to the 50th power bytes or, in decimal, approximately a thousand terabytes.

In recently announcing how many Fibre Channel storage arrays they had sold, Sun Microsystems stated that it had shipped an aggregate of two petabytes of storage or the equivalent of 40 million four-drawer filing cabinets full of text. IBM says that it has shipped four petabytes of SSA Storage.

Exabyte (EB)

An exabyte (EB) is a large unit of computer data storage, two to the sixtieth power bytes. The prefix exa means one billion billion, or one quintillion, which is a decimal term. Two to the sixtieth power is actually 1,152,921,504,606,846,976 bytes in decimal, or somewhat over a quintillion (or ten to the eighteenth power) bytes. It is common to say that an exabyte is approximately one quintillion bytes. In decimal terms, an exabyte is a billion gigabytes.

ASCII (American Standard Code for Information Interchange)

ASCII is the most common format for text files in computers and on the Internet. In an ASCII file, each alphabetic, numeric, or special character is represented with a 7-bit binary number (a string of seven 0s or 1s). 128 possible characters are defined.

UNIX and DOS-based operating systems (except for Windows NT) use ASCII for text files. Windows NT uses a newer code, Unicode. IBM's System 390 servers use a proprietary 8-bit code called EBCDIC. Conversion programs allow different operating systems to change a file from one code to another.

ASCII was developed by the American National Standards Institute (ANSI).

EBCDIC (extended binary-coded decimal interchange code)

EBCDIC (pronounced either "ehb-suh-dik" or "ehb-kuh-dik") is a binary code for alphabetic and numeric characters that IBM developed for its larger operating systems. It is the code for text files that is used in IBM's OS/390 operating system for its S/390 servers and that thousands of corporations use for their legacy applications and databases. In an EBCDIC file, each alphabetic or numeric character is represented with an 8-bit binary number (a string of eight 0's or 1's). 256 possible characters (letters of the alphabet, numerals, and special characters) are defined.

IBM's PC and workstation operating systems do not use IBM's proprietary EBCDIC. Instead, they use the industry standard code for text, ASCII. Conversion programs allow different operating systems to change a file from one code to another.

Unicode

(which is used by Windows NT and it is a latest technique/standard)

Unicode is an entirely new idea in setting up binary codes for text or script characters. Officially called the Unicode Worldwide Character Standard, it is a system for "the interchange, processing, and display of the written texts of the diverse languages of the modern world." It also supports many classical and historical texts in a number of languages.

Currently, the Unicode standard contains 34,168 distinct coded characters derived from 24 supported language scripts. These characters cover the principal written languages of the world.

Additional work is underway to add the few modern languages not yet included.

The currently most prevalent script or text codes are, ASCII, EBCDIC and. The Unicode.

Language

Generations in Programming Language

In the computer industry, these abbreviations are widely used to represent major steps or "generations" in the evolution of programming languages.

1GL or first-generation language was (and still is) machine language or the level of instructions and data that the processor is actually given to work on (which in conventional computers is a string of 0s and 1s).

2GL or second-generation language is assembler (sometimes called "assembly") language. A typical 2GL instruction looks like this:

```
ADD    12,8
```

An assembler converts the assembler language statements into machine language.

3GL or third-generation language is a "high-level" programming language, such as PL/I, C, or Java. Java language statements look like this:

```
public boolean handleEvent (Event evt) {

        switch (evt.id)  {

            case Event.ACTION_EVENT:  {

                  if ("Try me" .equald(evt.arg)) {
```

A compiler converts the statements of a specific high-level programming language into machine language. (In the case of Java, the output is called bytecode, which is converted into appropriate machine language by a Java virtual machine that runs as part of an operating system platform.) A 3GL language requires a considerable amount of programming knowledge.

4GL or fourth-generation language is designed to be closer to natural language than a 3GL language. Languages for accessing databases are often described as 4GLs. A 4GL language statement might look like this:

```
EXTRACT ALL CUSTOMERS WHERE "PREVIOUS PURCHASES" TOTAL MORE THAN $1000
```

5GL or fifth-generation language is programming that uses a visual or graphical development interface to create source language that is usually compiled with a 3GL or 4GL language compiler. Microsoft, Borland, IBM, and other companies make 5GL visual programming products for developing applications in Java, for example. Visual programming allows you to easily envision object-oriented class hierarchies and drag icons to assemble program components. Microbrew AppWare and IBM's VisualAge for Java are examples of 5GL "languages."

Ada (first computer programmer)

Ada (pronounced AY-duh) is a programming language somewhat similar to Pascal that was selected in a competition and made a U.S. Defense Department standard. (It is named for Augusta Ada Byron, Countess of Lovelace (1815-1852), who helped Charles Babbage conceive how programs might run in his mechanical Analytical Engine. She is often considered the first computer programmer.) Ada was originally intended for real-time embedded systems.

By its supporters, Ada is described as a programming language that avoids error-prone notation, is relatively quick to implement, encourages reuse and team coordination, and is relatively easy for other programmers to read. The most recent version, Ada 95, is apparently a significant improvement over earlier versions. Among hackers, according to The New Hacker's Dictionary, Ada has a reputation as a committee-written language, with poor exception-handling and interprocess communication features. It's not clear that "hackers" still feel this way. The Ada home page says: "The original Ada design was the winner of a language design competition; the winning team was headed by Jean Ichbiah (Ichbiah's language was called "Green"). The 1995 revision of Ada (Ada 95) was developed by a small team led by Tucker Taft. In both cases, the design underwent a public comment period where the designers responded to public comments."

Ada 95 can be used with object-oriented design methodology and source code can be compiled into Java classes by the Ada 95 compiler. These classes can be run as Java applets or applications on a Java virtual machine.

Operating System (OS)

An operating system (sometimes abbreviated as "OS") is the program that, after being initially loaded into the computer by a bootstrap program, manages all the other programs in a computer. The other programs are called applications. The applications make use of the operating system by making requests for services through a defined application program interface
(API). In addition, users can interact directly with the operating system through an interface such as a command language.

An operating system performs these services for applications:

- In multitasking operating systems where multiple programs can be running at the same time, the operating system determines which applications should run in what order and how much time should be allowed for each application before giving another application a turn.
- It manages the sharing of internal memory among multiple applications.
- It handles input and output to and from attached hardware devices, such as hard disks, printers, and dial-up ports.
- It sends messages to the applications or interactive user (or to a system operator) about the status of operation and any errors that may have occurred.
- It can offload the management of what are called batch jobs (for example, printing) so that the initiating application is freed from this work.
- On computers that can provide parallel processing, an operating system can manage how to divide the program so that it runs on more than one processor at a time.

All major computer platforms (hardware and software) require and sometimes include an operating system., DOS, Windows 95, UNIX, DEC's VMS, IBM's OS/2, AIX, and OS/390 are all examples of operating systems.

Batch file

A batch file is a text file that contains a sequence of commands for a computer operating system. It's called a batch file because it batches (bundles or packages) into a single file a set of commands that would otherwise have to be presented to the system interactively from a keyboard one at a time. A batch file is usually created for command sequences for which a user has a repeated need. Commonly needed batch files are often delivered as part of an operating system. You initiate the sequence of commands in the batch file by simply entering the name of the batch file on a command line.

In the DOS operating system, a batch file has the file name extension ".BAT". (The best known DOS batch file is the AUTOEXEC.BAT file that initializes DOS when you start the system.) In UNIX-based operating systems, a batch file is called a shell script. In IBM's mainframe VM operating systems, it's called an EXEC.

Example for batch file of our normal life

1. Get up,
2. Wash your face,
3. Take a cup of Tea,
4. Take bath,
5. Don’t forget to dress-up before coming from the bathroom,
6. Take breakfast,
7. If you are an unemployed, try to do some useful task (and skip #8 - #11)
8. If you are an employer Ready to go for work,
9. Go to work
10. During office hours take some snacks and refreshment,
11. Comeback home with the grace of God,
12. Take lunch,
13. Go to sleep.

GUI (graphical user interface)

A GUI (usually pronounced GOO-ee) is a graphical (rather than purely textual) user interface to a computer. As you read this, you are looking at the GUI or graphical user interface of your particular Web browser. The term came into existence because the first interactive user interfaces to computers were not graphical; they were text-and-keyboard oriented and usually consisted of commands you had to remember and computer responses that were infamously brief. The command interface of the DOS operating system (which you can still get to from your Windows operating system) is an example of the typical user-computer interface before GUIs arrived. An intermediate step in user interfaces between the command line interface and the GUI was the non-graphical menu-based interface, which let you interact by using a mouse rather than by having to type in keyboard commands.

Today's major operating systems provide a graphical user interface. Applications typically use the elements of the GUI that come with the operating system and add their own graphical user interface elements and ideas. A GUI sometimes uses one or more metaphors for objects familiar in real life, such as the desktop, the view through a window, or the physical layout in a building. Elements of a GUI include such things as: windows, pull-down menus, buttons, scroll bars, iconic images, wizards, the mouse, and no doubt many things that haven't been invented yet. With the increasing use of multimedia as part of the GUI, sound, voice, motion video, and virtual reality interfaces seem likely to become part of the GUI for many applications. A system's graphical user interface along with its input devices is sometimes referred to as its "look-and-feel."

The GUI familiar to most of us today in either the Mac or the Windows operating systems and their applications originated at the Xerox Palo Alto Research Laboratory in the late 1970s. Apple used it in their first Macintosh computers. Later, Microsoft used many of the same ideas in their first version of the Windows operating system for IBM-compatible PCs.

When creating an application, many object-oriented tools exist that facilitate writing a graphical user interface. Each GUI element is defined as a class from which you can create object instances for your application. You can code or modify prepackaged methods that an object will use to respond to user stimuli.

Application

1) In information technology, an application is the use of a technology, system, or product.

2) The term application is a shorter form of application program. An application program is a program designed to perform a specific function directly for the user or, in some cases, for another application program. Examples of applications include word processors, database programs, Web browsers, development tools, drawing, paint, and image editing programs, and communication programs. Applications use the services of the computer's operating system and other supporting applications. The formal requests and means of communicating with other programs that an application program uses is called the Application Program Interface (API).

Word processor (Application)

A word processor is a computer program that provides special capabilities beyond that of a text editor and usually provides a graphical user interface. The term originated to distinguish editors that were "easy to use" from conventional text editors and to suggest that the program was more than just an "editor." An early user of this term was Wang, which made a popular workstation system designed especially for secretaries and anyone else who created business letters and other documents.

The most popular word processors are WordPerfect, now owned by Corel, and Microsoft Word.

In general, word processors screen the user from structural or printer-formatting markup (although WordPerfect and other word processors optionally let you see the markup they insert in your text). Without visible markup, it's possible to describe a word processor as having a WYSIWYG (what you see is what you get) user interface.

CAD

CAD (computer-aided design) software is used by architects, engineers, drafters, artists, and others to create precision drawings or technical illustrations. CAD software can be used to create two-dimensional (2-D) drawings or three-dimensional (3-D) models.

CAD/CAM (computer-aided design/computer-aided manufacturing) is software used to design products such as electronic circuit boards in computers and other devices.

Data

(1) In computing, data is information that has been translated into a form that is more convenient to move or process. Relative to today's computers and transmission media, data is information converted into binary or digital form.

2) In computer component interconnection and network communication, data is often distinguished from "control information," "control bits," and similar terms to identify the main content of a transmission unit.

3) In telecommunications, data sometimes means digitally-encoded information to distinguish it from analog-encoded information such as conventional telephone voice calls. In general, "analog" or voice transmission requires a dedicated continual connection for the duration of a related series of transmissions. Data transmission can often be sent with intermittent connections in packets that arrive in piecemeal fashion.

4) Generally and in science, data is a gathered body of facts.

Some authorities and publishers, cognizant of the word's Latin origin and as the plural form of "datum," use plural verb forms with "data". Others take the view that since "datum" is rarely used, it is more natural to treat "data" as a singular form.

Database

A database is a collection of data that is organized so that its contents can easily be accessed, managed, and updated. The most prevalent type of database is the relational database, a tabular database in which data is defined so that it can be reorganized and accessed in a number of different ways. A distributed database is one that can be dispersed or replicated among different points in a network. An object-oriented database is one that is congruent with the data defined in object classes and subclasses.

Databases contain aggregations of data records or files, such as sales transactions, product catalogs and inventories, and customer profiles. Typically, a database manager provides users the capabilities of controlling read/write access, specifying report generation, and analyzing usage. Databases and database managers are prevalent in large mainframe systems, but are also present in smaller distributed workstation and mid-range systems such as the AS/400 and on personal computers. SQL is a standard language for making interactive queries from and updating a database such as IBM's DB2, Microsoft's Access, and database products from Oracle, Sybase, and Computer Associates.

Relational Database

A relational database is a collection of data items organized as a set of formally-described tables from which data can be accessed or reassembled in many different ways without having to reorganize the database tables. The relational database was invented by E. F. Codd at IBM in 1970.

The standard user and application progam interface to a relational database is the structured query language (SQL). SQL statements are used both for interactive queries for information from a relational database and for gathering data for reports.

In addition to being relatively easy to create and access, a relational database has the important advantage of being easy to extend. After the original database creation, a new data category can be added without requiring that all existing applications be modified.

A relational database is a set of tables containing data fitted into predefined categories. Each table (which is sometimes called a relation) contains one or more data categories in columns. Each row contains a unique instance of data for the categories defined by the columns. For example, a typical business order entry database would include a table that described a customer with columns for name, address, phone number, and so forth. Another table would describe an order: product, customer, date, sales price, and so forth. A user of the database could obtain a view of the database that fitted the user's needs. For example, a branch office manager might like a view or report on all customers that had bought products after a certain date. A financial services manager in the same company could, from the same tables, obtain a report on accounts that needed to be paid.

When creating a relational database, you can define the domain of possible values in a data column and further constraints that may apply to that data value. For example, a domain of possible customers could allow up to ten possible customer names but be constrained in one table to allowing only three of these customer names to be specifiable.

The definition of a relational database results in a table of metadata or formal descriptions of the tables, columns, domains, and constraints.

Program

In computing, a program is a specific set of ordered operations for a computer to perform. In the modern computer that John von Neumann outlined in 1945, the program contains a one-at-a-time sequence of instructions that the computer follows. Typically, the program is put into a storage area accessible to the computer. The computer gets one instruction and performs it and then gets the next instruction. The storage area or memory can also contain the data that the instruction operates on. (Note that a program is also a special kind of "data" that tells how to operate on "application or user data.")

Programs can be characterized as interactive or batch in terms of what drives them and how continuously they run. An interactive program receives data from an interactive user (or possibly from another program that simulates an interactive user). A batch program runs and does its work, and then stops. Batch programs can be started by interactive users who request their interactive program to run the batch program. A command interpreter or a Web browser is an example of an interactive program. A program that computes and prints out a company payroll is an example of a batch program. Print jobs are also batch programs.

When you create a program, you write it using some kind of computer language. Your language statements are the source program. You then "compile" the source program (with a special program called a language compiler) and the result is called an object program (not to be confused with object-oriented programming). There are several synonyms for object program, including object module and compiled program. The object program contains the string of 0s and 1s called machine language that the logic processor works with.

The machine language of the computer is constructed by the language compiler with an understanding of the computer's logic architecture, including the set of possible computer instructions and the length (number of bits) in an instruction.

Mnemonic

1) In general, a mnemonic (from Greek mnemon or mindful; pronounced neh-MAHN-ik) is a word, abbreviation, rhyme, or similar verbal device you learn or create in order to remember something. The technique of developing these remembering devices is called "mnemonics." Mnemonics is used to remember phone numbers, all your new department colleagues' names, or the years of the reigns of the Kings and Queens of England. A number of approaches are used. Here's a simple one for remembering a list of unrelated items in order: Start at the top of the list and make up an outlandish story connecting the first item to the next, continue by connecting the second item to the third, and so on. When your story is done and the list is removed, you'll have a mental picture of a story that, as you recall its progression, will lead you from one remembered item to the next.

2) In computer assembler (or assembly) language, a mnemonic is an abbreviation for an operation. It's entered in the operation code field of each assembler program instruction. For example, on an Intel microprocessor, inc ("increase by one") is a mnemonic. On an IBM System/370 series computer, BAL is a mnemonic for "branch-and-link."

Pseudocode

Pseudocode (pronounced SOO-doh-kohd) is a detailed yet readable description of what a computer program or algorithm must do, expressed in a formally-styled natural language rather than in a programming language. Pseudocode is sometimes used as a detailed step in the process of developing a program. It allows designers or lead programmers to express the design in great detail and provides programmers a detailed template for the next step of writing code in a specific programming language.

Because pseudocode is detailed yet readable, it can be inspected by the team of designers and programmers as a way to ensure that actual programming is likely to match design specifications. Catching errors at the pseudocode stage is less costly than catching them later in the development process. Once the pseudocode is accepted, it is rewritten using the vocabulary and syntax of a programming language. Pseudocode is sometimes used in conjunction with CASE-based methodologies.

It is possible to write programs that will convert a given pseudocode language into a given programming language.

Algorithm

The term algorithm (pronounced "AL-go-rith-um") is a procedure or formula for solving a problem. The word derives from the name of the Arab mathematician, Al-Khowarizmi (825 AD). A computer program can be viewed as an elaborate algorithm. In mathematics and computer science, an algorithm usually means a small procedure that solves a recurrent problem.

Syntax

Syntax is the grammar, structure, or order of the elements in a language statement. (Semantics is the meaning of these elements.) Syntax applies to computer languages as well as to natural languages. Usually, we think of syntax as "word order." However, syntax is also achieved in some languages such as Latin by inflectional case endings. In computer languages, syntax can be extremely rigid as in the case of most assembler languages or less rigid in languages that make use of "keyword" parameters that can be stated in any order.

C.W. Morris in his Foundations of the Theory of Signs (1938) organizes semiotics, the study of signs, into three areas: syntax (the study of the interrelation of the signs); semantics (the study of the relation between the signs and the objects to which they apply); and pragmatics (the relationship between the sign system and the user).

Flowchart

A flowchart is a formalized graphic representation of a program logic sequence, work or manufacturing process, organization chart, or similar formalized structure. In computer programming, flowcharts were formerly used to describe each processing path in a program (the main program and various subroutines that could be branched to). Programmers were admonished to always flowchart their logic rather than carry it in their heads. With the advent of object-oriented programming (OOP) and visual development tools, the traditional program flowchart is much less frequently seen. However, there are new flowcharts that can be used for the data or class modeling that is used in object-oriented programming.

Traditional program flowcharting involves the use of simple geometric symbols to represent a process (a rectangle), a decision (a diamond), or an I/O process (a symbol looking something like the home plate in baseball). These symbols are defined in ANSI x 3.5 and ISO 1028.

DBMS (database management system)

A DBMS (database management system), sometimes just called a database manager, is a program that lets one or more computer users create and access data in a database. The DBMS manages user requests (and requests from other programs) so that users and other programs are free from having to understand where the data is physically located on storage media and, in a multi-user system, who else may also be accessing the data. In handling user requests, the DBMS ensures the integrity of the data (that is, making sure it continues to be accessible and is consistently organized as intended) and security (making sure only those with access privileges can access the data). The most typical DBMS is a relational database management system (RDBMS). A standard user and program interface is the Structured Query Language (SQL). A newer kind of DBMS is the object-oriented database management system (OODBMS).

A DBMS can be thought of as a file manager that manages data in databases rather than files in file systems. In IBM's mainframe operating systems, the nonrelational data managers were (and are, because these legacy systems are still used) known as access methods.

A DBMS is usually an inherent part of a database product. On PCs, Microsoft Access is a popular example of a single- or small-group user DBMS. Microsoft's SQL Server is an example of a DBMS that serves database requests from multiple (client) users. Other popular DBMSs (these are all RDBMSs, by the way) are IBM's DB2 and Oracle's line of database management products.

IBM's Information Management System (IMS) was one of the first DBMSs. A DBMS may be used by or combined with transaction managers, such as IBM's Customer Information Control System (CICS).

RDBMS (relational database management system)

An RDBMS is a program that lets you create, update, and administer a relational database. An RDBMS takes Structured Query Language (SQL) statements entered by a user or contained in an application program and creates, updates, or provides access to the database. Some of the best-known RDBMS's include Microsoft's Access, Oracle's Oracle7, and Computer Associates' CA-OpenIngres.

The majority of new corporate, small business, and personal databases are being created for use with an RDBMS. However, a new database model based on object-orientation, ODBMS, is beginning to contend with the RDBMS as the database management system of the future.

The first commercial RDBMS was the Multics Relational Data Store, first sold in 1978.

ODBC (Open Database Connectivity)

Open Database Connectivity (ODBC) is a standard or open application programming interface (API) for accessing a database. By using ODBC statements in a program, you can access files in a number of different databases, including Access, dBase, Excel, and Text. In addition to the ODBC software, a separate module or driver is needed for each database to be accessed. The main proponent and supplier of ODBC programming support is Microsoft.

ODBC is based on and closely aligned with the Open Group standard Structured Query Language (SQL) Call-Level Interface. It allows programs to use SQL requests that will access databases without having to know the proprietary interfaces to the databases. ODBC handles the SQL request and converts it into a request the individual database system understands.

Open Database Connectivity (ODBC)

The Microsoft Open Database Connectivity (ODBC) interface is an industry standard and a component of Microsoft® Windows® Open Services Architecture (WOSA). The ODBC interface makes it possible for applications to access data from a variety of database management systems (DBMSs). ODBC permits maximum interoperability—an application can access data in diverse DBMSs through a single interface. Furthermore, that application will be independent of any DBMS from which it accesses data. Users of the application can add software components called drivers, which create an interface between an application and a specific DBMS.

UDA

Short for Universal Data Access, a high-level specification developed by Microsoft for accessing data objects regardless of their structure. One of the main components of UDA is the ActiveX Data Objects (ADO) interface.

UDA_Universal Data Access

Universal Data Access is the Microsoft strategy for providing access to information across the enterprise. Universal Data Access provides high-performance access to a variety of relational and nonrelational information sources, and an easy-to-use programming interface that is tool and language independent. These technologies enable you to integrate diverse data sources, create easy-to-maintain solutions, and use your choice of best-of-breed tools, applications, and platform services.

Universal Data Access does not require expensive and time-consuming movement of data into a single data store, nor does it require commitment to a single vendor’s products. Universal Data Access is based on open industry specifications with broad industry support, and works with all major established database platforms.

The Microsoft Universal Data Access Web site (www.microsoft.com/data/) provides a central location for you to learn about Universal Data Access and the technologies that make it possible. Here you will find information and the latest news about these technologies.

Microsoft Data Access Components Overview

The Microsoft® Data Access Components (MDAC) are the key technologies that enable Universal Data Access. Data-driven client/server applications deployed over the Web or a LAN can use these components to easily integrate information from a variety of sources, both relational (SQL) and nonrelational. These components include Microsoft® ActiveX® Data Objects (ADO), OLE DB, and Open Database Connectivity (ODBC).

Universal Data Access (UDA)
section of the Data Access SDK

View the latest Microsoft Data Access Components documentation in the Data Access SDK (DASDK) section of the MSDN library by following this link to the Welcome Page, and then clicking the "Show TOC" button at the top of the Welcome page.

Microsoft Data Access SDK Overview

The Microsoft® Data Access SDK is the primary source of information and instruction on using data access technologies. Its tools, samples, and documentation are designed to help developers create solutions for their data access needs. For the latest news and updates about the data access technologies, go to the Microsoft Data Access Web site http://www.microsoft.com/data.

For information on support for Microsoft Data Access Components, see the Getting Help from Microsoft Technical Support page.

ActiveX Data Objects (ADO)

Microsoft ActiveX Data Objects (ADO) is the strategic application programming interface (API) to data and information. ADO provides consistent, high-performance access to data and supports a variety of development needs, including the creation of front-end database clients and middle-tier business objects that use applications, tools, languages, or Internet browsers. ADO is designed to be the one data interface needed for single and multitier client/server and Web-based data-driven solution development. The primary benefits of ADO are ease of use, high speed, low memory overhead, and a small disk footprint.

ADO provides an easy-to-use interface to OLE DB, which provides the underlying access to data. ADO is implemented with minimal network traffic in key scenarios, and a minimal number of layers between the front end and data store—all to provide a lightweight, high-performance interface. ADO is easy to use because it uses a familiar metaphor—the COM automation interface, available from all leading Rapid Application Development (RAD) tools, database tools, and languages on the market today.

OLE DB

OLE DB is the Microsoft strategic system-level programming interface to data across the organization. OLE DB is an open specification designed to build on the success of ODBC by providing an open standard for accessing all kinds of data. Whereas ODBC was created to access relational databases, OLE DB is designed for relational and nonrelational information sources, including mainframe ISAM/VSAM and hierarchical databases; e-mail and file system stores; text, graphical, and geographical data; custom business objects; and more.

OLE DB defines a collection of COM interfaces that encapsulate various database management system services. These interfaces enable the creation of software components that implement such services. OLE DB components consist of data providers, which contain and expose data; data consumers, which use data; and service components, which process and transport data (such as query processors and cursor engines). OLE DB interfaces are designed to help components integrate smoothly so that OLE DB component vendors can bring high-quality OLE DB components to market quickly. In addition, OLE DB includes a bridge to ODBC to enable continued support for the broad range of ODBC relational database drivers available today.

API (Application Program Interface)

An API (application program interface) is the specific method prescribed by a computer operating system or by another application program by which a programmer writing an application program can make requests of the operating system or another application.

An API can be contrasted with a graphical user interface or a command interface (both of which are direct user interfaces) as interfaces to an operating system or a program.

Data Dictionary

A data dictionary is a collection of descriptions of the data objects or items in a data model for the benefit of programmers and others who might need to refer to them. A first step in analyzing a system of objects with which users interact is to identify each object and its relationship to other objects. This process is called data modeling and results in a picture of object relationships. After each data object or item is given a descriptive name, its relationship is described (or it becomes part of some structure that implicitly describes relationship), the type of data (such as text or image or binary value) is described, possible predefined values are listed, and a brief textual description is provided. This collection can be organized for reference into a book called a data dictionary.

When developing programs that use the data model, a data dictionary can be consulted to understand where a data item fits in the structure, what values it may contain, and basically what the data item means in real-world terms. For example, a bank or group of banks could model the data objects involved in consumer banking. They could then provide a data dictionary for a bank's programmers. The data dictionary would describe each of the data items in its data model for consumer banking (for example, "Account holder" and ""Available credit").

Bug

In computer technology, a bug is a coding error in a computer program. (Here we consider a program to also include the microcode that is manufactured into processors.) The process of finding bugs before program users do is called debugging. Debugging starts after the code is first written and continues in successive stages as code is combined with other units of programming to form a software product, such as an operating system or an application program. After a product is released or during public beta testing, bugs are still apt to be discovered. When this occurs, users have to either find a way to avoid using the "buggy" code or get a patch from the originators of the code.

A bug is not the only kind of problem a program can have. It can run bug-free and still be difficult to use or fail in some major objective. This kind of flaw is more difficult to test for (and often simply isn't). It is generally agreed that a well-designed program developed using a well-controlled process will result in fewer bugs per thousands of lines of code.

The term's origin has been wrongly attributed to the pioneer programmer, Grace Hopper. In 1944, Hopper, a young Naval Reserve officer, went to work on the Mark I computer at Harvard, becoming one of the first people to write programs for it. As Admiral Hopper, she later described an incident in which a technician is said to have pulled an actual bug (a moth, in fact) from between two electrical relays in the Mark II computer. In his book, The New Hacker's Dictionary, Eric Raymond reports that the moth was displayed for many years by the Navy and is now the property of the Smithsonian. Raymond also notes that Admiral Hopper was already aware of the term when she told the moth story. Bug was used prior to modern computers to mean an industrial or electrical defect.

Less frequently, the term is applied to a computer hardware problem.

Debugging

In computers, debugging is the process of locating and fixing or bypassing bugs (errors) in computer program code or the engineering of a hardware device. To debug a program or hardware device is to start with a problem, isolate the source of the problem, and then fix it. A user of a program that does not know how to fix the problem may learn enough about the problem to be able to avoid it until it is permanently fixed. When someone says they've debugged a program or "worked the bugs out" of a program, they imply that they fixed it so that the bugs no longer exist.

Debugging is a necessary process in almost any new software or hardware development process, whether a commercial product or an enterprise or personal application program. For complex products, debugging is done as the result of the unit test for the smallest unit of a system, again at component test when parts are brought together, again at system test when the product is used with other existing products, and again during customer beta testing, when users try the product out in a real world situation. Because most computer programs and many programmed hardware devices contain thousands of lines of code, almost any new product is likely to contain a few bugs. Invariably, the bugs in the functions that get most use are found and fixed first. An early version of a program that has lots of bugs is referred to as "buggy."

Debugging tools help identify coding errors at various development stages. Some programming language packages include a facility for checking the code for errors as it is being written.

Data modeling

Data modeling is the analysis of data objects that are used in a business or other context and the identification of the relationships among these data objects. Data modeling is a first step in designing an object-oriented program. As a result of data modeling, you can then define the classes that provide the templates for program objects.

A simple approach to creating a data model that allows you to visualize the model is to draw a square (or any other symbol) to represent each individual data item that you know about (for example, a product or a product price) and then to express relationships between each of these data items with words such as "is part of" or "is used by" or "uses" and so forth. From such a total description, you can create a set of classes and subclasses that define all the general relationships. These then become the templates for objects that, when executed as a program, handle the variables of new transactions and other activities in a way that effectively represents the real world.

Several differing approaches or methodologies to data modeling and its notation have recently been combined into the Unified Modeling Language (UML), which is expected to become a standard modeling language.

UML (Unified Modeling Language)

UML (Unified Modeling Language) is a standard notation for the modeling of real-world objects as a first step in developing an object-oriented program. Its notation is derived from and unifies the notations of three object-oriented design and analysis methodologies:

Grady Booch's methodology for describing a set of objects and their
relationships (see The Booch Method)
James Rumbaugh's Object-Modeling Technique (OMT)
Ivar Jacobson's approach which includes a use case methodology

Other ideas also contributed to UML, which was the result of a work effort by Booch, Rumbaugh, Jacobson, and others to combine their ideas, working under the sponsorship of Rational Software. UML has been fostered and now is an accepted standard of the Object Management Group (OMG), which is also the home of CORBA, the leading industry standard for distributed object programming. Vendors of CASE products are now supporting UML and it has been endorsed by almost every maker of software development products , including IBM and Microsoft (for its Visual Basic environment).

Martin Fowler, in his book UML Distilled, observes that, although UML is a notation system so that everyone can communicate about a model, it's developed from methodologies that also describe the processes in developing and using the model. While there is no one accepted process, the contributors to UML all describe somewhat similar approaches and these are usually described along with tutorials about UML itself.

Among the concepts of modeling that UML specifies how to describe are: class (of objects), object, association, responsibility, activity, interface, use case, package, sequence, collaboration, and state. Fowler's book provides a good introduction to UML. Booch, Rumbaugh, and Jacobson all have or soon will have published the "offficial" set of books on UML.

Keith Dawson reports in his newsletter, Tasty Bits from the Technology Front, that UML books are best-sellers in the computer sections of bookstores.

OOP (object-oriented programming)

A revolutionary concept that changed the rules in computer program development, object-oriented programming (OOP) is organized around objects rather than actions, data rather than logic. Historically, a program has been viewed as a logical procedure that takes input data, processes it, and produces output data. The programming challenge was seen as how to write the logic, not how to define the data. Object-oriented programming takes the view that what we really care about are the objects we want to manipulate rather than the logic required to manipulate them. Examples of objects range from human beings (described by name, address, and so forth) to buildings and floors (whose properties can be described and managed) down to the little widgets on your computer desktop (such as buttons and scroll bars).

The first step in OOP is to identify all the objects you want to manipulate and how they relate to each other, an exercise often known as data modeling. Once you've identified an object, you generalize it as a class of objects (think of Plato's concept of the "ideal" chair that stands for all chairs) and define the kind of data it contains and any logic sequences that can manipulate it. The logic sequences are known as methods. A real instance of a class is called (no surprise here) an "object" or, in some environments, an "an instance of a class." The object or class instance is what you run in the computer. Its methods provide computer instructions and the class object characteristics provide relevant data.

The concepts and rules used in object-oriented programming provide these important benefits:

- The concept of a data class makes it possible to define subclasses of data objects that share some or all of the main class characteristics. Called inheritance, this property of OOP forces a more thorough data analysis, reduces development time, and ensures more accurate coding.
- Since a class defines only the data it needs to be concerned with, when an instance of that class (an object) is run, the code will not be able to accidentally access other program data. This characteristic of data hiding provides greater system security and avoids unintended data corruption.
- The definition of a class is reuseable not only by the program for which it is initially created but also by other object-oriented programs (and, for this reason, can be more easily distributed for use in networks).
- The concept of data classes allows a programmer to create new data types that are not defined in the language itself.

One of the first object-oriented computer languages was called Smalltalk. C++ and Java are the most popular object-oriented languages today. A subset of C++, the Java programming language is designed especially for use in distributed applications on corporate networks and the Internet.

Class

In object-oriented programming, a class is a template definition of the methods and variables in a particular kind of object. Thus, an object is a specific instance of a class; it contains real values instead of variables.

The class is one of the defining ideas of object-oriented programming.
Among the important ideas about classes are:

- A class can have subclasses that can inherit all or some of the characteristics of the class. In relation to each subclass, the class becomes the superclass.
- Subclasses can also define their own methods and variables that are not part of their superclass.
- The structure of a class and its subclasses is called the class hierarchy.

SQL

SQL (Structured Query Language) is a standard interactive and programming language for getting information from and updating a database. Although SQL is both an ANSI and an ISO standard, many database products support SQL with proprietary extensions to the standard language. Queries take the form of a command language that lets you select, insert, update, find out the location of data, and so forth. There is also a programming interface.

SUPRA® Server SQL - Cincom

Are your legacy database systems keeping you from moving into the 21st Century? We can help!

Cincom promoted and sold Databases before anyone else. You may have heard of TOTAL®, the first database management system sold as independent software. We are still in the database market and will continue to be strongly into the 21st century. Why not look to the oldest, most experienced database company in the world to help solve your newest problems?

Is the price and scalability of Windows NT® attractive to you?

SUPRA® Server is the easiest to maintain Relational Database on the market today. It was designed and built to be scalable and with the recent release of the Windows NT version, is positioned to help you solve any data management problem you may have.

SUPRA Server has the same look and feel on any major hardware platform, IBM MVS and VSE™, UNIX, OpenVMS and NT. All applications are completely portable across any of these systems and the data is managed the same on NT as on MVS. This makes SUPRA Server extremely powerful and able to handle any volume of data needed. In addition, any of these platforms can function as a client.

Is your data on VSAM? SUPRA Server can access it natively and make it available to any SQL based application using ODBC.

We are not content to rest on our laurels. Cincom keeps moving forward with TOTAL Framework's Data Access FrameWork and UniSQL® Server. This object-relational system allows you to map and access data from both pre-relational and relational databases. UniSQL Server's SQL, Object SQL, extends the SQL language to allow the creation of objects and extend the capabilities of relational tables. UniSQL Server is a cost-effective means to move forward technologically while maintaining your investment in legacy systems and their attendant skill sets.

We offer MANTIS®, a widely used, 4th generation application development tool, to enable you to customize and build any solution you need. For those wanting the latest technology, we offer ObjectStudio™, an object-based application development package that utilizes all the latest object-oriented technology.

Cincom allows you to continue getting the performance of your mainframe while moving forward into the world of client/server and object-oriented technologies. Cincom is the only database company in the world that can meet the goals of high performance, cross-platform integration and spanning the technology bridge into the 21st century.

SUPRA® Server SQL

This advanced, ANSI-compliant Relational Database Management System can help you bring your legacy mainframe data into the SQL, client/server, ODBC world. It is by far the easiest database to administer and maintain available today. The NT version makes scalability a non-issue; it was designed and built with scalability in mind. SUPRA was intended to be a client/server database and runs on any major hardware platform, from IBM® mainframes to OpenVMS™, UNIX and of course Windows NT®.

Is your existing data on VSAM KSDS? SUPRA can access this data and make it available to ODBC-based applications.

Please call Cincom about getting a trial copy of SUPRA Server SQL for NT and see for yourself what high performance and cross-platform access can do for you.

SUPRA Server PDM

Provides a high-performance, extremely reliable production database that, as with all Cincom databases, runs on most major hardware platforms. If you need speed and dependability, this is the database for you. This data can also be accessed through both of our newer technologies, SUPRA Server SQL and TOTAL Data Access FrameWork as well as ODBC.

Desktop Access To PDM

SUPRA Server PDM can be accessed from your desktop by two methods:

1. Windows Client Support allows you to develop and run your SUPRA PDM applications from a Windows environment using the DATBAS interface.
2. Data Access Services allows you to access your SUPRA PDM from desktop tools that use ODBC and the SQL language.

Cincom is dedicated to helping you solve your database technology problems. Please call today and ask for more information about the easiest, most cost-effective way to the 21st Century.

Cincom Systems, a developer and marketer of strategic business software to leading corporations for 30 years, is ranked among the world's largest independent software companies, and operates in every major market worldwide.

Cincom Systems, Inc.
World Headquarters   55 Merchant Street
Cincinnati, OH 45246-3732   (513) 612-2300
E-mail: info@cincom.com or fax (513) 612-2000

Metadata

Meta is a prefix that in most information technology usages means "an underlying definition or description." Thus, metadata is a definition or description of data and metalanguage is a definition or description of language. Meta (pronounced MEH-tah in the U.S. and MEE-tah in the U.K.) derives from Greek, meaning "among, with, after, change." Whereas in some English words the prefix indicates "change" (for example, metamorphosis), in others, including those related to data and information, the prefix carries the meaning of "more comprehensive or fundamental."

The Standard Generalized Markup Language (SGML) defines rules for how a document can be described in terms of its logical structure (headings, paragraphs or idea units, and so forth). SGML is often referred to as a metalanguage because it provides a "language for how to describe a language." A specific use of SGML is called a document type definition (DTD). A document type definition spells out exactly what the allowable language is. A DTD is thus a metalanguage for a certain type of document. (In fact, the Hypertext Markup Language (HTML) is an example of a document type definition. HTML defines the set of HTML tags that any Web page can contain.)

The Extensible Markup Language (XML), which is comparable to SGML and modelled on it, describes how to describe a collection of data. It's sometimes referred to as metadata. A specific XML definition, such as Microsoft's new Channel Definition Format (CDF), defines a set of tags for describing a Web channel. XML could be considered the metadata for the more restrictive metadata of CDF (and other future data definitions based on XML).

In the case of SGML and XML, "meta" connotes "underlying definition" or set of rules. In other usages, "meta" seems to connote "description" rather than "definition." For example, the HTML <META> tag is used to enclose descriptive language about an HTML page.

One could describe any computer programming or user interface as a metalanguage for conversing with a computer. And an English grammar and dictionary together could be said to define (and describe) the metalanguage for spoken and written English.

Front-end and Back-end

Front-end and back-end are terms used to characterize program interfaces and services relative to the initial user of these interfaces and services. (The "user" may be a human being or a program.) A "front-end" application is one that application users interact with directly. A "back-end" application or program serves indirectly in support of the front-end services, usually by being closer to the required resource or having the capability to communicate with the required resource. The back-end application may interact directly with the front-end or, perhaps more typically, is a program called from an intermediate program that mediates front-end and back-end activities.

For example, the Telephony Application Program Interface (TAPI) is sometimes referred to as a front-end interface for telephone services. A program's TAPI requests are mapped by Microsoft's TAPI Dynamic Link Library programs (an intermediate set of programs) to a "back-end" program or driver that makes the more detailed series of requests to the telephone hardware in the computer.

As another example, a front-end application might interface directly with users and forward requests to a remotely-located back-end program in another computer to get requested data or perform a requested service. Relative to the client/server computing model, a front-end is likely to be a client and a back-end to be a server.

Visual Basic

Visual Basic is a programming environment from Microsoft in which a programmer uses a graphical user interface to choose and modify preselected chunks of code written in the BASIC programming language.

Since Visual Basic is easy to learn and fast to write code with, it's sometimes used to prototype an application that will later be written in a more difficult but efficient language. Visual Basic is also widely used to write working programs. Microsoft says that there are at least 3 million developers using Visual Basic.

3-tier application

A 3-tier application is an application program that is organized into three major parts, each of which is distributed to a different place or places in a network. The three parts are:

The workstation or presentation interface
The business logic
The database and programming related to managing it

In a typical 3-tier application, the application user's workstation contains the programming that provides the graphical user interface (GUI) and application-specific entry forms or interactive windows. (Some data that is local or unique for the workstation user is also kept on the local hard disk.)

Business logic is located on a local area network (LAN) server or other shared computer. The business logic acts as the server for client requests from workstations. In turn, it determines what data is needed (and where it is located) and acts as a client in relation to a third tier of programming that might be located on a mainframe computer.

The third tier includes the database and a program to manage read and write access to it. While the organization of an application can be more complicated than this, the 3-tier view is a convenient way to think about the parts in a large-scale program.

A 3-tier application uses the client/server computing model. With three tiers or parts, each part can be developed concurrently by different team of programmers coding in different languages from the other tier developers. Because the programming for a tier can be changed or relocated without affecting the other tiers, the 3-tier model makes it easier for an enterprise or software packager to continually evolve an application as new needs and opportunities arise. Existing applications or critical parts can be permanently or temporarily retained and encapsulated within the new tier of which it becomes a component.

The 3-tier application architecture is consistent with the ideas of distributed object-oriented programming.

CASE (computer-aided software engineering)

CASE (computer-aided software engineering) is the use of a computer-assisted method to organize and control the development of software, especially on large, complex projects involving many software components and people. Using CASE allows designers, code writers, testers, planners, and managers to share a common view of where a project stands at each stage of development. CASE helps ensure a disciplined, check-pointed process. A CASE tool may portray progress (or lack of it) graphically. It may also serve as a repository for or be linked to document and program libraries containing the project's business plans, design requirements, design specifications, detailed code specifications, the code units, test cases and results, and marketing and service plans.

CASE originated in the 1970s when computer companies were beginning to borrow ideas from the hardware manufacturing process and apply them to software development (which generally has been viewed as an insufficiently disciplined process). Some CASE tools supported the concepts of structured programming and similar organized development methods. More recently, CASE tools have had to encompass or accommodate visual programming tools and object-oriented programming. In corporations, a CASE tool may be part of a spectrum of processes designed to ensure quality in what is developed. (Many companies have their processes audited and certified as being in conformance with the ISO 9000 standard.)

Some of the benefits of CASE and similar approaches are that, by making the customer part of the process (through market analysis and focus groups, for example), a product is more likely to meet real-world requirements. Because the development process emphasizes testing and redesign, the cost of servicing a product over its lifetime can be reduced considerably. An organized approach to development encourages code and design reuse, reducing costs and improving quality. Finally, quality products tend to improve a corporation's image, providing a competitive advantage in the marketplace.

A template (from French templet, diminutive of temple, a part of a weaving loom for keeping it stretched transversely) is a form, mold, or pattern used as a guide to making something. Here are some examples:

A ruler is a template when used to draw a straight line.
A document in which the standard opening and closing parts are already filled in is a template that you can copy and then fill in the variable parts. (This HTML page was created using such a template.)
An overlay that you put on your computer keyboard telling you special key combinations for a particular application is a template for selecting the right keys to press.
Flowcharting templates (not used much now) help programmers draw flowcharts or logic sequences in preparation for writing the code.
In programming, a template is a generic class or other unit of source code that can be used as the basis for unique units of code. In C++, an object-oriented computing language, there are Standard Template Libraries from which programmers can choose individual template classes to modify. The Microsoft Foundation Class (MFC) Library is an example.

Open system

An open system (as opposed to a proprietary system) is one that adheres to a publicly-known and sometimes standard set of interfaces so that anyone using it can also use any other system that adheres to the standard. In computers, an open operating system is one for which one can write application programs that will then run on other companies' open operating systems currently or in the future. The best-known open operating system is UNIX, which was created or allowed to develop as a public collaboration, mainly among some large universities. Today, all operating systems that adhere to the Single UNIX Specification can be considered to be open. The advantages of openness are that users (including programmers and engineers) can learn a single set of skills and find that they are portable across the industy they work in. Likewise, companies will find they can spend less on developing skills for using and working on their own product development.

Among a number of organizations that are concerned with promoting open systems are the Open Software Foundation (OSF) and the X/Open Company, which, in February, 1996, combined their organizations as The Open Group.

Legacy Applications

In information technology, legacy applications and data are those that have been inherited from languages, platforms, and techniques earlier than current technology. Most enterprises who use computers have legacy applications and databases that serve critical business needs. Typically, the challenge is to keep the legacy application running while converting it to newer, more efficient code that makes use of new technology and programmer skills. In the past, much programming has been written for specific manufacturers' operating systems. Currently, many companies are migrating their legacy applications to new programming languages and operating systems that follow open or standard programming interfaces. Theoretically, this will make it easier in the future to update applications without having to rewrite them entirely and will allow a company to use its applications on any manufacturer's operating system.

In addition to moving to new languages, enterprises are redistributing the locations of applications and data. In general, legacy applications have to continue to run on the platforms they were developed for. Typically, new development environments account for the need to continue to support legacy applications and data. With many new tools, legacy databases can be accessed by newer programs.

Artificial Intelligence (AI)

Artificial intelligence (AI) is the simulation of human intelligence processes by machines, especially computer systems. These processes include learning (the acquisition of information and rules for using the information), reasoning (using the rules to reach approximate or definite conclusions), and self-correction. Particular instances of AI are called expert systems.

Expert System

An expert system is a computer program that simulates the judgement and behavior of a human or an organization that has expert knowledge and experience in a particular field. Typically, such a system contains a knowledge base containing accumulated experience and a set of rules for applying the knowledge base to each particular situation that is described to the program. Sophisticated expert systems can be enhanced with additions to the knowledge base or to the set of rules.

Among the best-known expert systems have been those that play chess and that assist in medical diagnosis.

Driver

A driver is a program that interacts with a particular device or special (frequently optional) kind of software. The driver contains the special knowledge of the device or special software interface that programs using the driver do not. In personal computers, a driver is often packaged as a dynamic link library (DLL) file.

Device Driver

A device driver is a program that controls a particular type of device that is attached to your computer. There are device drivers for printers, displays, CD-ROM readers, diskette drives, and so on. When you buy an operating system, many device drivers are built into the product. However, if you later buy a new type of device that the operating system didn't anticipate, you'll have to install the new device driver. A device driver essentially converts the more general input/output instructions of the operating system to messages that the device type can understand.

Dynamic link library (DLL)

In computers, a dynamic link library (DLL) is a collection of small programs, any of which can be called when needed by a larger program that is running in the computer. The small program that lets the larger program communicate with a specific device such as a printer or scanner is often packaged as a DLL program (usually referred to as a DLL file).

The advantage of DLL files is that, because they don't get loaded into random access memory (RAM) together with the main program, space is saved in RAM. When and if a DLL file is needed, then it is loaded and run. For example, as long as a user of Microsoft Word is editing a document, the printer DLL file does not need to be loaded into RAM. If the user decides to print the document, then the Word application causes the printer DLL file to be loaded and run.

A DLL file is often given a ".dll" file name suffix. DLL files are dynamically linked with the program that uses them during program execution rather than being compiled with the main program. The set of such files (or the DLL) is somewhat comparable to the library routines provided with programming languages such as C and C++.

Default

In computer technology, a default (noun, pronounced dee-FAWLT) is a predesigned value or setting that is used by a computer program when a value or setting is not specified by the program user. The program user can be either an interactive user of a graphical user interface or command line interface, or a programmer using an application program interface. When the program receives a request from an interactive user or another program, it looks at the information that has been passed to it. If a particular item of information is not specified in the information that is passed, the program uses the default value that was defined for that item when the program was written. In designing a program, each default is usually preestablished as the value or setting that most users would probably choose. This keeps the interface simpler for the interface user and means that less information has to be passed and examined during each program request.

To the program requestor, to default (verb) is to intentionally or accidentally allow the preestablished value or setting for an item to be used by the program. The program is said to default when it uses a default value or setting.

Default (adjective) pertains to something that is used when something else is not supplied or specified. For example, a default printer is a type of printer that is assumed to be connected to a computer unless the computer user specifies another type that is actually connected.

File system

1) In a computer, a file system is the way in which files are named and where they are placed logically for storage and retrieval. The DOS, Windows, OS/2, Macintosh, and UNIX-based operating systems all have file systems in which files are placed somewhere in a hierarchical (tree) structure. A file is placed in a directory (folder in Windows) or subdirectory at the desired place in the tree structure. File systems specify conventions for naming files. These conventions include the maximum number of characters in a name, which characters can be used, and, in some systems, how long file name suffixes can be. A file system also includes a format for specifying the path to a file through the structure of directories.

2) Sometimes the term refers to the part of an operating system or an added-on program that supports a file system as defined in (1). Examples of such add-on file systems include the Network File System (NFS) and the Andrew file system (AFS).

File

1) In data processing, using an office metaphor, a file is a related collection of records. For example, you might put the records you have on each of your customers in a file. In turn, each record would consist of fields for individual data items, such as customer name, customer number, customer address, and so forth. By providing the same information in the same fields in each record (so that all records are consistent), your file will be easily accessible for analysis and manipulation by a computer program. This use of the term has become somewhat less important with the advent of the database and its emphasis on the table as a way of collecting record and field data. In mainframe systems, the term data set is generally synonymous with file but implies a specific form of organization recognized by a particular access method. Depending on the operating system, files (and data sets) are contained within catalogs, directories, or folders.

2) In any computer system but especially in personal computers, a file is an entity of data available to system users (including the system itself and its application programs) that is capable of being manipulated as an entity (for example, moved from one file directory to another). The file must have a unique name within its own directory. Some operating systems and applications describe files with given formats by giving them a particular file name suffix. (The file name suffix is also known as a file name extension.) For example, a program or executable file is sometimes given or required to have an ".exe" suffix. In general, the suffixes tend to be as descriptive of the formats as they can within the limits of the number of characters allowed for suffixes by the operating system.

Extension or Suffix

1) In computer operating systems, a file name extension is an optional addition to the file name in a suffix of the form ".xxx" where "xxx" represents a limited number of alphanumeric characters depending on the operating system. (In Windows 3.1, for example, a file name extension or suffix can have no more than three characters, but in Windows 95, it can have more.) The file name extension allows a file's format to be described as part of its name so that users can quickly understand the type of file it is without having to "open" or try to use it. The file name extension also helps an application program recognize whether a file is a type that it can work with.

Path (and Pathname)

1) In a computer operating system, a path is the route through a file system to a particular file. A pathname (or path name) is the specification of that path. Each operating system has its own format for specifying a pathname. The DOS, Windows, and OS/2 operating systems use this format:

driveletter:\directoryname\subdirectoryname\filename.suffix

Windows uses the term folder instead of directory.

In UNIX-based systems, the format is:

/directory/subdirectory/filename

In UNIX, the storage drive location is not an explicit part of the path name (and UNIX systems usually use two words for path name).

In all operating systems, an absolute pathname (or fully qualified path name) specifies the complete path name. A relative pathname specifies a path relative to the directory to which the operating system is currently set.

The World Wide Web's HTTP program uses a pathname as part of a Uniform Resource Locator (URL).

2) In a network, a path is a route between any two points or nodes.

3) In a number of products or applications, a path is a route to or between points within a given organized structure.

4) In IBM's Virtual Telecommunication Access Method (VTAM), a path identifies a particular dial-out port.

File Allocation Table (FAT and FAT32)

A file allocation table (FAT) is a table that an operating system maintains on a hard disk that provides a map of the clusters (the basic unit of logical storage on a hard disk) that a file has been stored in. When you write a new file to a hard disk, the file is stored in one or more clusters that are not necessarily next to each other; they may be rather widely scattered over the disk. A typical cluster size is 2,048 bytes, 4,192 bytes, or 8,096 bytes. The operating system creates a FAT entry for the new file that records where each cluster is located and their sequential order.

When you read a file, the operating system reassembles the file from clusters and places it as an entire file where you want to read it. For example, if this is a long Web page, it may very well be stored on more than one cluster on your hard disk.

Until Windows 95 OSR2 (OEM Release 2), DOS and Windows file allocation table entries were 16 bits in length, limiting hard disk size to 128 megabytes, assuming a 2,048 size cluster. Up to 512 megabyte support is possible assuming a cluster size of 8,192 but at the cost of using clusters inefficiently. DOS 5.0 and later versions provide for support of hard disks up to two gigabytes with the 16-bit FAT entry limit by supporting separate FATs for up to four partitions.

With 32-bit FAT entry (FAT32) support in Windows 95 OSR2, the largest size hard disk that can be supported is two terabytes! However, personal computer users are more likely to take advantage of FAT32 with 5 or 10 gigabyte drives.

Font

A font is a set of printable or displayable text characters in a specific style and size. The type design for a set of fonts is the typeface and variations of this design form the typeface family. Thus, Helvetica is a typeface family, Helvetica italic is a typeface, and Helvetica italic 10-point is a font. In practice, font and typeface are often used without much precision, sometimes interchangably.

An outline font is a software typeface that can generate a scalable range of font sizes. A bitmap font is a digital represention of a font that is already fixed in size or a limited set of sizes. The two most popular outline font software programs on today's computers are TrueType and Adobe's Type 1. TrueType fonts come with both Windows and Macintosh operating systems. However, Type 1 is a standard outline font (ISO 9541). Both TrueType and Type 1 fonts can be used by Adobe's PostScript printers (although Adobe says that Type 1 fonts makes fuller use of the PostScript language).

Independent developers and graphic designers create new typefaces for both TrueType and Type 1. Adobe states that there are over 30,000 Type 1 fonts available. Fonts (in addition to those that come with your computer) can be purchased as individual typeface families or in typeface collections.

Fonts

Fonts are characters of a specific style and size within an overall typeface design. Printers use resident fonts and soft fonts to print documents. Resident fonts are built into the hardware of a printer. They are also called internal fonts or built-in fonts. All printers come with one or more resident fonts. Additional fonts can be added by inserting a font cartridge into the printer or installing soft fonts to the hard drive. Resident fonts cannot be erased unlike soft fonts. Soft fonts are installed onto the hard drive and then sent to the computer's memory when a document is printed that uses the particular soft font. Soft fonts can be purchased in stores or downloaded from the Internet.

There are two types of fonts used by the printer and screen display, bitmap fonts and outline fonts. Bitmap fonts are digital representations of fonts that are not scalable. This means they have a set size or a limited set of sizes. For example, if a document using a bitmap font sized to 24 point is sent to the printer and there is not a bitmap font of that size, the computer will try to guess the right size. This results in the text looking stretched-out or squashed. Jagged edges are also a problem with bitmap fonts. Outline fonts are mathematical descriptions of the font that are sent to the printer. The printer then rasterizes or converts them to the dots that are printed on the paper. Because they are mathematical, they are scalable. This means the size of the font can be changed without losing the sharpness or resolution of the printed text. TrueType and Type 1 fonts are outline fonts. Outline fonts are used with Postscript and PCL printer languages.

Typeface

A typeface is a design for a set of printer or display fonts, each for a set of characters, in a number of specific sizes. Since outline fonts such as TrueType and Type 1 are scalable, a computer typeface designer must anticipate the possibility of the design being scaled through a range of sizes.

Typefaces often come as a family of typefaces, with individual typefaces for italic, bold, and other variations in the main design.

Bit map (or bitmap or Bmp)

A bit map defines a display space and the color for each pixel or "bit" in the display space. A GIF and a JPEG are examples of graphic image file types that contain bit maps.

A bit map does not need to contain a bit of color-coded information for each pixel on every row. It only needs to contain information indicating a new color as the display scans along a row. Thus, an image with much solid color will tend to require a small bit map.

Because a bit map uses a fixed or raster method of specifying an image, the image cannot be immediately rescaled by a user without losing definition. A vector graphic image, however, is designed to be quickly rescaled. Typically, an image is created using vector graphics and then, when the artist is satisifed with the image, it is converted to (or saved as) a raster graphic file or bit map.

WYSIWYG (what you see is what you get)

A WYSIWYG (pronounced "wiz-ee-wig") editor or program is one that allows an interface or content developer to create a graphical user interface (GUI) or page of text so that the developer can see what the end result will look like while the interface or document is being created. A WYSIWYG editor can be contrasted with more traditional editors that require the developer to enter descriptive codes (or markup) and do not permit an immediate way to see the results of the markup.

For example, this page was created with a very handy tool, HTML Assistant Pro, that assists in inserting markup but still requires that the developer think in terms of markup. (HTML Assistant Pro and similar editors do let you test your markup very readily with a browser.) A true WYSIWYG editor, such as Microsoft's FrontPage or Adobe's PageMill, conceals the markup and allows the developer to think entirely in terms of how the content should appear. (One of the trade-offs, however, is that a WYSIWYG editor does not always make it easy to fine-tune its results.)

Macintosh

The Macintosh (often called "the Mac"), introduced in 1984 by Apple Computer, was the first widely-sold personal computer with a graphical user interface (GUI). The Mac was designed to provide users with a natural, intuitively understandable, and, in general, "user-friendly" computer interface. Many of the user interface ideas in the Macintosh derived from experiments at the Xerox Parc laboratory in the early 1970s, including the mouse, the use of icons or small visual images to represent objects or actions, the point-and-click and click-and-drag actions, and a number of window operation ideas. Microsoft was successful in adapting user interface concepts first made popular by the Mac in its first Windows operating system.

The Macintosh has its own operating system, Mac OS. Originally built on a line of Motorola microprocessors, Mac versions today are powered by the PowerPC microprocessor, which was developed jointly by Apple, Motorola, and IBM. The Mac is actually a line of personal computers, configured for individual users and businesses with different needs. A recent product, iMac, provides the Mac technology and interface in a low-cost package.

While Mac users represent only about 5% of the total numbers of personal computer users, Macs are highly popular and almost a cultural necessity among graphic designers and online visual artists and the companies they work for. In general, Mac users tend to be enthusiasts.

Mac OS

Mac OS is the computer operating system for Apple Computer's Macintosh line of personal computers and workstations. A popular feature of its latest version, Mac OS 8.5, is Sherlock, a search facility similar to a "find a file" command. However, Sherlock searches popular directories and search engines on the Internet and then formats the results somewhat as though they were clickable files in the Macintosh file system.

Mac OS comes with Apple Computer's iMac and Power Macintosh line of computers.

Windows 98

Windows 98 (called "Memphis" during development and previously called "Windows 97" based on an earlier schedule) is the latest release of Microsoft's Windows operating system for personal computers. Windows 98 expresses Microsoft's belief that users want and should have a global view of their potential resources and that Web technology should be an important part of the user interface. Although building Microsoft's own Web browser into the user desktop has been an issue in the U.S. Justice Department's suit, Windows 98 was released as planned with its tightly integrated browser.

In Windows 98, Microsoft's Internet Explorer is an integral part of the operating system. Using the Active Desktop of Windows 98, you can view and access desktop objects that reside on the World Wide Web as well as local files and applications. The Windows 98 desktop is, in fact, a Web page with HTML links and features that exploit Microsoft's ActiveX controls.

With Windows 98 (or with Internet Explorer 4.0 in Windows 95), you can set up news and other content to be pushed to you from specified Web sites.

Windows 98 also provides a 32-bit file allocation table (FAT32) that allows you to have a single-partition disk drive larger than 2 Gbytes. Other features in Windows 98 include:

- Support for Universal Serial Bus (USB), which makes it easy to plug in new devices
- Support for Digital Versatile Disc (DVD)
- Support for a new industry-standard form of power management called Advanced Configuration and Power Interface (ACPI)

Windows 98's desktop browser is being challenged by Netscape's Netcaster (previously known as Constellation), a feature of its Communicator suite that uses a similar desktop approach but runs on a number of platforms besides Windows.

In the next major version of Windows for personal computer users, Windows 98 and Windows NT are expected to become a single operating system.

Windows NT

Windows NT is the Microsoft Windows personal computer operating system designed for users and businesses needing advanced capability. Windows NT (which may originally have stood for "New Technology," although Microsoft doesn't say) is actually two products: Microsoft NT Workstation and Microsoft NT Server. The Workstation is designed for users, especially business users, who need faster performance and a system a little more fail-safe than Windows 95 (and perhaps Windows 98). The Server is designed for business machines that need to provide services for LAN-attached computers. The Server is required, together with an Internet server such as Microsoft's IIS, for a Windows system that plans to serve Web pages.

Windows NT Workstation: Microsoft says that 32-bit applications will run 20% faster on this system than on Windows 95 (assuming both have 32 megabytes of RAM). Since older 16-bit applications run in a separate address space, one can crash without crashing other applications or the operating system. Security and management features not available on Windows 95 are provided. The Workstation has the same desktop user interface as Windows 95. It's expected that Windows 98 and the Windows NT line of operating systems will converge in the next release beyond Windows 98.

Windows NT Server: The NT Server is probably the second most installed network server operating system after Novell's NetWare operating system. Microsoft claims that its NT servers are beginning to replace both NetWare and the various UNIX-based systems such as those of Sun Microsystems and Hewlett-Packard. NT Server 5.0., still in beta test in early 1999, is now a product line called Windows 2000. Notable features of the Windows 2000 products are:

A fully-customizable administrative console that can be based on tasks rather than files, applications, or users

A new file directory approach called Active Directory that lets the administrator and other users view every file and application in the network from a single point-of-view.

Dynamic Domain Name Server (DNS), which replicates changes in the network using the Active Directory Services, the Dynamic Host Configuration Protocol (DHCP), and the Windows Internet Naming Service (WINS) whenever a client is reconfigured.

The ability to create, extend, or mirror a disk volume without having to shut down the system and to back up data to a variety of magnetic and optical storage media.

A Distributed File System (DFS) that lets users see a distributed set of files in a single file structure across departments, divisions, or an entire enterprise.

Close integration with and support for Microsoft's Message Queue Server, Transaction Server, and Internet Information Server (IIS).

Windows 2000

Windows 2000 is the new version of the Windows operating system that Microsoft plans to release in 1999. Previously called Windows NT 5.0, Windows 2000 will be advertised as "Built on NT Technology." As a name, Windows 2000 is designed to appeal to small business and professional users as well as to the more technical and larger business market for which the NT was designed. For many Windows 95 and Windows 98 users, Windows 2000 may be regarded as the next step.

According to Microsoft, the Windows 2000 product line will consist of four products:

Windows 2000 Professional, aimed at individuals and businesses of all sizes. It will include security and mobile use enhancements. It will be the most economical choice. Windows 2000 Server, aimed at small-to-medium size businesses. It can function as a Web server and/or a workgroup (or branch office) server. It can be part of a two-way symmetric multiprocessing (SMP) system. NT 4.0 servers can be upgraded to this server. Windows 2000 Advanced Server, aimed at being a network operating system server and/or an application server, including those involving large databases. This server facilitates clustering and load-balancing. NT 4.0 servers with up to eight-way SMP can upgrade to this product.

Windows 2000 Datacenter Server, designed for large data warehouses, online transaction processing (OLTP), econometric analysis, and other applications requiring high-speed computation and large databases. The Datacenter Server will support up to 16-way SMP and up to 64 GB of physical memory.

Freeware

Freeware is programming that is offered at no cost. However, it is copyrighted so that you can't incorporate its programming into anything you may be developing. The least restrictive "no-cost" programs are uncopyrighted programs in the public domain. These include a number of small UNIX programs. When reusing public domain software in your own programs, it's good to know the history of the program so that you can be sure it really is in the public domain.

You can find a great deal of shareware and freeware at http://www.shareware.com.

Shareware

Shareware is software that is distributed free on a trial basis with the understanding that the user may need or want to pay for it later. Some software developers offer a shareware version of their program with a built-in expiration date (after 30 days, the user can no longer get access to the program). Other shareware (sometimes called liteware) is offered with certain capabilities disabled as an enticement to buy the complete version of the program.

Freeware is programming that is offered at no cost. However, it is copyrighted so that you can't incorporate its programming into anything you may be developing. The least restrictive "no-cost" programs are uncopyrighted programs in the public domain. These include a number of small UNIX programs. When reusing public domain software in your own programs, it's good to know the history of the program so that you can be sure it really is in the public domain.

Liteware

Liteware is a term for software that is distributed freely in a version having less capability than the full for-sale version. It's usually designed to provide a potential customer with a sample of the "look-and-feel" of a product and a subset of its full capability. Liteware can be considered a type of shareware (where shareware also includes products distributed freely, usually on a trial basis, that do have full capability).

An example of liteware is HTML Assistant Pro, an HTML editor. We tried their liteware version, which allows you to create HTML files (Web pages) and shows you, but doesn't let you use, table and form creation facilities. Since we liked the liteware version and needed all the capabilities, we ordered the full product.

Data Warehouse

A data warehouse is a central repository for all or significant parts of the data that an enterprise's various business systems collect. The term was coined by W. H. Inmon. IBM sometimes uses the term "information warehouse." Typically, a data warehouse is housed on an enterprise mainframe server. Data from various online transaction processing (OLTP) applications and other sources is selectively extracted and organized on the data warehouse database for use by analytical applications and user queries. Data warehousing emphasizes the capture of data from diverse sources for useful analysis and access, but does not generally start from the point-of-view of the end user or knowledge worker who may need access to specialized, sometimes local databases. The latter idea is known as the data mart.

Data mining and decision support systems (DSS) are two of the kinds of applications that can make use of the data warehouse.

Virus

A virus is a piece of programming code inserted into other programming to cause some unexpected and, for the victim, usually undesirable event. Viruses can be transmitted by downloading programming from other sites or be present on a diskette. The source of the file you're downloading or of a diskette you've received is often unaware of the virus. The virus lies dormant until circumstances cause its code to be executed by the computer. Some viruses are playful in intent and effect ("Happy Birthday, Ludwig!") and some can be quite harmful, erasing data or causing your hard disk to require reformatting.

Generally, there are three main classes of viruses:

File infectors. These viruses attach themselves to program files, usually selected .COM or .EXE files. Some can infect any program for which execution is requested, including .SYS, .OVL, .PRG, and .MNU files. When the program is loaded, the virus is loaded as well.

System or boot-record infectors. These viruses infect executable code found in certain system areas on a disk. They attach to the DOS boot sector on diskettes or the Master Boot Record on hard disks. A typical scenario (familiar to the author) is to receive a diskette from an innocent source that contains a boot disk virus. When your operating system is running, files on the diskette can be read without triggering the boot disk virus. However, if you leave the diskette in the drive, and then turn the computer off or reload the operating system, the computer will look first in your A drive, find the diskette with its boot disk virus, load it, and make it temporarily impossible to use your hard disk. (Allow several days for recovery.) This is why you should make sure you have a bootable floppy.

Macro viruses. These are among the most common viruses, and they tend to do the least damage. Macro viruses infect your Microsoft Word application and typically insert unwanted words or phrases.

The best protection against a virus is to know the origin of each program or file you load into your computer. Since this is difficult, you can buy anti-virus software that typically checks all of your files periodically and can remove any viruses that are found. From time to time, you may get an e-mail message warning of a new virus. Chances are good that the warning is a virus hoax.

Anti-virus software

Anti-virus (or "anti-viral") software is a class of program that searches your hard drive and floppy disks for any known or potential viruses. The market for this kind of program has expanded because of Internet growth and the increasing use of the Internet by businesses concerned about protecting their computer assets.

Here are three of the most popular anti-virus programs. You can download free-trial copies from their sites.

- Dr. Solomon's Software
- MacAfee Virus Scan
- Norton Anti-Virus

Year 2000 or "Y2K"

The year 2000 (also known as "Y2K") raises problems for anyone who depends on a program in which the year is represented by a two-digit number, such as "97" for 1997. Many programs written 10 or 15 years ago when storage limitations encouraged such information economies are still running in many companies. The problem is that when the two-digit space allocated for "99" rolls over to 2000, the next number will be "00." Frequently, program logic assumes that the year number gets larger, not smaller - so "00" may wreak havoc in a program that hasn't been modified to account for the millenium.

So pervasive is the problem in the world's legacy payroll, billing, and other programs that a new industry has sprung up dedicated to helping companies solve the problem. IBM and other major computer manufacturers, software houses, and consultants are presently offering tools and services to address this problem.