UNCLASSIFIED

AR-006-927

ELECTRONICS RESEARCH LABORATORY

Information Technology Division

GENERAL DOCUMENT
ERL-0594-GD

INFORMATION TECHNOLOGY IN PLAIN WORDS

SUMMARY

This document is a glossary of IT-related (Information Technology-related) terms directed specifically at the non-specialist reader. It is not for scientific professionals in any specific area, although their input has been invaluable in compiling this work. Further, it is not a formal Defence glossary in the sense of providing approved formal definitions of terms, and does not replace such definitions. This is a supplementary, and to some extent complementary, work.

JAN 92 COPY NO

APPROVED FOR PUBLIC RELEASE

POSTAL ADDRESS: Director, Electronics Research Laboratory, PO Box 1500, Salisbury, South Australia, 5108. ERL-0594-GD

UNCLASSIFIED

PREFACE

This document is a glossary of IT-related (Information Technology-related) terms directed specifically at the **non-specialist** reader. It is **not** for scientific professionals in any specific area, although their input has been invaluable in compiling this work. Further, it is not a formal Defence glossary in the sense of providing approved formal definitions of terms, and does not replace such definitions. This is a supplementary, and to some extent complementary, work.

It is hoped that, through this glossary, the non-specialist reader will be able to gain sufficient understanding of specialist areas so as to be comfortable reading technical papers, during technical and scientific discussions, and ultimately during the carriage of normal duties. To this end, and given the inherently informal nature of the document, some licence has been taken with respect to style, metre, perspective, use of colloquialisms, etc., for the sake of effect. This is a deliberate strategy, the absence of which would cause this document to be relegated to the bottom shelf of the bookcase, instead of to the desktop where it would be in easy reach.

Every effort has been made to ensure that the information given here is consistent with the formal definitions available from the approved Defence definitions available in numerous other formal documents. Notice of any discrepancies, omissions, suggestions or requests for further copies of this document will be gratefully received. Such comments and contributions should be forwarded to:

Chief, Information Technology Division
Defence Science and Technology Organisation
P.O. Box 1500
Salisbury
S.A. 5108

THIS IS A BLANK PAGE

CONTENTS

THIS IS A BLANK PAGE

A

acoustic coupler

A device which allows a telephone handset to provide access to the PSTN (Public Switched Telephone Network) for transmitting **data** between devices (computers). The PSTN carries voice, but the computer data is in the form of bits of 1s and 0s. Essentially, a 1-bit will be represented by a sound of a particular pitch, and a 0-bit will be represented by a markedly different pitch. These sounds are then transmitted over the telephone network (which is geared to transmit sound), and converted back into bits by an acoustic coupler at the other end.

acronym

Word formed from the initial letters of other words - e.g. ANSI, ASCII, Unesco, radar, laser, scuba. It doesn't have to be only the first letter, but can be the first few letters. Sometimes acronyms are made up that don't even match up with the meaning, for example NCOM, which stands for the Northern Territory Government Computing Service.

Ada

Ada is a high-level programming language originally sponsored by the US Department of Defense for use in the so-called *embedded system* application area. An embedded system is one in which the computer is an integral part of a larger system such as a chemical plant, missile or dishwasher.

The story of Ada goes back to about 1974 when the US DoD realized that it was spending far too much on software. A detailed analysis of the situation revealed that over half of the costs were directly attributable to embedded systems. Further analysis revealed that COBOL was universally used for data processing systems and FORTRAN was universally used for engineering computations. The number of different languages used for embedded systems, however, was enormous, and included many assembly languages. There was a need to standardize on one language.

A requirements specification was drawn up and published in 1975. This document was referred to as **Strawman.** This was later refined, and became **Woodenman.** A further iteration produced **Tinman** in 1976. One more iteration produced **Ironman,** and proposals were invited from contractors to design a new language against Ironman. Seventeen proposals were received, and four were selected to proceed in a design competition. These were from CII Honeywell Bull, Intermetrics, Softech and SRI International. The final choice was made on 2nd May 1979, when CII Honeywell Bull was declared the winner.

The DoD then announced that the new language would be known as **Ada** in honour of Augusta Ada Byron, Countess of Lovelace (1815-52). Ada, the daughter of Lord Byron, was the assistant and patron of Charles Babbage, and worked on his mechanical analytical engine. She could be considered the world's first programmer.

Ada is not to be thought of as just another programming language. Ada is about software engineering, software reusability and disciplined ways of working. In parallel with the language design, a series of requirements documents for an Ada Programming Support Environment (APSE) were developed (Sandman, Pebbleman and finally Stoneman). Currently, a number of smaller environment systems are emerging, with longer term efforts being focused on the establishment of Public Tool Interfaces which are intended to allow tools to be moved between different environments.

ADPCM

Adaptive Differential Pulse Code Modulation. A CCITT-recommended technique widely used for converting analogue signals to digital, normally used in voice applications. ADCPM can use 32Kbps links - *cf.* PCM.

AI

Artificial Intelligence. See "Artificial Intelligence".

algorithm

A defined procedure, or recipe, for doing something, whether this is to calculate the roots of a polynomial, or to bake a cake.

ANSI

This is an acronym for the American National Standards Institute, which is a body set up to define, maintain and co-ordinate standards in the United States. Data processing related standards are supervised by committees which are nominated "X" followed by a number as an identifier. For example, X3 is the FORTRAN Committee, X9 is the data encryption committee and X12 is the EDI standards committee. These should not be confused with the CCITT X.Series, which have a period (.) after the X - e.g. X.3 concerning packet assembly/disassembly.

Apple

The name of a company that manufactures PCs, the most famous model being the Macintosh. Apple PCs and IBM PCs are designed with entirely different architectures, and run with entirely different Operating Systems. The industry thrust towards developing facilities (software, hardware, services) for the IBM architecture transpired because of the sheer size of the IBM market (i.e. IBM market dominance). This is beginning to change because of the thrust towards "Open Systems", and facilities to interconnect proprietary systems, with products and communications protocols being developed to enable these different architectures to interoperate.

application

This is a general term for a program or suite of programs developed to process a particular function within a business or organization - e.g. payroll, staff leave, car usage, dole cheques, air traffic control.

Application Specific Integrated Circuit

Please refer to the entry under "ASIC".

architecture

Architecture is the science of building, or the style of building. A computer architecture is the way all the components are put together to form a computer. A network architecture is the way in which the computer network is constructed, and how it is integrated with the computer architecture.

Artificial Intelligence

Artificial Intelligence (AI) attempts to make a computer solve problems that give it the appearance of exhibiting some form of intelligence.

The computer, of course, has none of this apparent intelligence. It is merely acting as a superfast symbol manipulator. Humans still need to write programs to tell it how to manipulate the symbols.

A lot of AI work has progressed by studying games - particularly 2-person games, and it is useful to consider games by way of explanation. The classic example is Chess. Here, the AI techniques are tested to their limit. Consider the problem of trying to get a computer to play Chess. First of all, we have to write down all the rules. Things like "IF a Knight is on square (X,Y) THEN the legal moves are (X+2,Y+1) : (X+2,Y-1) : etc.".

When we come to write a program to figure out the BEST move to make, we're going to have to consider every POSSIBLE move from a given position. Now, this probably isn't going to enlighten us too much, because we usually need to look ahead quite a few moves before we can come to a reasonable decision about which is the best. In fact, the BEST move can only be determined by looking at every possible move, and for EACH of these, looking at every possible subsequent move, and for EACH of these, looking at every possible subsequent move, and so on until we hit a checkmate in our favour.

Now, this is an enormous task. No computer in the world could handle this, even if you gave it a million years - and that's a long time to wait for your opponent to move. Therefore, something has to be done to prune this huge tree of possibilities.

One thing that can be done is to employ a technique known as "alpha-beta cutoff", which is a particular pruning method that can be definitively applied given certain conditions relating to possible moves at the same level. One obvious pruning exercise is to not look ahead to the end of the game, but only down a few moves, and then somehow try to place a value on the (as yet incomplete) situations, or board positions. We can then backtrack to find out what the "best" move is. Of course, it's no longer the BEST move. It's only a guess at the best move.

Another pruning exercise is to try to pre-guess which, out of all the legal moves in a given situation, seem most likely to lead to a favourable situation. Such pruning could potentially save a huge amount of searching. Again, by resorting to this pruning technique, we can no longer assume that we will definitely arrive at the best situation, only that we guess that we are moving along a favourable path.

In both of these pruning techniques, we must employ something that is fundamental to AI - that is, the application of "heuristics". We need, in order to place some numerical value on a given board position, some way of deciding what is good and what is bad. We need, in order to decide to not pursue a particular legal move, some way of deciding what is good and what is bad. We can never KNOW, without searching the whole tree - begging the question, in the true sense of the expression.

A heuristic is a guess at what is good and what is bad, and placing some numerical value on it according to just how good or bad it is. Now obviously, there can be good heuristics and bad heuristics. For example, I might decide that it's a good idea to castle as soon as possible, but an EXPERT might say that this is likely to spell disaster. So, the programmer who is trying to write this Chess program needs the input of a chess expert to specify what is good and what is bad.

So we are now introduced to the idea of an "Expert System". One of the earliest perceived applications of these techniques is in the area of medical diagnosis. If we start with a symptom - say a sniff - the number of possible actual conditions, for which a sniff is a symptom, is huge. So we consider another symptom, in order to prune the tree of possibilities.

Finally, in order to make these Expert Systems simpler to develop, the "programming" is done using a special type of language. Imagine trying to write a chess program in COBOL! The available languages (e.g. LISP, PROLOG) are designed to make it easy to define the rules, and how the rules can be applied.

See also "Expert Systems".

Artificial Reality

See "Virtual Reality".

ASCII

An acronym for the American Standard Code for Information Interchange. This is just a definition of the way particular letters, numbers and various other special characters are to be represented inside a computer (in the form of bit patterns). There is another definition called EBCDIC (pronounced "ebsidic" - Extended Binary Coded Decimal Information Code), which is a code developed by IBM, and used in most of its mainframe-type computers. Interestingly enough, the IBM PC (Personal Computer) and its derivatives are all ASCII machines. Problems arise when an ASCII machine is fed some EBCDIC information. For example, the EBCDIC representation of the letter "A" is not the same as the ASCII representation, and the ASCII machine will misinterpret the code. So, when an ASCII computer talks to an EBCDIC computer (i.e. exchanges information), these codes have to be translated from one to the other.

ASIC

Application-Specific Integrated Circuit. This is a small custom-designed electronic component that has been purpose-built to provide functions that are specific to a particular type of application. Thus, what was previously a software function has been integrated with the hardware function. One result is a dramatic improvement in performance. Other primary benefits could be in power, size, production costs, inventory size, maintainability and reliability. The trade-off is that the software version can be easily modified, whereas the hardware version cannot. ASIC technology can provide an alternative way of approaching digital hardware development, avoiding a mess of low-functionality Integrated Circuits (ICs) by designing an equivalent small rugged IC.

assembly language

A computer instruction typically comprises two parts - an operator part and an operand part. The operator part is the actual instruction, for example add, subtract, multiply, shift, mask, search, jump, write, read, and so on. The operand part points to the things that are to be operated upon. For an ADD, for example, the operand part might point to two memory cells containing the numbers that are to be added. A shift instruction has to be told which register to work on.

These instructions are stored in the computer's memory in binary. One such instruction might be "10111000101100101111010100011011". Suppose for example that the first 8 bits were the operator (instruction) and the remainder were the operand. In this particular case, the instruction is "10111000", and this might represent an ADD instruction to the computer. Different instructions would be represented by different bit patterns. The computer understands the meanings of these bit patterns, and, in fact, this is referred to as the computer's "machine language". Ultimately, any program that we write must be translated into the computer's machine language.

Clearly, it would be tedious for us to have to write our programs directly in the computer's machine language. To help with this, an "Assembler" is provided by the computer manufacturer. An Assembler is a program that recognizes our programs written in a language that is easier for us to understand, and translates these into machine-language versions. These "higher-level" languages are called "assembly languages", and there is an almost one-to-one correspondence between assembler language instructions and the machine language instructions. For example, the addition of two numbers might be written as "ADD A,B", and the Assembler would translate this into the binary machine-language equivalent. A (fictitious) assembly-language program could be as follows:

```
A        DEF  42        :Set aside a memory cell - we'll call it "A",
                         and give it the value of 42.
B        DEF  23        :This one we'll call B, value 23.
C        DEF  0         :We'll put the answer in here.
*

START
CLA      ; clear accumulator (ie load it with the valua 0)
ADD A    ; add contents of A to accumulator
ADD B    ; add contents of B to accumulator
STO C    ; store accumulator contents in C
END
```

The assembler would translate these assembly-language instructions into the machine-language equivalents, thereby saving the programmer considerable effort.

This concept is extended further to even higher-level languages. For example, one statement such as "C:=A+B" would be put through what is known as a "compilation" process. The "Compiler" would take the statement and convert it to machine language (although, what normally happens is that the Compiler translates the statements into assembly language, then the Assembler finishes the job).

asynchronous

This word is mostly used with reference to communications between computers, but generally it refers to processes that are not dependent on other processes for their continuation. For example, what this author is doing right now is proceeding asynchronously with what you, the reader, are doing right now. With reference to communications, we talk about asynchronous transmission. In this method, each character (letter, number, special symbol) has some additional information wrapped around it (called start and stop bits) that wakes up the receiver just before the character is transmitted, and tells it when the character has been transmitted. In this way, the sending of successive characters do not have to be synchronized with each other. They can be fired off to the receiver asynchronously.

A/UX

Apple's version of UNIX that runs on the Macintosh.

B

backplane

A hardware device that houses a bus (see "**bus**" below), and into which are slotted circuit boards to enable various devices, or components, to communicate with the computer, and with each other.

bandwidth

Refers to the capacity of a communications line, whether digital (in bits per second) or analogue (in hertz).

baud rate

Often confused with bits per second. However, this is also often the case - i.e. a specification of, say, 9600 baud is actually intended to also imply 9600 bps. The term, though, means the number of times a second that a system (e.g. a data transmission channel) changes state, and for a **binary** channel, 1 baud = 1 bps. For a general channel, however, more than 1 bit could be conveyed for each state change, so that 1 baud could represent several bits per second.

benchmark test

Can be a program, a set of programs, a set of particular database transactions and other things of that nature that can be run on several different computers, or against several different database systems, so as to compare their capabilities in a consistent manner. See "Dhrystone MIPS".

binary

Just means two. The binary number system has 2 digits, 0 and 1. Our familiar decimal number system has 10 digits, 0,1,2,...,9. Another popular one is the hexadecimal number system (referred to as HEX) which has 16 digits, 0,1,...,9,A,B,C,D,E,F. Each hex digit is a nybble (4 bits). The distinction between a **binary** file and an **ASCII** file is that an ASCII file only contains bytes (bit patterns) that represent valid ASCII characters, whereas a binary file contains bytes that could be any bit pattern.

binary patch

It often happens that, between releases of software, significant errors are encountered. These errors won't be fixed until the next release. To overcome this, a "patch" is prepared, where the machine language version of the software (the binary version) is altered, or patched.

bit

This is nearly an acronym. It is short for binary digit (0 or 1). Bits are fundamental to computers because their values can be represented by ON or OFF, or anything that can be in one of two states, and these states can be represented electronically.

bottom-up development

An approach to program or system development in which progress is made by composition of available elements, beginning with the primitive elements and ending when the desired program or system is developed. At each stage, new, more powerful elements are constructed by combining the lower-level elements. These new elements are then combined to create even more powerful elements until the final program or system is created.

In practice, "pure" bottom-up development is not possible. The construction of new elements must always be guided by a look-ahead to the requirements of the eventual program or system, and even then it will often be discovered at a later stage that some earlier construction sequence was inappropriate, leading to an iterative development. Compare "**top-down development**".

Bulletin Board

An electronic notice board used in information networks for the display of general information - e.g. public information systems.

bundled

Refers to a number of products and/or services that come under the one all-inclusive price. Unbundled is the opposite. For example, the cost of maintenance of a system might be unbundled from the purchase price of the system, and a separate agreement would need to be negotiated.

bus

This term is used in a number of specific ways, but generally it is a communications or signal path, usually of limited length, along which are attached a number of "devices" that wish to communicate with each other. Diagrammatically, a bus is represented simply by a straight line with a specific beginning and a specific end, with various devices or components attached along its length.

In a Local Area Network (LAN), the bus would likely be a length of coaxial cable, with devices (e.g. terminal servers) attached at various points (as well as a computer, of course).

Inside a computer, a bus could be a length of conducting material. Devices, or components, could be in the form of circuit boards that fit into slots along the length of the bus. At an even lower level, a bus could be a very short electronic path for passing low level information from one processing unit of the computer to another (e.g. memory addresses).

byte

A byte is a group of bits - usually 8. There are 256 different combinations of bits (bit patterns) in an 8-bit byte - e.g. 01001011, 11111001, 10101011, and so on. These can be used to represent different characters. For example, 0100001 might represent the letter A, 00101100 might be a full stop, and so on. ASCII (actually a seven bit pattern) is a definition of which bit patterns represent which characters. EBCDIC is another one, and they're different. The byte unit is often used with reference to capacity (of memory or disk storage). A Kilobyte (Kb) is one thousand bytes (1,024 actually - see the entry under "Kilo-"), and a Megabyte (Mb) is one million bytes. A Gigabyte is one thousand Megabytes. We also have Terabytes and Petabytes too.

C

C

A language used for programming computers of all sizes as are PASCAL, Ada, COBOL, FORTRAN, PL/1 BASIC and ALGOL. "C" is a so-called 3rd generation language. The UNIX operating system is largely written in "C".

cache

This is typical of a word borrowed from English by the Computing World and used in a different context, thus masking its original meaning. It is supposed to be a hiding place for treasure, provisions, ammunition (French "cacher"). It is used to mean a small piece of high-speed memory in which is placed data (e.g. from a disk) in anticipation of it being required by a program. This anticipatory read-ahead mechanism can provide significant speed improvements.

CAD/CAM/CIM

A CAD/CAM system would provide high performance, high speed interactive graphics functions for draughtsmen, engineers and other technical professionals to enable them to design and draw up plans, and to store, view and modify them easily.

CALS

Computer-Aided Acquisition and Logistics Support. A US DoD initiative that defines standards and procedures for electronically creating, storing and transmitting documents that are traditionally manually exchanged between the Department and its civilian contractors.

CASE

CASE is an acronym for Computer Aided Software Engineering. It also stands for Common Application Service Element, which is an OSI term that is discussed under "OSI". Software Engineering is the process of developing software in a controlled way. It has always been a source of some amazement to the industry that engineers can build magnificent complex physical structures involving the co-operation of dozens of organizations and hundreds of people, but that we cannot seem to be able to build software systems involving only a few organizations and dozens of people. The argument to explain this problem is that software was never "engineered", because there were never any established procedures to follow.

So was born software "engineering". Software engineering attempts to define what is needed to be done, and how to go about doing it. Thus, software engineering "methodologies" were developed. Now we've had software engineering tools for quite a number of years, but these are not computer-aided. These tools often go by the names of their authors e.g. DeMarco, Yourdon & Constantine, Gane & Sarson, Jackson, Martin, Ward, Mellor, Hartley. They are simply attempts to formalize the process of software development to ensure that everything has been covered, and in the desired order.

CASE tools usually embody a structured methodology such as those cited above, but in addition provide such things as automated graphics facilities for producing graphs and diagrams, screen painters and report generators, data dictionaries, extensive reporting facilities, analysis and checking tools, documentation generators and code generators. A number of products for the PC are available today. These tend to be a bit expensive (around the $10,000 mark), but for a large project, they could save ten times that in development costs and costs associated with omissions and the like.

CCITT

Comité Consultatif Internationale de Télégraphique et Téléfonique. This is a committee belonging to the International Telecommunications Union, which itself is a part of the United Nations. Its function is the setting of international communications standards. CCITT members include PTTs (the Public Telephone and Telegraph Utilities of the world such as TELECOM Australia), private companies and scientific and trade organizations.

CD-ROM

Stands for Compact Disc ROM. ROM stands for Read Only Memory. CD-ROM is a way of storing large amounts of information (about 600 Megabytes) in a machine-readable form on a very small disk. There is a rapidly expanding role for CD-ROM in information distribution (e.g. in place of microfiche).

cell relay

A technique for relaying packets based on a fixed packet size. See "MAN".

client/server

Refers to style of computing where an intelligent process requests services from the system, and the system provides these services, but the requesting "client" has no need to know where the services are located. An example might be a user at a PC who creates a document locally and requests the system to store the document in the corporate electronic filing system. Software in the PC arranges to transfer the document to the corporate file server. The user doesn't care where this is, and, in fact, the organization could change to a different file server without the user needing to know. Once filed, the user might want to do some database processing, and locally prepares a transaction and requests the system to process it. Software in the PC interacts with the remote database server to process the transaction. The database server could be an entirely different machine from the file server, and could be interstate or even overseas. The user doesn't care, and is simply a client requesting a service.

co-axial

"co-axial" means to have a common axis. Co-axial cable is a transmission cable having two concentric conductors separated by an insulating material. It achieves the same result as twisted pair cabling, but is much more robust.

CODASYL

Acronym used to describe a database management system that follows guidelines set down by the Conference on Data System Languages.

Command and Control System

The facilities, equipment, communications, procedures and personnel essential to a commander (and staff) for planning, directing and controlling operations of assigned forces pursuant to the missions assigned.

Command Information

Pertinent information and data which is used by a commander and staff for operational decision making and is not dependent on automated assistance for its interpretation. Such information is frequently assessed, collated or otherwise processed before being passed up, along or across the chain of command as a report, or down the chain of command as an order. Command Information includes, but is not necessarily limited to, data such as:

_Operational plans
_Readiness (Status of Forces) information
_Location of forces
_Movement data
_Surveillance data
_Intelligence (including Indications & Warning)
_Order of Battle data

_Infrastructure information
_Logistics data
_Environmental data
_Personnel data
_Contingency plans

Command Support System (CSS)

A computer system designed to assist a commander and staff with the receipt, collation, storage, retrieval, manipulation, dissemination and display of command information. A CSS incorporates the computing equipment and software, and includes data communications integral to the system, along with facilities to provide external access to other systems via the various Defence communications networks.

compiler

A compiler is a piece of software that reads in computer program statements that have been written in a particular programming language (e.g. FORTRAN, COBOL, PASCAL, etc.), then translates these into the equivalent in the machine's own language. See also "assembly language".

Computer Aided Software Engineering

See "CASE".

Computer Supported Collaborative Work (CSCW)

See "groupware".

computer vision

A term used to cover computer recognition of shapes on an assembly line through to robot vision which mimics that of a human. See also "image understanding".

concatenate

Much the same as juxtapose - place things side by side, except that "concatenate" implies that the two parts are now somehow linked together to form a single entity. For example, if we take the text string "SOME" and concatenate it with the text string "HOW" we form the single text string "SOMEHOW".

concurrent programming

A near-synonym for parallel processing. The term is used both to describe the act of creating (writing) a program that contains sections to be executed (run on the computer) in parallel, as well as their actual execution. The understanding is that some subsequent task will proceed after the completion of the concurrent tasks.

COTS

An acronym for Commercial Off-The Shelf equipment, software or systems generally.

CPU

Central Processing Unit. Together with other units such as Input/Output, Arithmetical and Memory, as well as peripheral devices and a set of operating instructions, it forms a computing facility. See also "PC".

CRT

CRT is an acronym for Cathode Ray Tube, that being the technology used to display the characters on the screen. It is often misused to refer to a screen and keyboard computer terminal. More usually, the screen itself is referred to as a VDU (Visual Display Unit), a VDT (Visual Display Terminal) or a Monitor.

Cryptography

Cryptography is the generic name for the processes involved in transforming readable information into unreadable information for transmission over some non-secure medium (e.g. the public telephone system), and transforming the unreadable information back into readable information at the receiver's end. The transformation from readable to unreadable is termed "encryption", and the transformation from unreadable to readable is termed "decryption".

The usual process of encryption and decryption is via "keys". A key is a special number or bit pattern known only to the sender and the receiver. The sender encrypts the information by applying a formula to it, part of the formula involving the secret key. The complexity of the formula, or of the transformation process, will determine the degree to which the sender and receiver can be sure that an eavesdropper will not be able to decipher the encrypted information. For example, the transformation might be simply to add the key to each number of the transmitted data, but this would probably be easy to "crack".

References:

_Rivest, R., A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *CACM*, **21**(2), pp. 120-126, February 1978.

_National Bureau of Standards, Data Encryption Standard (DES), *Federal Information Processing Standard Publication 46*, January 1977.

CSCW

Computer Supported Collaborative Work. See "groupware".

D

data

"Data" is the plural of "datum", but is often treated as singular. Thus, while it is more correct to say "these data", current usage tends towards "this data" - i.e. treating data as singular. This isn't entirely unreasonable anyway, as often we are referring to the **collective set** of numbers, text strings etc. that go to make up the "data".

It is often useful to distinguish between "data" and "information", even though the OED refers to data as "facts or **information**". A simple distinction is that "data" comprises some sort of unprocessed quantities, such as numbers, text strings, readings from sensors or other instruments, whereas "information" comprises quantities derived from these by some process, e.g. through calculations, inferences, transformations and the like.

This explanation is somewhat specious, and open to interpretation. For example, consider a radar system that provides details of aircraft in the vicinity of an airport, these details being fed into a radar data processing computer for interaction with the air traffic controller. Some radar sensors transmit the raw data to the computer, which then goes through a smoothing process before going through a further process in order to present information to the controller. Other sensors have added intelligence, and perform the smoothing calculations before handing the results to the radar computer. Is this now information? It's quite likely that the computer considers it to be raw data which it then processes in order to interact with the air traffic controller. The distinction might be subtle (and arguable), but can be important in certain scientific disciplines.

Database

Database technology has undergone considerable change over the years, but appears to have distilled into three areas. First of all, a database is just a bunch of data. It turns out, though, that if you arrange this bunch in different ways, you can do things more efficiently. The way you arrange it depends on the way you most want to access the data. The central idea is that data can be shared by many users and is important to an organisation. There is benefit therefore in defining its format and enforcing this through standard interfaces which employ appropriate security and integrity measures. Here is a quick description of the three different types. They are **Hierarchical, Network** and **Relational.**

The **HIERARCHICAL** model has nothing to do with the three divisions of angels. It has to do with the organization of things ranked one above the other - like the usual personnel structure of any organization. This is a very simple structure, and is used by IBM's IMS and DL1 products. For example, in a database of customers and orders, records are set out by customer, and under each customer is the list of that customer's orders. This makes it easy to find out all of a given customer's current orders, but difficult to get a list of all the customers who currently have a particular item on order. Clearly, you wouldn't set things out this way if you wanted to do that sort of thing a lot.

The **NETWORK** model attempts to overcome the problems of the hierarchical model by introducing "link records", or pointers back to the owning entities. For example, if we have lots of employees with lots of different skills, we would set up one file of employees, and one file of skills. Each employee record would contain links, or pointers, to the skills that the employee has, and each skill would contain pointers to the employees that have that skill. This way, it's easy to list an employee's skills, and to list the employees with a skill (a skill's employees, if you like). Incidentally, the network model is the approach adopted by CODASYL.

There is, in fact, a one-to-one correspondence between the network model and the **RELATIONAL** model. The relational model refines the network model by the removal of redundant information. The network model often holds the same information in more than one file. The result in the relational model is a series of tables (called relations - somewhat misleadingly) that define the data, without defining items more than once. For example, a "customer" relation might be a table of customer number, name and address. A "product" relation might be a table of product number and corresponding product description. The "glue" that binds them together might be a table that relates a customer to a product number and an order quantity.

Data Fusion

Refers to the joint processing of data from multiple sources, each providing data about the same objects of interest, so as to extract more accurate, reliable and comprehensive information about the objects. See also "Information Fusion" and "Sensor Fusion".

DBMS

An acronym for DataBase Management System.

DCE

Data Circuit-terminating Equipment. In a network, this is the piece of equipment (e.g. a modem) that provides access to the network. The customer's DTE (Data Terminal Equipment - e.g. PC) connects to the DCE for network access.

D-connector

Simply the shape of a connector at the physical end of a cable. D-connectors can be male or female, and may have differing numbers of pin connections depending on the specific usage (25 and 9 are common).

Debug

This little word describes the activity of bug removal. A bug is an error in somebody's computer program that does things like send you a bill for 5 cents with a minimum payment of 1 cent, when the software specification stated that if the account was less than $3, wait until NEXT month to send the bill. Why waste time and money? Some might call this an error of omission, but it's still a bug, because the software is doing something it's not supposed to do.

Decision Aids

Simply tools to assist with the decision-making process.

Decision Support Systems

Decision Support Systems offer a mix of problem-solving, analytical capabilities ranging from basic spreadsheet functions through financial modelling to sophisticated credit and investment analysis applications. The software is used by analysts and managers for budgeting, strategic planning and forecasting. High-level packages are capable of complex data analysis applications like goal seeking, risk analysis, time series and trends.

decryption

See "Cryptography"

DESINE

Defence EDP Systems Integrated Network Environment. The DESINE project attempts to set up an integrated computer and communications architecture for administrative systems "across the range" (i.e. from microcomputer systems to mainframe systems). The contract was awarded to IBM for a 5-year period.

Dhrystone MIPS

A general-purpose counterpart to the Whetstone benchmark. Whereas the Whetstone benchmark emphasises floating-point performance, Dhrystone emphasises integer arithmetic and general data processing. It is written in C, and tests general CPU performance. Because of its small program size and limited data space usage, it is inordinately optimized by systems with cache memories (a problem supposedly corrected by later versions).

diagnostic

This is generally a self-testing routine internal to the computer which produces a message from the computer to you to tell you that something has gone wrong. Hopefully it is informative, so as to help you to diagnose the problem. Sometimes, however, these messages are excruciatingly cryptic, for example "There's an S222 in your DCB spec. causing an E37".

digitized voice

A process where sound is sampled, usually at 8,000 times a second, and the pitch of each sample recorded as a number, usually in the range 0-255 as this fits into one byte (8 bits). Thus, the sound is now translated to a series of numbers - it has been "digitized".

Note that this digitization process introduces errors. Firstly, not all of the voice signal over each second is sampled - suppose only 8,000 samples are taken. This could be increased, but there needs to be a trade-off between the reproduced quality and the amount of data that is needed to produce the quality. Secondly, each sample allocates an integer value to the sound (between 0 and 255). The sound sample might actually be correctly digitized to, say, 123.345, but it is recorded as 123. At the other end, only the pitch that relates to 123 is generated. These errors generate distortion, and this is measured in **Quantization Distortion Units**, or QDUs. Carriers usually specify the maximum number of QDUs allowable through networks that connect to the public network so as to ensure a given level of reproduced voice quality.

disk

Circular platters coated with a magnetic material that can be used to store computer data (bits). The term "floppy disk" is used for a flexible single-platter disk contained within a protective sleeve. It is "removable" in the sense that it can be taken out of the computer and stored elsewhere. The term "floppy disk" is also used to refer to the newer style 3_ inch disks, even though they are far from floppy. The intention is to distinguish between these "removable" types and the "fixed" types that are referred to as "hard disks". A hard disk is a platter (or several platters in a stack) that are rigid and sealed inside a container, the whole thing being fixed inside the PC (although some removable versions are becoming available). They are sometimes called "Winchester" disks from the name of the (IBM) project that developed the technology. They can hold much more information than floppies. For larger computers, all the disks are usually "hard" disks as defined above, but are usually housed in their own separate enclosures. Further, the platter stacks (disk volumes) can usually be removed as a unit and stored elsewhere.

DQDB

Distributed Queue Dual Bus. A fast packet switching technology. See "MAN".

DTE

Data Terminal Equipment. Equipment (e.g. a PC or a communications processor) that connects to DCE for network access.

E

EDI

Electronic Data Interchange. EDI can be described as "paperless trading". The basic principle of EDI is that computer-generated trading documents (e.g. orders, invoices) are transmitted directly to a company's trading partner's computer(s) across a telecommunications network. The benefits include time saving (bypassing the postal system), cost saving (no duplicate data entry and attendant transcription errors), stock savings (more immediate inventory control), cash flow improvements (improved speed of trade) and marketing advantages (through closer relationships with trading partners).

A typical EDI Service provides an EDI Server computer on which is implemented EDI Mailboxes and EDI Translation Services. The EDI Translation Services convert custom-designed documents (e.g. a company's own invoice) to an industry-standard format, which is then stored on the EDI Server, and sent to the requested EDI mailboxes (within the same server, or, for a distributed or international service, to a remote server). The trading partners then access their EDI mailboxes to retrieve the documents.

Because the EDI server must cater for a wide variety of interconnections to company computer systems, the communication is generally effected using an X.25 packet-switching service, as most computer manufacturers provide some form of X.25 capability. EDI standards are directed to the format and control of documents, not to the method of transmission, but communication standards are being developed, particularly in relation to using the X.400 E-mail standards for transmitting EDI documents.

Many industries have developed their own EDI document standards. Examples include AIAG (Automotive Industry Action Group), WINS (Warehouse Information Network Standards) and CIDX (Chemical Industry Document Exchange). Standards are now being rationalized through the United Nations ISO (International Standards Organization) EDIFACT (EDI for Administration, Commerce and Transportation) standards. Industry derivatives from the ANSI X12 standards are still the most commonly used, but they will be bridged across to EDIFACT as approvals are obtained.

EDP

Just an acronym for Electronic Data Processing. The EDP department is the department that runs the computer. Sometimes it's called ADP (Automatic Data Processing).

EFT

Electronic Funds Transfer, the generic term for sending payment instructions over a computer network.

EFTPOS

Electronic Funds Transfer at Point Of Sale - the use of credit or debit cards over a network to facilitate payments for (generally) retail sales.

EGA

Enhanced Graphics Adaptor. Unfortunately, not all computers are capable of displaying graphics, although these days, most are. The quality of the graphics depends on the quality of the graphics screen driver and the quality of the screen. EGA is one of several kinds of special screen driver. Prior to EGA was CGA (Colour Graphics Adaptor) that was used in the earlier PCs. VGA (Virtual Graphics Adaptor) is the next version after EGA.

electronic mail (E-mail)

A means by which users and groups of users can exchange information over computer communications networks. The information can be text, spreadsheets, graphics and so on. Typically, a user prepares a mail message on a computer, and submits this message to the E-mail system (which is a suite of computer programs linked over a communications network). An example of a mail message might be a piece of text like "Joe, attached is the end-of-year financial statement spreadsheet under Lotus 1-2-3", plus an attached spreadsheet file. The E-mail system then transmits the message and attachment (which is just a computer file) over the network to the recipient. The recipient might receive a "mail-arrived" message, connect up to the E-mail system, and retrieve the message plus spreadsheet file. This is a "store-and-forward" system.

Over the years, quite a number of proprietary E-mail systems have been developed, with the concomitant problem of different E-mail systems being unable to exchange documents. This has prompted the development of an international standard for E-mail systems, and this is discussed under the entry "X.400". A system that predates X.400 is the Simple Mail Transfer Protocol (SMTP) which currently links more than 500,000 computers on the US Defence Advance Research Projects Agency's (DARPA) Internet.

embedded systems

An "embedded system" is one which forms an integral, dedicated manufactured part of some larger system. This is usually a computer that is controlling some part of the larger system, for example in a navigation system. A more homely example would be the computer that controls the cycles that you have selected on your dishwasher. The dishwashing process is the larger system, involving plumbing components, electrical components, mechanical components plus the embedded cycle control system.

encryption

See "Cryptography".

ergonomics

See "Human Factors".

Ethernet

A technique used to connect computers into a Local Area Network. Sometimes referred to as DIX (Digital, Intel, Xerox, who invented the thing). Often referred to as IEEE 802.3, that being a related standard, based on the Ethernet principles. This technology operates at Layer 2 of the OSI model. It is a so-called "broadcast" technique, in that all stations on the LAN receive all the packets, but ignore the ones that do not contain their network address.

Ethernet cabling comes in two basic forms (plus twisted pair lately), that of thick and thin Ethernet. Thick Ethernet is the original cabling medium, and comprises thick pieces of co-axial cable that can be configured in 500-metre segments. It is inflexible to handle. Thin Ethernet uses a much thinner, flexible cable, but is restricted to configurations that are essentially star configurations (useful for one floor of a building). A typical Ethernet network in a building would use thick cable in the riser, and thin cable across each floor.

execute

Refers to the actual running of a computer program by the computer, as distinct from the preparatory phases of writing and compiling the program.

Expert Systems

An Expert System is a computer program that contains the kind of knowledge that an expert would use to solve a problem in a particular area of expertise. This expertise knowledge forms the knowledge base which can then be queried by a user to help solve the type of problems that only an expert in that particular area could solve. A feature of Expert Systems is that they can explain how a particular solution to a problem was derived (in contrast to neural networks, for example, which cannot). This can help the users to make up their own minds about the validity of the solution offered by the expert system.

An Expert System comprises three parts:
_Facts;
_Knowledge Base; and
_Inference Engine.

A fact is some information that the expert system elicits about the state of the current situation relating to the problem that has to be solved. The knowledge base contains the knowledge that the human expert has provided about a particular domain, or field of knowledge. This is typically represented in a rule format, although other representations may be used.

A rule consists of two parts - a condition and an action. For example, in the rule "**IF** the temperature exceeds 45 degrees **THEN** notify the fire brigade", the condition is the temperature exceeding 45 degrees, and the action is to notify the fire brigade.

The part of the system that decides which rule is applicable to the current situation is called the "inference engine". The inference engine reasons about a situation based on the facts and the rules contained in the knowledge base. The most common methods of reasoning are called forward and backward chaining.

Forward chaining means that the inference engine will try to match the facts to the **condition** part of all of its rules, to see what actions should be taken. Backward chaining means that the inference engine will find all of its rules that match a particular **action**, and will then try to match the conditions of these rules to the facts.

Expert Systems can be built by using expert system **shells**. These are commercial software products that contain the inference engine. It is left up to the user to provide the knowledge base and the facts for the inference engine to reason about. It is often very difficult to get experts to describe how they go about solving particular problems, so knowledge bases are sometimes built with the help of Knowledge Engineers. It is the job of the Knowledge Engineer to elicit the knowledge from the expert, and structure it in a way that it can be reasoned about by the inference engine.

F

facsimile

A facsimile machine (or "fax", as it is often called) operates by scanning a document from left to right beginning at the top of the document, and gradually moving down the page. Each "line" so scanned creates a string of bits, the 0s representing white space, and the 1s representing dark space. Some machines actually encode scales of grey into bit strings, but the principle is still the same. These strings of bits are then transmitted over the telephone network (after being modulated by a modem) to the receiving fax machine, which re-creates the image by forming black spots on the paper where the 1-bits are.

The scanning process creates quite large bit strings, so these are compressed before being sent over the network. The most common compression technique is the Huffman code technique. Here, strings of 0s (i.e. lengths of white space) and strings of 1s (i.e. lengths of black space) are assigned specific codes according to their lengths. For example, a string of 60 0s is assigned the code "01001011". This technique works well for most documents, because of the frequent large blocks of white and black space.

FDDI

FDDI is an acronym for Fibre Distributed Data Interface - it is a standard for setting up a fibre optic LAN in a ring topology with a circumference of up to 100 kilometres, and it runs at 100 Mbs (Megabits per second). Now, there is an IBM developed 4 Mbps Token Ring standard, a 10 Mbps standard used by Digital systems (and others, of course) (Ethernet/802.3), and a newly announced 16 Mbps Token Ring offering by IBM. Where do they all fit?

The 16 Mbps Token Ring would seem to fit neatly as a backbone network for the interconnection of existing 4 Mbps Token Rings. A lot of people would say that this is not nearly enough. In fact, that's what FDDI would be much better suited for. FDDI incorporates a token ring technology, and runs at 100 Mbps. The initial industry thrust, however was to cater for the Ethernet type LANs, but Token Ring products are now becoming available.

There is a view that Token Ring connections will overtake Ethernet connections in the next couple of years, and the major reason has to do with PCs. PC connections are the fastest growing segment of the networking marketplace, and IBM accounts for about 35% of all PC shipments. From this perspective alone it could be argued that if you want to network them, you'll probably prefer the IBM solution.

Also, have you heard of the OTF? This is the Open Token Foundation. OTF was formed by 3COM, and is a group of vendor organizations. OTF's goal is to promote awareness of Token Ring, and to ensure agreement on implementation and technical issues.

Now, there are all sorts of arguments and counter arguments about Ethernet and Token Ring, and this author is not going to be able to resolve them. Suffice it to say that IBM seems to have timed its 16 Mbps products rather strategically.

fibre optics

This is an efficient way of conveying binary data (which is what computers understand, and is just a string of 1s and 0s, or ONs and OFFs). Fibre optic cables transmit light, and the light can be switched on and off at remarkable speeds, thereby transferring the data.

file

A generic term for a collection of information, usually stored on a disk. There are all sorts of files - sequential, random, inverted, indexed sequential.

firmware

This is a cross between hardware and software. It refers to the sequence of steps that a computer must go through in order to perform, or execute, an instruction. By allowing these sequences to be changed, the characteristics of instructions can be changed. The instructions, therefore, are not "hard-wired" into the computer, because their operation can be modified. Modifying the behaviour of computer instructions, however, is a process that is not available to the general software programmer, who is generally only interested in the outcome of each instruction, not on how it produces the outcome. Hence the term "firmware". The sequences of steps that go to make up computer instructions are also referred to as "microcode".

flowchart

A flowchart is just a lot of boxes drawn on a piece of paper denoting "decision analysis" points, connected by lines and arrows, depicting the flow of control of a computer program (or other process).

formal methods

This is a term that covers the activities needed to develop systems in a formal and rigorous manner, such that the resultant systems can be formally shown (proven) to operate according to the specifications. See also "Z" and "VDM".

FORTRAN

FORTRAN is an acronym for **Formula Translation**. It was probably the first and most widely used so-called 3rd generation programming language in the world. It remains in everyday use, being the official language of many large engineering organizations (for example Australian Construction Services).

frame relay

A fast packet switching technology based on variable-length packets. See "MAN".

Front-End

This usually refers to a small computer that, on one side connects directly to the network (communication lines), and on the other side connects to the main computer over what is called a channel (high speed communications line). The front-end off-loads all the mundane tedious communications processing from the main computer.

The term is also widely used to refer to some form of pre-processing. An example would be the preparation of a transaction at the point of creation (e.g. by a bank teller) prior to sending the transaction request over the communications network for processing by the host computer. The transaction preparation would be a front-end to the transaction processing.

full-duplex

Refers to a communication path where both parties can send and receive, at the same time. With half-duplex, they can both send and receive, but only one at a time. With simplex, only one party can send.

G

Gateway

A process (hardware, software or a combination) of allowing communication and interaction between two systems that have different architectures. Gateways generally have to provide some conversion services so as to map between the different architectures. Gateways can be used at various levels of the OSI stack. For example, an X.400 gateway would be an application layer product that permitted interchange between your proprietary electronic mail system and the standardized X.400 world. An X.25 gateway would be a network layer product that might be used, say, to interconnect two Ethernet LANs over a packet switching WAN. A gateway is different to a **bridge,** in that it involves some translation process, whereas a bridge simply filters packets without any translation.

Geographical Information System (GIS)

A system (i.e. a collection of hardware, software, tools, processes) comprising a database and facilities for its manipulation (editing, updating, displaying). The database is particularly oriented towards land-related information, and so must be able to represent two- and possibly three-dimensional information, as well as the traditional text and numbers. Similarly, the database manipulation facilities must be oriented towards this type of data.

gigabyte

The prefix giga- means 10 raised to the 9th power - that is, a thousand million, which is a US billion.

GIS

See "Geographical Information System".

GOSIP

Government Open Systems Interconnection Profile. This is a specification of a Government-defined subset of international standards for data communications between proprietary and open systems. The entry under "OSI" discusses GOSIP, and places it into the OSI perspective.

graphics

The process of portraying pictures or representations other than characters (letters, numbers, special characters) on your screen, printer, plotter or whatever. Most computers come with a character-generator as standard. This makes it easy for the software to put text on the screen. To draw pictures, however, you have to circumvent the character generator, and use a line and curve generator. This usually comes in the form of a special board to slot into your PC.

groupware

Essentially, this is software that supports workgroups. It is often referred to as "computer supported collaborative work - CSCW". The evolving workgroup environment comprises people interacting with each other over computer networks. The emphasis of groupware is in user-to-user interactions. This stage of computing has been variously defined. Some of the definitions currently being tossed about are along the following lines:

_Specialized computer aids designed for use by collaborative workgroups which are usually small and project-oriented;

_Software applications that are designed to support small or narrow-focussed groups, especially software that recognizes the different rôles that the users of the application have;

_Computer-based systems that support two or more users engaged in a common task and provide users with an interface to a shared environment; and

_Computer-based tools that can be used by workgroups to facilitate the exchange and sharing of information including user adaptations of individual tools to support group functionality.

Common to the theme of all definitions is the merging of computing and communications and the creation of common tasks and shared environments or spaces where there are more than two users. Typical groupware would include electronic mail, calendaring, bulletin boards, document indexing and retrieval, task and project management.

H

hacker

Highly skilled but possibly undisciplined programmer. This was the original and preferred definition now supplanted in popular usage, especially the news media, by the following:
A person (or even a computer) who tries all sorts of ways of logging in to your computer, without initially having an account or password. A hacker hacks away to get into your system, then hacks away at trying to read all your files - or delete them - or leak them to the press. These attempts are considered to be intellectual challenges to hackers, and their efforts lead to manufacturers producing more secure and robust systems.

handshaking

Two devices talking to each other before sending any information, to make sure they're operating according to the same rules. The devices can be hardware devices or software (e.g. one program conversing with another program in another computer).

hardcopy

Usually means a printout of your data onto paper.

hard disk

Not a floppy, or flexible, disk. It is usually housed inside a shockproof case, and is capable of holding much more data than your floppy, with considerably faster access. It is the recording medium which is not usually removable. See also "disk".

hardware

The plastic, metal, wires, glass, components, printed circuits, electronics, knobs, beepers, cables and so on that make up the actual computer. The software comprises the programs (computer instructions) that make it actually do anything. There is another operational element called "firmware". This really is a collection of software sequences that direct how the machine language instructions are to behave. You can see, therefore, that this is at a lower level, even, than the Assembly Language.

HCI

Human-Computer Interface (or Interaction). Please refer to the explanation under "Human Factors".

Hertz

A unit for measuring frequency, and meaning "cycles per second" (named after its developer, a German physicist named Heinrich Hertz).

heuristic

A rule of thumb. An approach expected, but not guaranteed, to yield satisfactory results or a rule that, when applied, isn't guaranteed to provide you with a definitive answer. It could be thought of as a "global estimate". The word actually means "serving to find out", and is specifically applied to a form of education where the pupil is encouraged to find things out for himself. Used today in the area of Artificial Intelligence as well as other areas of computing.

HF

An acronym for High Frequency. This is a band in the radio frequency spectrum (3MHz - 30Mhz) generally reserved for shortwave communications. Signals in this band are at a higher frequency than the medium frequency band which contains local broadcast amplitude modulation stations.

hidden file

A file that's on your system, but that usually doesn't show up in the directory listing, and that can't be accessed by the operating system commands (e.g. COPY and DEL). They can only be accessed by programming. Tools (utility programs) are usually available for removing the "hidden" attribute, so that the file can be accessed by normal operating system commands.

hierarchy

A body of persons or things ranked in grades, orders or classes, one above another.

host

This generally means the computer that your terminal is attached to. Essentially, it is the computer that is the host of all of your software and services. It is generally NOT your PC. If your PC is linked to a larger computer, then the larger computer is called the host.

hotkey

The action of flipping between two or more programs that are available at the same time. The hotkey flips from one to the other, and starts up where you left off.

housekeeping

A term used to refer to the "chores" involved in looking after a system - things like taking backups and updating to new versions of software.

Human-Computer Interface (HCI)

See "Human Factors".

Human Factors

Refers to the interaction between people and machines, how they are affected by the interaction, and how to facilitate the interaction (i.e. make the interaction simple to use, friendly and meaningful). Studies in Human Factors are not restricted to interactions with computers, although this is a major area. For example, aircrew vision issues are part of this subject (distance judgement with night vision goggles, approach path visual guidance, aircrew visual performance). It is safe to say, however, that Human Factors Studies in a more commercially oriented field relate to effective design of human-computer interaction, to training and to performance under stress, and to development of technology for the improvement of human-computer interaction (including display technology).

The aims of Human Factors Studies are to harness and expand human productivity and creativity in the workplace and elsewhere by matching the demands of environments, products, systems and tasks to the physical and mental characteristics of those people who use or operate within them.

The study of human physical and mental attributes, including limitations, has long been recognized as a major contributor to product design. This acceptance is based upon the realization that no matter how brilliant the technical concept and execution of a product or system, it will fail if the users for whom it is intended cannot or will not use it. Human Factors is therefore closely associated with a user-centred approach to product or system design.

The word "ergonomics" (derived from the Greek *ergon* meaning "work") is usually considered as synonymous. Historically "ergonomics" was seen as emphasizing the physical aspects of system design. This perception has not been eradicated, with the result that many of those working in fields associated with Information Technology find it necessary to prefix "ergonomics" with qualifying terms (e.g. "software ergonomics").

In the United States, the term used is "Human Factors", while Europeans tended in the past to use "Ergonomics". For the reasons given above, and because the term "Human Factors" emphasizes the human in the system, "Human Factors" has become (and is becoming) more generally acceptable.

"Human Factors" is a multi-disciplinary applied science. Practitioners may be designers, engineers, linguists, medical practitioners, physical ergonomists, physiologists, software ergonomists, and so on.

hypermedia

See "multi-media"

hypertext

Hypertext is a new area that is gaining wider and wider use. It is new in the sense that computer systems are appearing that provide hypertext capabilities, but it is far from new in concept. You can probably gather that the term implies "something more than" text - something above straight text - a sort of "Metatext". The term was coined by one Ted Nelson in the 1960s (Nelson, T.H. "Getting it out of our System", Information Retrieval, Thompson books, 1967, 191-210). When explaining the hypertext concept to people, Nelson states that relatively few people grasp the concept straight away. What they don't understand is that writers write better if they don't have to do it in a sequential manner, and that readers read better if *they* don't have to read sequentially. Nelson knew that he had the *right* word with hypertext. "Hyper" to most scientists and mathematicians means *extended* and *generalized*, as in hyperspace, hyperdimensional, hypercube, hypersphere and even hyperchess. Thus, *hypertext* would clearly be the extended generalized form of writing.

Although Nelson coined the word, the concept can be traced back to one Vanevar Bush's 1945 description of "the memex":

> "A device in which an individual stores his books, records and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory."

Bush's concept of data storage had an associative structure that closely modelled the structure of human memory:

> "The human mind operates by association. With one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts, in accordance with some intricate web of trails carried by the cells of the brain. Selection by association, rather than indexing, may yet be mechanized. One cannot hope to equal the speed and flexibility with

which the mind follows an associative trail, but it should be possible to beat the mind decisively in regard to the permanence and clarity of the items resurrected from storage."

In other words, we've got good memories, but they're short. The first serious attempt to build a memex did not occur until 1968, when one Doug Engelbart conducted a live demonstration of his Augment system in which he worked collaboratively on a hypertext document with a colleague 800 kilometres away. Since then, interest has grown steadily, but it is only recently that Hypertext is really coming to the fore. One reason was Apple's introduction of the Hypercard, and the corresponding aggressive promotion of it. Another was Hypertext '87, the first major conference devoted to hypertext.

Now, all of this still doesn't explain what hypertext is. Basically, it is an approach to information management in which data is stored in a network of nodes connected by links. Nodes can contain text, graphics, audio, video, as well as source code or other forms of data. Each node can be thought of as analogous to a short section of an encyclopaedia article, or perhaps a graphic image with a short explanation. The links join these sections to one another to form the article as a whole, and to join the articles to form the entire hypertext document. These links represent paths to other bits of information associated with a particular area of the document. Thus, you can get to anywhere in the document from anywhere by following (selecting) an association path.

Hypertext documents are much more flexible than conventional documents, and we can expect to see more and more hypertext systems coming on the market.

Hz

See entry under **Hertz**

I

icon

In computing, an icon is a small pictorial representation on a VDU screen that suggests an action that an operator would make. For example, a picture of a Filing Cabinet on a screen could represent the computer's data base. A wastebasket could represent the process of throwing something away (deleting a file, maybe). The user "points" to the icon using a mouse, then presses a mouse button so as to select the action represented by the icon. Working with icon-based systems is generally easier, or "friendlier", than working on systems that accept line-by-line commands, and can significantly improve individual productivity.

image

An image is a computer representation of a picture. Images are displayed on a computer screen by representing the screen in memory as a fine grid, or matrix, of values. Each point on the grid is referred to as a pixel (picture element), and the value of the pixel determines the intensity and colour (or grey scale value) of the corresponding point on the screen. In its simplest form, a pixel is a single number with a value of 0 (for white) or 1 (for black). A more complex pixel could be a vector (r,g,b) of three numbers, representing intensities of red, green and blue, respectively. A horizontal row of pixels is referred to as a "raster".

image processing

"image processing" concerns the technical manipulation of images. The term "imaging" is generally used in the commercial sense, and is discussed in a separate entry. There are a number of terms associated with image processing:

_An *image operator* or *transform* is a function that takes an image as input, and produces a transformed image as output. For example, this might be to rotate the image by a specific amount (number of degrees).

_A *binary image* is one in which each pixel has a value of 0 (white) or 1 (black).

_*Thresholding* is the generation of a binary image from a grey scale image by assigning 0 to all pixels with a value less than a certain intensity (or threshold), and 1 to the remainder.
_*Image smoothing* is any image operation that simplifies the input image.

_*Image compression* is an image operation that transforms the image into one that requires less space, but which preserves all or most of the image information.

_*Image restoration* is the restoration of a degraded image to its original form, or to an image judged "better" in some sense.

_*Image enhancement* is any image operation that improves the detectability of objects or features in the image.

_*Image segmentation* is the partitioning of an image into subsets, called *regions*.

_Image *texture* refers to the distribution of pixel intensities within a given region of pixels.

_*Image morphology* is the study of shapes of regions in an image.

image understanding

Whereas "image processing" concerns the technical manipulation of images, "image understanding" concerns the ability of a computer to infer meaning from the image, just as a human would - e.g. to recognize and classify objects in the image. This is essentially the same as the difference in text processing between syntax (how a sentence is constructed) and semantics (what the sentence means).

Image exploitation is the extraction of information from an image. Exploitation involves the analyst in *image interpretation* or *image evaluation,* the process of going from raw image to a symbolic representation in which the important objects and their relative locations are determined.

Image understanding is the wider ability to go beyond the information in the image itself to infer such things as sun angle from shadows.

imaging

In a commercial environment, "imaging" concerns the processing of still pictures in a batch or store-and-forward system. Therefore, imaging is not video-conferencing. Video describes moving pictures in real-time. A major thrust of imaging is to reduce paper handling and paper storage, but it also has the potential to increase handling efficiencies, by allowing simultaneous accesses to the same image. Imaging is also not the same as graphics. There is a basic distinction, and that is that images are **captured** whereas graphics are **created**, or generated.

It is staggering to think that, today, about 95% of all information used in most enterprises is **non-coded**. Here's our first piece of terminology. In coded data, each unit of information (usually a byte or a word in a computer) has a meaning, e.g. a letter of the alphabet, or a number.

Non-coded information isn't like that. For example, a captured image is just a lot of bytes (or bits even) representing light levels, and these convey no meaning to a computer in terms of interpreting what the image represents. A scanned piece of text is a picture of that text, and this has two important consequences.

One is that it requires about 10 times as much storage to store a picture of the text as it does to store the text in coded form. The other is that you can't revise the text with a word processor. Word processors need to work on bytes of interpretable data - i.e. they expect each byte to be the computer representation of the characters of the text. One could run a scanned image through an OCR (Optical Character Reader), thereby converting it to text for interpretation by traditional text editors.

This leads to our next piece of jargon, and that is "**vision**". Vision is a term used to mean that one (a computer, for example) can not only see and display an image, but also interpret and understand the image in some way. An OCR, for example, is a means of understanding an image. Another more advanced example would be that of a robot viewing a scene through a television camera, then interpreting the image to locate a part in an assembly line.

A typical use of an image system to reduce paper handling and significantly improve service could be in an insurance business. Claims would be submitted by the claimants and immediately digitized by a scanner or a camera. This digital image is then indexed. At this stage, the image would be held on magnetic disk. Magnetic disk access speeds still outperform optical disk access speeds.

From this point on, this little piece of electronic paper can whizz around from pillar to post at electronic speeds. More than one person can be operating on the claim at the same time. Historical documents can be retrieved from optical storage (used for archiving because of its immense capacity). Documents never leave the system, eliminating the "out-of-file" problem. Physical storage space is reduced to a minimum. No more huge rooms full of movable filing cabinets. There are huge savings to be made here as well as productivity gains and customer satisfaction.
From a user's point of view, the introduction of an imaging application is certainly going to change the way we work, and the way we respond to customers (presumably this will provide a superior service). From a computing point of view, there is a significant change as well. We will need new technology and more capacity.

Consider that a page of text stored as text requires about 3Kb of storage. This would be, say, an A4 page with a few paragraphs. As a rule of thumb, that same page stored in image form needs about 25Kb, and that's at a resolution of 200 dots per inch. At 300dpi it's about 40Kb, and that's for a common-or-garden business letter. An insurance form would probably need double that storage. For engineering work, an E-size page (33"x44") can range in requirement from 250Kb to 500Kb.

There are, therefore, significant requirements for compression, storage, display and transmission.

There are at present two classes of compression - so called "lossless" and "lossy". In lossless compression, the decompressed image is exactly the same as the original. In lossy compression, the decompressed image is almost the same. Facsimiles use a lossy form of compression. A faxed document can be hard to read, depending on the size of the text. Generally, lossless compression only realizes a compression ratio of 2 or 3 to 1, whereas lossy compression can yield 20 to 1.

Storage requirements will need to be different. The class of storage becomes important in image processing - short-term, near-term and long-term. Magnetic disk is the most common form of short-term storage. It is fast and reusable. On-line juke boxes of optical disk platters are suitable for near-term storage. Access times of seconds are achievable, and the storage capacity is in the gigabyte range. Long-term storage is data that is archived, and is therefore off-line, and requires an operator to load up the platters.

Image display requires graphical display devices. Dumb terminals won't do.

Finally, communications bandwidth becomes critical. It's not just that there is more data, but that there is more data stored electronically, and we are now operating our office in a different way, and need to be able to whiz these documents from pillar to post very rapidly. This becomes even more critical if the imaging application has to be distributed over a wide area, so that WAN design is affected.

In summary, imaging applications are a new way of doing business. They are changing the way we do our work, the way we can respond to customers and clients, and the way we set up our computing environments.

information

See "data" for a discussion on the difference between "data" and "information".

Information Fusion

Refers to the joint processing of separate collections of information, possibly of different types, and from different sources (e.g. processed sensor data and/or intelligence observations), but concerning the same objects of interest. The objective of this joint processing is to obtain more accurate, reliable and comprehensive information upon which to base decisions about the objects of interest.

Information Systems (IS)

A generic term for computer systems (computers, networks and application programs) within an organization.

intelligent terminal

An intelligent terminal is a computer terminal that is actually a small computer, and can process the terminal user's input (up to a point - e.g. edit checks) without having to send every keystroke to the host computer. A lot of intelligent terminals are PCs these days.

integer

A whole number - i.e. without a fractional part - e.g. 5, -23, 64, etc. Integers are useful because the computer logic (electronics) needed to manipulate them is a whole lot simpler and faster than that needed to handle numbers with fractional parts.

integrated

A combination of hardware and software is called integrated when all the constituent elements know about the workings of the others, and they work together to perform a common function in a unified way. An example might be a collection of programs to provide office automation. The collection would be referred to as "integrated" if the various components worked harmoniously, without cumbersome format conversions, or multiple step processes - for example, being able to incorporate a spreadsheet table directly into a word processing document, or a business graphics histogram similarly.

interface or standard interface

A point of interconnection between two systems or parts of systems, e.g. that between a processor and its peripherals (e.g. disk drives). An interface may be classed as "standard" on the basis of manufacturer, industry or international usage. The Centronics parallel interface for printers and the RS232 serial interface are common standard interfaces.

interoperability

This term is used extensively to refer to the ability of two or more computer systems to exchange information, even though the systems run different operating systems. The term "exchange information" can mean file transfer, but it can also mean electronic mail, users on one system logging on to the other systems, but even further, it means the integration of applications over different fields. For example, events that take place in the supply system (e.g. ship mortars to Northern Territory) can automatically cause other events to take place in the manufacturing (e.g. arrange to get more mortars) and finance systems (e.g. bill someone for all those mortars). This ability is often implemented by the disparate systems running products that conform to internationally recognized standards and protocols (e.g. FTAM for file transfer and X.400 for electronic mail).

More formally, it is the ability of systems, units or forces to provide services to, and accept services from, other systems, units or forces, and to use the services so exchanged to enable them to operate effectively together. For Command Support Systems, interoperability may be further defined as the ability to exchange selected data between operational systems resulting in an equivalent understanding by all participants in the data exchange. Interoperability in this sense, therefore, refers to data exchange between operators; capability refers to data exchange between machines.

Interpreter

This is a program that reads a program that you have written, but in its normal form - i.e. a text file. The Interpreter then examines the text, works out what you want the computer to do, then makes the computer do it - right there on the spot. This is different to the two-step process of passing your text-file program through a Compiler, which creates a machine language version of your program (in a separate file), and then running that machine language version. Interpreters are great for program development, but they're slow. Most BASIC languages available on Micros are Interpreters.

interrupt

Refers to the temporary suspension of a running computer program because of some external or extraneous event. The computer immediately transfers control to another (usually memory-resident) program to handle the situation that caused the interrupt. When it is finished, the computer resumes its interrupted program from where it left off. Interrupts can be an integral part of the application (e.g. to sample radar co-ordinates, say 50 times a second), or can be extraneous (e.g. a temperature or a power warning).

I/O

Short for Input/Output. Generally refers to the processes of reading information into a computer from an external medium (e.g. a disk) and writing information out from a computer to an external medium (e.g. a printer).

ISDN

Integrated Services Digital Network. ISDN is a switched digital network providing channels of 64Kbps in accordance with CCITT standards. 64Kbps is a reasonably high capacity (or "bandwidth" as it is called), and can be used successfully to bridge LANs. An important aspect of ISDN is that it is switched. This means that the 64Kbps connection can be set up on demand, so there is no need to pay for a 24 hour/day connection.

Telecom provides ISDN services in one of two forms called Basic Rate Access (BRA) and Primary Rate Access (PRA). These are also referred to as the Microlink and Macrolink services, respectively.

BRA is referred to as 2B+D because it comprises 2 "Bearer" channels of 64Kbps and one "Data" channel of 16Kbps through a single connection. This terminology is somewhat confusing as the Data channel is really intended as a common signalling channel to control the operation of the Bearer channels (so-called out-of-band signalling). The Bearer channels are intended to carry the data, and this can be data in the traditional sense, or digitized voice. However, although the D channel is intended primarily for signalling, it is likely that the full bandwidth would not be needed just for that, so it could also be used simultaneously for slower speed data.

The rationale for BRA is its suitability for the smaller business, providing the capability to send voice and data simultaneously. It is important to note that the ISDN only provides the switched network, and for a business to use it, the business needs to purchase an ISDN multiplexer. This is the device that sets up calls and multiplexes the voice and data (or facsimile) onto the ISDN, and provides the D channel operation.

PRA is referred to as 30B+D because it comprises 30 "Bearer" channels of 64Kbps and one "Data" channel through a single connection. The Data channel in this instance is 64Kbps as it has to control more Bearer channels than the BRA. Nevertheless, a user could anticipate being able to use some of the D channel for actual data.

PRA is intended for larger organizations, and is suitable for providing multiple channels between PABXs as well as multiple data channels.

The term "nB+D" is used to denote the aggregation of n B channels to provide a higher bandwidth than 64Kbps. For example, video can be successfully carried over 384Kbps, so the ability to aggregate 6 B channels into a single functioning 384Kbps channel would enable video-conferencing to be carried out over the ISDN. Users wanting this capability must ensure the capability is available from the multiplexer, as the aggregation process is complex, and not always likely to be implemented in every manufacturers' equipment.

ISO

International Standards Organization. An organization within the United Nations to which all national or other standards-setting bodies (should) defer.

IT

Information Technology. A general term that refers to processing of information by automated means.

J

jargon

It used to mean the twittering and chattering of birds, probably from an onomatopoeic base (like "cuckoo", which sounds like the sound of a cuckoo). Later it meant unintelligible or meaningless talk. These days it generally applies (usually contemptuously) to the terminology of a public science or art. The IT (Information Technology) area is particularly guilty of using jargon because of the prolific use of acronyms.

journaling

This is a process whereby the computer records everything you've entered at the keyboard during a particular session. The information is stored in a file, or a journal. The reason for doing this is so that if the computer breaks down during your session, and you haven't remembered to save your work periodically, you can recreate your work up to the point of failure by running the journal against your process (edit session or whatever).

joystick

In computing, it takes the appearance of the original joystick of the old flying days. It's plugged into your computer, and is used to send signals to the software to tell it to move something around on the screen during a computer game. It could be used outside of games, but a mouse is generally preferred.

jump

This is a programming term used to indicate the act of skipping over a block of code so as to carry on from another spot (instead of the next sequential line, which is what the computer would normally do). This is what a GOTO does in BASIC.

justified

Used in Word Processing (WP) for adjusting the spacing between characters in each line of paragraphs so that they line up on the left or the right margin, or both, or even in the centre.

K

Kermit

A computer program that implements a set of procedures (i.e. a protocol) for transferring files between computers. Kermit goes beyond the usual ASCII file transfer, because it includes defined procedures on what to do when errors occur. Xmodem is another such protocol.

kernel

This word is used in a couple of instances, but it is basically a low level process on which the rest of the system is built. For example, an Operating System Kernel would be a fundamental set of routines that provide some low level services (e.g. getting the next keystroke from the keyboard). Other, more usable, services would then be built on top of the kernel, e.g. to read a whole line from the keyboard. Often you'll hear of a program operating in kernel mode. This means that the program is allowed to access the low-level facilities of the computer, that normal programs can't.

key

This can mean lots of different things. For some reason, those little buttons that you press on your computer ended up being called keys. The term used to mean some form of lever pressed with your finger on an organ, piano, flute, concertina and so on, and then was used for the levers on a typewriter, and the term just stuck (much the same as the keys). Now we have buttons that everybody calls keys.

keyboard

Because of the above, the thing that houses all the keys for your computer, plus the keys themselves, is call a keyboard.

keyword

This is usually an item in a database that is used to quickly access records in the database. For example, a customer number might be used as a key. The database system makes up a separate index sorted in the order of the key, and showing where in the database the whole record of information is stored. This is the primary reason why people always want to know your account number. It's because the database system uses it as a key to finding your data. Names can be used as keys, but the search isn't as efficient.

kill

Whatever you kill, doesn't exist any more. For example, you can kill a file, or kill a job that is currently running on your computer.

Kilo

Kilo actually means 1000, but because computers are binary devices, Kilo and its abbreviation K mean 1024, which is an integer power of 2. Therefore, a computer with 64K bytes of memory actually has 65,536 bytes of memory (64x1024).

kludge

Much like fudge. It usually means a quick fix, or patch, to a system to keep it running, because to do it properly (which is what should have happened in the first place) would take too long. Kludges beget kludges - you only need one kludge, and pretty soon your whole system is riddled with them.

Knowledge-Based Expert System

An area of Artificial Intelligence aimed at providing computer programs that imitate the behaviour of humans in solving problems normally thought to require experts, or specialists, for their resolution. An example might be medical diagnosis, where the computer can, through analysis of history and symptoms, and using information or knowledge previously input from diagnosticians, make a reasonable diagnosis of the patient's illness.

Knowledge Engineering

Refers to the process of acquiring knowledge from experts, and storing this into a knowledge-based expert system.

KWIC/KWOC

Means "KeyWord In Context/KeyWord Out of Context". Have you ever seen one of those indexes in a book where the index words are down the middle of the page, and the rest of each line (before and after the keyword) is a part of the sentence that the word appears in, in the book? That's showing the keyword "in context". If the keyword is just there by itself, then it's "out of context".

L

LAN

Local Area Network. The usual conception of a LAN is a series of PCs connected together. Some of the PCs are simply workstations that access one or more other PCs that are file or print servers (they've got disks and printers). It is also possible to connect minis and mainframes in a LAN for high speed sharing of processing.

language

The rules of syntax for constructing computer programs - for example COBOL, FORTRAN, BASIC, PASCAL languages. The syntax is how the sentences of the language are constructed - for example, commas must separate items in a list. Semantics, on the other hand, concern the meaning that is to be given to the sentences, as interpreted by the particular language. For example, the sentence (or "statement", as it is usually called) "X=Y" in one language might mean "pick up the value of the variable in memory location Y, and store that value in memory location X, replacing its previous value". Both variables X and Y still exist as separate entities. In another language, however, the same statement might mean "make all references to X and to Y synonymous" - i.e. have only one memory location, but allow it to be referred to with two different names. Thus, syntactically the two sentences are the same, but semantically, they are significantly different.

Laptop

A portable PC small enough to set on your lap. It is characterized by low power consumption memory and a flip up LCD. Some portables are called "luggables" because they are portable, but weigh a ton (sorry, tonne) - something like a sewing machine. A new classification is appearing these days, and that is a "notepad". This is a computer that is even smaller than a laptop, and might even fit in your pocket.

LAWN

Local Area Wireless Network. These are LANs that utilize short-range radio broadcast technology for communication, thereby eliminating wiring of computers in the office. The technology, known as "spread-spectrum", facilitates a highly reliable means of wireless communications, and is becoming the accepted approach to wireless LANs, or LAWNS as they are called.

The US Federal Communications Commission (FCC) enacted legislation in 1985 opening the way for short range wireless data communications through radio waves. Three frequency bands (called ISM bands, for Industry, Science and Medicine) were allocated for unlicensed spread-spectrum usage (902 to 928 MHz, 2400 to 2483.5 MHz and 5725 to 5850 MHz).

In Australia, the 902 to 928 MHz band has been reserved for cellular 'phone applications, and therefore a product in this band is unlikely to be certified for usage here. DOTAC (Department of Transport and Communications) indicates that the 2400 MHz band may be available.

In September 1990, the IEEE 802.11 committee was formed to develop a wireless LAN standard. It has already focussed its attention on a standard for unlicensed usage utilizing the ISM spread-spectrum bands. The committee is chaired by NCR and co-chaired by IBM.

LCD

Liquid Crystal Display. This is the technology that has been used in digital watches for years, but that is now sophisticated enough to be used for computer screens. It is particularly useful for portable PCs because of the low power consumption (less drain on the battery).

lightpen

A pointing device available for some screens that is used to select items on the screen. Placing the pen on the screen at the desired spot and pressing it against the screen triggers a flash of light that is detected by the screen's electronics, and transmitted to the computer software, so that the software can tell where on the screen the choice was made, thereby causing it to respond with "ILLEGAL CHOICE", or some such other. Lightpens are going out of fashion these days, as more and more people use a mouse.

link

A communications line. "Linking" is also another process of collecting together separate modules of a program and creating one machine language image of it so that it can be loaded into the computer and executed. Computer programs are generally developed in a modular fashion, and there is this final "linking" phase to created the whole program.

local

Terminals are local if they are directly connected to the computer, and remote if they are connected using communications lines. A terminal, whether locally-connected or remotely-connected, can be referred to as being in "local-mode", which means that it is no longer "on-line" to the computer, but is controlled directly through the keyboard.

In programming, variables are local if they can only be accessed by the subprogram in which they are defined. Here, the opposite is global, which means that the variables can be accessed by all the subprograms.

log on/off

(Or eg in UNIX, login/logout) Logging ON is the process of identifying yourself to the computer, so that it can establish that you are a bona fide user of the system.

Logging OFF is the process of informing the computer that you no longer want to use it. With PCs, you usually don't have to bother with such things, unless it is connected to a LAN, in which case, the LAN operating system might require it.

logical

In computing, something's logical if it isn't physical. Something's physical if you can touch it. For example, see "LU6.2" below. A Logical Unit is really the protocol used to talk to a device. This is distinct from the manifestation of the LU, which is the physical device reacting in a specific way. Several different physical devices can use the same LU protocol. This saves the software from having to do different things for different physical devices.

"logical" is also a term used in programming for a variable that can have one of two values, the values being TRUE or FALSE.

LU6.2

A discussion of LUs and IBM's SNA generally is beyond the scope of this document. In short, LU stands for Logical Unit, and is usually associated with a terminal (screen or printer). There are several types of LUs (0, 1, 2, 3 and so on) corresponding to different types of devices. The LU type determines how to communicate with the device - i.e. the communication protocol (conversation rules and message formats). LU6.2 is a fairly new protocol, and is associated with programs, not terminals. It is a set of rules for program-to-program communication.

M

macro

A macro is a shorthand definition of a whole sequence of commands. These commands can be assembly language commands, spreadsheet commands, 3GL or 4GL programming language commands, operating system commands, and so on. The idea is that if you are using a certain system, and you find that you are constantly having to enter the same sequence of commands into the system over and over, it is possible to gather the sequence together into a macro definition, give the macro a name (preferably a mnemonic - see below), and then invoke the sequence by simply keying in the name of the macro.

MAN

Metropolitan **A**rea **N**etwork. As the name suggests, this is a network that services a metropolitan area. It could be considered to lie somewhere between a LAN and a WAN, but the term "MAN" is generally used to refer to a network that implements a particular protocol, that protocol coming under the generic term "Fast Packet".

Fast Packet technology emerged from the increasing reliability of today's communications links, as we move from analogue to digital technology. Our normal view of packet-switching is *à la* X.25, that includes quite a lot of protocol processing in setting up sessions, and implementing error checking and recovery mechanisms, on the assumption that the underlying communications links are unreliable. In areas where the links are highly conditioned (i.e. low error rates, e.g. within the CBD (Central Business District)), this processing is largely unnecessary.

There are two general types of Fast Packet - frame relay and cell relay. The essential difference between the two is that frame relay operates on variable-length packets, and cell relay operates on fixed-length packets. Without delving too much into the design parameters, frame relay was designed as an interface type of technology (e.g. grabbing one packet from a LAN, whatever its size, and shipping it off to another LAN holus-bolus), whereas cell relay was designed as a general-purpose switching method for carrying voice, data, image and video. Consequently, frame relay techniques tend to be used from a private network perspective, and products are available today. Cell relay is preferred by the carriers for implementing public networks. Telecom is in the process of introducing a cell relay Fast Packet offering that has been coined Fastpac. This follows the cell relay standard IEEE 802.6, sometimes referred to as DQDB (Distributed Queue Dual Bus), and sometimes also referred to as QPSX (Queued Packet Switch eXchange). This technology was actually developed at the University of Western Australia, and a joint venture company (QPSX Communications Ltd.) was started between the University and Telecom to progress the technology.

Man-Machine Interface (MMI)

The Man-Machine Interface (MMI) is a term that encompasses what a human operator sees, hears and feels, and how a human operator reacts to a computer system. An MMI is "user-friendly" if this interaction flows easily and smoothly. See "Human Factors".

Mega-

Prefix for 10 to the power of 6 - i.e. one million. Actually, there is something of a discrepancy in the computing world about how this is used, along with the use of "Kilo", which means one thousand. When we talk of 1K - meaning one Kilosomething, we usually mean 1024, and not 1000, and that's because computers work in powers of 2, and 1024 is 2 to the power of 10. 1M, then is actually 1024x1024=1048576, which is actually a good bit bigger than one million.

memory

This is the means by which all those 1s and 0s are stored in a computer. There are various types of memory.

ROM is so-called "Read Only Memory" because the 1s and 0s can only be read by the computer, and cannot be written over. A special device is needed to store the 1s and 0s into the RAM in the first place. ROM is normally built using very fast access electronics, and is by necessity "non-volatile", which means that when you turn the power off, the 1s and 0s stay there (otherwise you'd have to use that special device to put the information back in). ROM is usually used for some very basic software components of computers and communications devices.

RAM stands for Random Access Memory, and like ROM is built using very fast access electronics. Its name is not meant to imply anything about other types of memory, as most memory can be accessed randomly. The exception, of course, is magnetic tape, where, if you want to read the bit at the end, you have to read **sequentially** through all the rest of it. One major use of RAM is to store your program whilst it is being executed by your computer. RAM is usually "volatile" (although it doesn't have to be - it's a lot cheaper to make it that way, though). This means that it doesn't remember any of the information stored there after the power is removed. This usually doesn't matter, because when the power is turned back on, the software can be loaded back in from some cheaper non-volatile source.

Disk storage is memory too, and is sometimes referred to as "rotating memory". Disks are platters coated with a magnetic oxide. The 1s and 0s are represented by magnetized spots of different polarities. Disks are cheaper than RAM in terms of bits/$.

Drum storage is technologically similar to disk storage, except that instead of the magnetic oxide being coated on flat platters, it is coated on the outside of a drum-shaped object.

Other technologies are becoming available, such as Compact Disc, and optical disks. These are still slower than the more conventional hard disks, but have the enormous advantage of extremely high capacity. Consequently, they are useful in such applications technologies as imaging and archiving.

Metropolitan Area Network
See "MAN".

MFLOPS
"million floating-point operations per second". The peak MFLOPS rating is a theoretical best rate for a CPU performing nothing other than floating-point calculations. Like the MIPS rating, it is only a general reference, and is seldom achieved in real applications.

MFLOPS (Linpack)
Floating-point benchmark conducted and published annually by Dr. Jack Dongarra of the University of Tennessee. One of the most widely respected technical-computing benchmarks, the Linpack tests more than 100 systems of all types using vector matrices of two sizes: 1,000x1,000 and 100x100. Because the 1,000x1,000 test, by its very nature, devotes a greater percentage of CPU time to pure computation than does the 100x100 test, the former always yields a much higher MFLOPS rating than the latter. The test can also be conducted using 32-bit (single-precision) and 64-bit (double-precision) floating-point formats.

MFLOPS (Livermore Loops)
Floating-point benchmark conducted by the Lawrence Livermore National Laboratory. It consists of 24 FORTRAN subroutines, or kernels, that test computational performance. The most accurate indicator of real application performance is the harmonic mean of the 24 individual kernels. Typically, the harmonic mean rating is from 1/40th to less than 1/100th of the vendors' peak performance claims.

micro/mini/midi/mainframe
Many years ago, computers came in two basic flavours - mainframes and minicomputers. Presumably, a mainframe was meant to represent the main computing power, and the fact that it arrived as a lot of wires rack mounted in frames, could have suggested the name. They are packaged differently today, but the terminology "mainframe" has a fairly well-defined meaning, and is represented by a large computer installation, servicing lots of people doing a variety of tasks, and probably doing a lot of database work (e.g. keeping track of inventory, processing claims, making sure every Australian is paying tax, and so on).

A minicomputer, on the other hand, was a much less powerful machine, usually dedicated to a particular task or area, and probably involved in some form of process control work (i.e., the computer would be electronically connected to some (say) manufacturing process, and would monitor and control it). The terminology suggests that it was just a little computer, a computer then being quite powerful. Gradually, minicomputers became more powerful and less expensive, and found their way into larger business organizations. This changed their role somewhat, and pressure increased to squeeze more and more out of them. Nowadays, people talk about "superminis", but there is no clear-cut distinction between a mini and a supermini - one can do more work in a given time. Bear in mind that these machines are still quite expensive, and are acquired only by larger organizations. Sometimes you will hear the terminology "departmental machines". These are usually minicomputers purchased by a large organization that already has a mainframe, but that wants to off-load some of its departments' computing requirements to smaller independent machines.

"Midframe" is another term used today to represent a machine poised somewhere between a supermini and a mainframe. Some of the distinctions are fuzzy, and people tend to call their machines by the way they perceive them to be - for example, someone was overheard recently referring to his departmental machine (which was used mostly for word-processing, and which at that particular time wasn't processing too many words) by saying "Ah! The **mainframe**'s gone down again"!

Then came the microcomputer, which, as the name implies, is a very small computer. The singular distinction that a Micro has is that it is a processor on a chip. This means that all its electronic wizardry has been condensed into a very small space, whereas with minis and mainframes, the required electronics occupies a lot more space (and generates a lot more heat - some mainframes have to be water-cooled). The microcomputer brought computing to the masses, and has become so popular, that some people forget that computing today encompasses not only the things we do on micros, but also the myriad of tasks that we do on minis through to mainframes.

When considering computers in the office, we are generally referring to micro-computers, either individually or as a collection linked in some way (referred to as networked). Using computers in the office is often termed Office Automation (abbreviated to OA). This term implies something magical that, sadly, isn't generally available. Even today, most computer manufacturers cannot offer an integrated set of fully communicating products that are easy to install and easy to use.

Some people think that OA is Word Processing (abbreviated to WP). It's usually not the office that thinks that, but the person selling the WP package. WP is certainly a part of OA, but is definitely not the whole. WP lets you do things like write letters, prepare reports, merge letters with address lists, and so on. It is essentially a way of entering text into the computer for ease of manipulation and economy of storage.

Having all that stuff in there is only a part of the battle. One of the most useful facilities in any OA system is an electronic document filing system. This lets you store your WP documents into electronic document cabinets. Such a system lets you eliminate a lot of your actual paper storage. Documents are maintained on disk, and filed away using keywords. This lets you do extremely useful things like locating all the letters sent out to customers or clients that made reference to one of your new products, for example, or finding the last report sent to your FAS. It is essentially a filing cabinet on a computer disk, but with powerful search and retrieval facilities. Ideally, your document filing system should also be able to handle documents other than those created by your WP system. Spreadsheets, for example, should be included. Ordinary text files should be included. In fact, all documents that you work with in your office should be included.

Another OA facility is Electronic Mail (abbreviated to EM). This is simply the ability to create messages and send them to other users of your OA system, and even users of other OA systems. Often, much is made of an EM facility, but in a small office, it has limited value outside of leaving people reminders and the like. The real worth of EM comes out in a large distributed organization, where you can send messages across the country. An indispensable feature of any EM facility is the ability to attach documents or reports to mail messages - e.g., you want to send a report to your boss, with a covering letter. Another feature is the ability to set up your own distribution lists and nicknames so that you can send information to all people in a certain group, for example.

These are some of the features of an OA system. There are many others, but an OA system with a fully integrated set of communicating facilities will be hard to find.

microcode

See "firmware".

microfiche

"fiche" is from the French for "a slip of paper". A microfiche is a piece of film bearing a microphotograph of a document.

MIPS

"million instructions per second". The rate at which the central processing unit performs instructions of all types. This is a general measurement (not a benchmark test) with little consistency from one vendor to another.

MMI

Please refer to "Man-Machine Interface".

mnemonic

This is something which is designed to aid the memory. The term is used a lot in assembly language programming, where instead of using the binary representation of a machine language instruction, the programmer uses a mnemonic which is later translated into the binary representation by an Assembler. For example, SUB might be a mnemonic for the instruction that does subtraction, DIV for division, JMP for "jumping" to another instruction sequence, and so on.

modem

An acronym for modulator/demodulator. This is a device that converts computer digital signals to analogue signals for transmission over telecommunication networks, and subsequently converts the signals back into digital signals at the other end. The long term future of modems is bleak, as most of the world's telecommunications are slowly being converted to digital technology.

mouse

An acronym for Machine Oriented User Sensing Equipment. (The author just made that up, but sometimes that's how acronyms are made up - someone thinks of a catchy name they'd like to call their product or process, and then fits some words to it). A mouse is a small hand-moved device that lets you move a cursor or icon about the screen. The movement of the icon on the screen follows the movement of the mouse on the desk.

multi-media

Refers to the integration of information from various sources (e.g. word-processing, voice, sound, graphics, full-motion video) into an information system.

multiplexing

This is generally (but not always) used as a communications and networking term, and is simply the process of taking several different data streams, sending them over the one communications line, and splitting them up at the other end into the original data streams. This allows you, for example, to run, say, four separate 2400bps data streams over a single 9600bps line, and therefore get some cost benefit. A device called a multiplexer (sometimes abbreviated to MUX) is needed to do the multiplexing.

multiprocessing/multitasking

There is sometimes some confusion between these terms. Multiprocessing is something that occurs in a multiprocessor environment. In such an environment, we have two or more CPUs accessing a single file system (disk system), and presenting what is called a "single system image" to the user - it appears to be a single computer. Multitasking (sometimes called multiprogramming), on the other hand, is the capability of a single CPU to run several tasks at once. Each CPU in a multiprocessing environment would probably also be multitasking.

N

nanosecond

A thousandth of a millionth of a second (a millionth of a second is called a microsecond - sometimes written as "_sec"). It is 10 raised to the power of -9.

Natural Language Processing

An area of Artificial Intelligence. Computer programs are written to understand the language used by people in their everyday conversations. There are two separate sub-areas of Natural Language Processing. One deals with understanding written text, the syntactic and semantic knowledge of the language - i.e., the computer knows exactly what has been said (in terms of sequences of words), and now must be able to interpret the meaning of these words. The other deals with actually understanding the spoken language by identifying individual sounds and eventually combining these into a meaningful sentence structure of words, and storing the sentences as text strings in the computer.

nesting

It's a bit like "stacking", where one thing is stacked inside another. If I have a program that calls a subroutine, but that subroutine needs to call another subroutine to complete its work, and that subroutine in turn needs to call another, then we say the routines are "nested" (to a depth of 3). There's usually a limit on the nesting depth, brought about by either a compiler limitation, or a limitation in the machine itself.

network

A collective term for the bits and pieces that are used to connect electronically a variety of processing equipment. (The term "network" can also apply to a group of people or contacts, but this usage is not defined here.) To those from a mainly voice-oriented background, networking means connecting PABXs together. To those from a mainly data-oriented background, networking means connecting computers together. The term applies equally to both areas, and there is a trend to integrate the two, along with others such as image and video.

Integrating voice and data can occur at different levels. Some would say that having one group in an organization looking after both voice and data somehow represents a form of integration. Another level is actually to use the same bit of wire for both, but use multiplexer devices to separate the voice and data into different channels. Another level is where the voice and data are packetized, and mixed up in the same channel. The separation is then at the packet level, not at the physical level as is the case with multiplexers.

network architecture

A network architecture is the way in which a communications network is constructed, and the communications protocols it supports. For example, the traditional IBM mainframe network architecture is a star-shaped arrangement (although this is changing) connecting end-user devices to the central computing site. The communications protocol is IBM's System Network Architecture (SNA).

This contrasts with the traditional DEC (Digital Equipment Corporation) approach of Local Area Networks (LANs) connected together over a wide area in various mesh configurations for load balancing and redundancy. DEC refers to its network architecture as Digital Network Architecture (DNA), and to its communications protocol as DECnet. The underlying protocol is called Ethernet, with higher level protocols building on that.

Neural Networks

A neural network is composed of many interconnected **processing elements** that operate in parallel. It works in a way similar to how we think the neurons in the human brain encode information.

Each processing element can have several inputs received from the output of other processing elements. Further, the link that carries the output from one processing element to an input of another processing element has a weight. The actual value that a processing element receives for each input is the input value multiplied by the link's weight. The processing element then performs a simple calculation (transformation) using the input values, and produces a single output value. This output value then fans out to help to make up the inputs of other processing elements.

The networks that are built up from the processing elements interconnected in this way form highly parallel, distributed information processing systems. They can perform complex mappings from a set of inputs to a set of outputs. Much of the current fascination with neural networks has to do with their ability to learn. The most popular learning algorithm today is called "back-propagation". To solve a problem with back-propagation, the network is presented with sample inputs along with the desired outputs, over and over, while the network "learns" by adjusting its weights until it produces the correct output for every input. Back-propagation proceeds in two passes. In the forward pass, inputs proceed through the network to produce the output. Then, in the backward pass, the difference between the actual and desired outputs generates an "error signal" that is propagated back through the network to "teach" it to come closer to the desired output. It can be seen that such networks could be the bases for pattern recognition techniques.

One important distinction between Neural Networks and Expert Systems is that Neural Networks can't explain how a solution to a particular problem was solved - it has merely "learned" to come up with the right answer. Expert Systems, on the other hand, can trace back through the steps taken to solve the problem.

A good introductory text to neural networks is "Neural Computing, Theory and Practice", Wasserman, P., Van Nostrand Reinhold, 1989, ISBN 0-422-23776-6.

node

A general term that refers to a communications processor. It usually represents a place where a number of communications lines are concentrated for trunking or switching.

notebook

Refers to a computer that is even smaller than a laptop. Terms such as luggable, transportable, lunch box, laptop and pocket are all variations on the theme of portable computing. Notebooks are generally used to perform the time management functions that are available on larger PCs (diary, calendar, calculator etc.).

null

A term used to differentiate between something you can't see and isn't there, and something you can't see but is there, the former being null. An example of the latter is the space character. If you can see it, then presumably it is there. One could argue, though, that one could see something that wasn't there by virtue of the fact that one could see that it wasn't there, but one couldn't determine whether it was null. The preceding account is somewhat specious, however, because the null character is represented by a byte of all zeros, and that definitely is there, even though it represents something that isn't.

Some computers have a null instruction, which asks the computer to do nothing for the normal number of clock cycles. This can be quite useful in getting the computer to delay for a required period of time.

number crunching

A term used to refer to computing that is computationally intensive (e.g. inverting matrices) as opposed to computing that relies on a lot of input and output (e.g. printing out a mailing list).

nybble

Half a byte.

O

object code

This terminology had quite a rational beginning, but is now a bit blurred. We used to have source code and object code. The source code was the code the programmer wrote, in some high level language (e.g. COBOL, FORTRAN). The object code was the same program, but compiled into the machine language of the object machine (the machine that the program would run on). The object code would run on the object machine. Nowadays, the object code, although generally specific to a particular machine architecture, cannot run on that machine. It is still in an intermediate form. So now we have source code (suitable for compilation over a number of different computer architectures) that is compiled into object code (suitable for "linking" over a number of different computers with similar architectures) that is in turn "linked" into load modules (suitable for running on computers with exactly the same architecture).

object-oriented databases

Databases where the data definition language and data manipulation language are object-oriented languages.

object-oriented design

A software or system development technique in which the system is seen as a collection of *objects* that communicate with other objects by passing *messages*. Design is targeted towards defining the kinds of objects, the methods (i.e. procedures of objects) and the messages passed. OOD is based on the principle of *information hiding*, i.e. disclosure of the essential higher level details, but not superfluous complexity of detail.

object-oriented programming

"Object-Oriented" is a term that refers to a methodology, i.e. a way of doing things. It began as a new way of computer programming - that is Object-Oriented Programming, or "OOP" as it is affectionately called. Sometimes you will see "OOPS", which is an OOP System. To understand the explanation given here, the reader will need to have a basic understanding of conventional programming.

OOP is actually an extension of the more familiar and conventional "structured programming". Objects are an extension of the conventional records and structures in that they also include the operations (called "methods") that can be invoked upon the actual data.

The main distinction between object-oriented programming and the traditional procedural programming (e.g. using COBOL, FORTRAN, BASIC, PASCAL) is that it is the *data* that are organized in a basic control hierarchy, whereas in procedural programming, it is the program statements that are organized into a structure, and these operate upon the data, which is treated as secondary. There are procedural programs in object-oriented programming, but these are secondary, and are encapsulated as part of the object definition. The program is invoked by a message being sent to an object (by a user, another program or another object).

It is important to note that "object-oriented" has nothing to do with graphical objects on a screen. Some software manufacturers take advantage of this misunderstanding by declaring their product to be "object-oriented" because the user interface is icon-based. "Object-oriented" applies to the method of developing the software, not to the user interface. A piece of software can have the same user interface whether it was developed using OOP techniques or structured programming techniques.

OOP assumes that it is better to base the structure of a software system on the objects of the system rather than on the actions which the system performs. The OO approach is rapidly gaining adherents because of the claimed benefits in software reusability (using software for other projects), extensibility (improving the functionality of a software system) and modularity (simplifying the structure of the software).

OOP comprises the specification of objects (using some OOP language) and their interrelationships in some hierarchy. To understand OOP techniques, there are three terms that need explanation: **encapsulation, inheritance** and **polymorphism**.

<u>Encapsulation</u> denotes the fact that an object has both **data fields** and **methods** - i.e. procedures and functions - that act on the data. Programs never access data directly, only through the provided **methods**. For example, we might have the object MESSAGE that writes a message on the screen. The object contains not only a data field to contain the message, but also a **method** (procedure) that is used to actually write the message on the screen. Encapsulation further promotes the concept of "information hiding", an important concept in software engineering. The idea is that the less one part of a program knows about the other parts, the

better - there is less likelihood of the first part making unnecessary assumptions about how the other part works, thereby minimizing the potential for having to change the first part just because we decided to change how the other part performed its operation.

Inheritance refers to the fact that objects can have descendants. It is often the case that we might want to create an object that is "just the same as that object over there, but...". For example, we might want to create an object called PLACEMESSAGE that is just like MESSAGE, but that writes the message at a particular position on the screen. In OOP, we would simply say "PLACEMESSAGE:=MESSAGE", and then proceed to include the differences. This promotes the re-usability of code already written. The descendants inherit all of the features of the ancestor, and can add **fields** and **methods**, or redefine **fields** and **methods**. Objects can have many descendants, but only one ancestor. Object structure is then a tree.

Polymorphism means multiform, or "many shapes". It has been adopted into the OOP jargon list to cover the concept of "virtual methods". An object's **method** (procedure, routine) is called "virtual" if its location in memory cannot be determined until the program is actually running. This can happen in a number of ways, but one way is through inheritance. An object can inherit a **method**, and that **method** might have been inherited from another object, and possibly changed. **Methods** might change according to events that only happen during run-time. By way of illustration, suppose we adopt a software design rule that all objects must include a **method** that will list onto the screen (or printer or file) a description of the object - what it does, how it does it, any oddities. We also stipulate that the method must go by the same name in each object, say we call it "TELL". We now want to create an object LISTTELLS that will go through and invoke all the other objects' TELL **methods**, the outcome being a complete description on the screen (or printer or file) of all the objects that make up the application. We program this as a loop, going through a list of object names (which might have been determined at run-time), and invoking their TELL **methods**. The OOP compiler doesn't know where in memory all of these TELL **methods** will be, so this determination will have to wait until run-time.

OED

Oxford English Dictionary. This abbreviation is included, as it is used throughout this glossary.

on-line

This signifies a process that is interacting with a computer in real-time. For example, an operator processing a transaction in a bank while you are waiting. The terminal is on-line to the host. This contrasts with off-line, where the transaction is simply requested on-line, but is then processed later on in the background (e.g. at night).

OOP

Object-Oriented Programming. Please refer to the discussion under "OBJECT-ORIENTED".

Open Systems

The term "Open Systems" refers to computer operating environments, application development environments and application running environments that are open in the sense that much is known about them (their design and specifications are in the public domain). Further, these environments must be common over several platforms (hardware and operating systems). Publishing a proprietary architecture does not make it open if the application environment is still restricted to particular platforms.

The goals of Open Systems are to rationalize the application development process, rationalize the application distribution and packaging process, and contribute to application portability.

UNIX, *per se,* is not an Open System. There are many different versions and variants. There is no guarantee that an application developed under one UNIX environment will run unchanged under another. All that one can hope for is that there will be enough commonality to reduce the porting exercise to a minimum.

The real gains will come when the application developers can assume a common interface into the operating system, coupled with a common application development environment and standard Graphical User Interface (GUI). If these were available, then the actual operating system wouldn't matter - it could be UNIX, MVS, VMS, MPE, OS/400, OS/2 or DOS, and the user interface would be the same.

POSIX is a program undertaken by the IEEE for the development of a Portable Operating System Computing Environment, and it defines, *inter alia,* standard interfaces into the operating system. An application interfacing into POSIX can run on any POSIX-compliant platform. DEC, for example, is developing a POSIX interface into VMS.

ANDF stands for Architecture-Neutral Distribution Format, and is an initiative of the Open Software Foundation. ANDF provides a single means of distribution to multiple platforms. Under ANDF, the application developer uses an ANDF "producer" to convert the software to a common intermediate ANDF code (much like a compiler). The ANDF code is then distributed to end users. The user then runs an ANDF "installer" to complete the compilation into code specific to the machine.

POSIX and ANDF are facilitators of Open Systems.

operator/operand

Terms used in assembly-language programming, and to some extent computer system command processing. The command is an operator, and the parameters with which it is to operate are operands.

operating system

An operating system is a suite of service programs that facilitates the use of a computer. For example, MS-DOS is an operating system for PCs that provides various file services (copying, creating, deleting, listing) as well as programming services in the form of system calls to perform low-level computer functions.

orphan

When the first line of your paragraph appears by itself at the bottom of a page, that's an orphan. See also "widow".

OS/2

OS/2 is a high-performance, multi-tasking operating system optimized to run on industry standard Intel 80286/386/486 computers. The operating system is supported by IBM AT and PS/2 computers, and can be fairly easily ported to other 80x86 computers.

The design of OS/2 was primarily to make it the most suitable operating system possible for mainstream microcomputer-based Office Automation (OA) tasks. These tasks span the spectrum from GUI-oriented end-user applications (Graphical User Interface) through to file, print, electronic mail, DBMS (Data Base Management System) and enterprise-wide connectivity server processes. OS/2 was designed from the outset to support the client/server model for distributed computing.

Presentation Manager is one of the subsystems of OS/2, and is the equivalent under OS/2 of Windows under DOS.

OSI

OSI stands for Open Systems Interconnection. It is, in fact, a model for the construction of networks, involving not only the hardware interfaces required to connect systems, but also the software needed to provide meaningful communications. From this model has emerged numerous STANDARDS for the development of OSI systems.

Consider as an example the situation where you went out and bought an IBM system because you wanted to use DB2, and you also went out and bought a DEC VAX system because you wanted to do some engineering work, as well as use ALL-IN-1. Then you would have some difficulty when you wanted your IBM terminals to access the DEC machine, and (to a lesser extent) vice versa. Terminal access is only one manifestation of the underlying problem, and the underlying problem is that IBM and DEC have developed independent proprietary network architectures.

IBM has SNA and DEC has DECnet, and they're different. Now this is going to be a problem for some time yet, because IBM has a huge investment in SNA, as does DEC in DECnet, and neither is going to abandon its architecture. This is not meant to imply that this is an IBM/DEC problem - it's a problem concerning ALL proprietary network architectures.

In recognition of the general problem, several organizations and bodies began work on developing a solution. The solution was to develop a model for interconnecting diverse systems that would be acceptable to everybody. An impossible task, of course, but over the years a suitable model has emerged that has mostly been endorsed as an International Standard. There's still a lot of work to go into the model, but there is enough stability in some of it for it to be implemented by computer manufacturers. This model is called the OSI Reference Model, and if everybody implemented it, then we'd all be able to communicate with each other. There are more standards organizations than would seem to be necessary, and a whole industry has evolved for the process of standards development, resulting in a whole host of buzzwords and jargon of no general interest to most of the population. A real live problem has evolved because of this, not only because different standards are being adopted by different countries, but also because there are so many standards options to choose from. So, IBM might say "OK, we'll give you some OSI software, but we've implemented a CONS plus only Transport Class 0", and DEC might say "OK, we'll give you some OSI software, but we've implemented a CLNS plus all Transport Classes up to 4", so that STILL never the twain shall meet. (This is not exactly what has happened, but is indicative of the type of thing can and does happen).

To alleviate this problem, a number of organizations have developed what are called "functional profiles". These are, essentially, specifications of the OSI options that the organization is prepared to accommodate. They are a kind of "standard of standards" for an organization. Governments have developed such profiles, and this is where you might come across the acronym GOSIP - Government OSI Profile. A GOSIP is of fundamental importance, because it implies a procurement restriction. That is, if a manufacturer cannot satisfy the GOSIP, then he'll have to sell his products elsewhere.

Now, the US has developed a GOSIP, and the UK has developed a GOSIP, and they are fundamentally different, notably at what is called the network layer (although each can accommodate the other). The Australian Government has also developed a GOSIP based on the UK model.

You can see the motive for developing an OSI model, and the subsequent headaches that are looming in attempting to standardize on the standards. Now, we will look at the actual structure of the OSI model, and so explain some of the terminology associated with it.

The OSI model is a model for interconnecting disparate systems. It is not a prescription for building individual systems, and was never intended to be so. It is a structure that defines the processes and functions that would be needed for systems to interconnect. In a way, OSI is inaptly named. It is not there to interconnect open systems, but was devised precisely to interconnect "closed", or proprietary systems.

The OSI model has delimited these processes and functions into seven main areas. The ultimate aim of the model is to get an application (i.e. a program) in one computer to talk to (interact with, exchange information with) an application in another computer.

Now, as far as the applications are concerned, they simply wish to issue commands like "CONNECT TO APPLICATION X", "SEND DATA BUFFER 1", "RECEIVE ACKNOWLEDGEMENT", "TERMINATE CONVERSATION", and so on. The computers are connected by a bit of wire, so somehow these application commands have to get translated into very low level commands that can communicate over the bit of wire. That is, the applications need the services of lower level functions that eventually translate into commands for sending data over the bit of wire.

The OSI model defines seven levels of these lower level functions, from the application level down to the physical level that actually transmits the data. Each level assumes the services of the level immediately below it to perform its function. In concept, it is very much like a program having to call a subroutine to perform a function, and that subroutine having to call another subroutine to perform its function, and so on until eventually a subroutine is invoked that sends the data over the bit of wire (seven levels down).

These levels are called layers, and that is really only to conceptualize the process. One could imagine the seven levels sitting one atop the other, appearing as a stack of layers, each dependent on the services of the layer immediately below it. This is where we get the terminology "protocol stack". Starting from the lowest layer, we have:

<u>Layer 1 - the Physical Layer</u> provides an interface to the physical media for the transmission of bit streams. It controls the electrical aspects of the communication, such as voltage levels, pin connector assignments, cable lengths.

Layer 2 - the Data Link Layer provides data transfer over the physical connection, involving character and bit stream framing, data identification, synchronization, and some error detection and correction.

Layer 3 - the Network Layer provides addressing and routing of data, even where this is across several physical connections, including the public network. This is the layer at which X.25 stops.

Layer 4 - the Transport Layer is a crucial element of the model. It provides the routing necessary within a system for end-to-end communication between individual processes. It can also optimize costs and performance by multiplexing over one network connection and/or splitting over several. It provides 5 classes of Quality of Service, referred to as Transport Classes 0, 1, 2, 3 and 4, to be used according to the reliability of the underlying real network.

Layer 5 - the Session Layer and above cease to be concerned with the means of transferring data, and are concerned with supplying services to applications to enable them to communicate. The Session Layer provides application dialogue co-ordination and management (like saying "Roger" when it's the other person's turn to speak).

Layer 6 - the Presentation Layer is concerned with agreeing the structure and encoding to be used for transferring data. This layer assumes that an end-to-end path exists, that it possesses the required Quality of Service, and that any required multiplexing, splitting, segmentation and dialogue control functions are available from lower layers. It is concerned purely with the representation of data fro transfer.

Layer 7 - the Application Layer provides the interface between the OSI communications environment and the application processes using it, including specific services relating to the type of communication (file transfer, remote processing, terminal interaction support). Within the Application Layer, there are common functions which offer an infrastructure on which to support real applications. Two such functions are Association Control Service Element (ACSE) to help set up an association between two OSI applications, and Commitment, Concurrency and Recovery (CCR) to help synchronize application processes (such as transaction processing) and recover from errors.

P

PABX

Private Automatic Branch Exchange. This is a telephone exchange that doesn't belong to the public telephone system, although it would usually be connected to it. It is an economical means of providing an organization with a telephone system without having to install masses of telephones through the public system. These days, PABXs are pretty clever, and can handle voice and data. This can provide a very cost-effective LAN in an organization - the LAN simply uses the existing telephone wiring.

pacing

This is a term used in computer communications. It is concerned with flow control in a network. Effectively, the computer "paces" itself when sending packets of data into the network so as not to flood it. It is usually implemented in the form of a pacing count, which is the maximum number of packets that any particular sender is allowed to send into the network without an acknowledgement from the receiver.

packet switching

See "X.25". It refers to a data movement scheme where the data is chopped up into small segments called packets. Each packet has a destination address. Lots of people can be sending data to lots of other people at the same time. A series of packet switches routes the packets according to their destinations. The process is analogous to the telephone system.

paging

This is a term used in computer memory management. Programs are allocated what is called a virtual memory space. This is the amount of memory you've always wanted, but can't afford. The computer never actually has this much real memory, so it breaks its real memory up into chunks called pages (maybe 4K bytes - maybe 16K bytes). To run your big program in its little memory, it only brings in enough pages of your program from disk to get started. As it processes your program, it "pages" your program in and out of memory according to which bits it needs at any one time.

parallel

This word is used in a number of contexts, but generally implies processes or events that are happening simultaneously. In communications, its counterpart, or complement, is "serial". Serial communications means that the bits are sent one after the other. Parallel communications means that groups of bits are sent out in succession. Clearly, parallel communications is faster, but the trade-off is that it is more expensive (more sophisticated communications equipment, higher transmission capacities, and so on).

PC (IBM Compatible)

The original IBM PC was built around the Intel 8088 processor. This is basically a CPU (Central Processing Unit) that recognizes a particular set of instructions. The architecture of the computer is the way the CPU interacts with all the other bits, plus the definition of the instructions that the computer understands. One fundamental feature of the 8088 architecture is that it has what is called an addressing space of one megabyte. This means that it can only handle up to one megabyte of memory. One megabyte equals 1024K.

The IBM PC has a clock rate of 4.77 MegaHertz (MHz). One MegaHertz equals a million Hertz, and a Hertz is just another name for "cycles per second", named after a German physicist. Now, 4.77 sounds like a strange value to pick, but it arose naturally from the frequency of the crystal that was used to derive the pulses. The crystal has a natural oscillation of 14.31818 MHz, and the PC clock takes every 3rd of these to drive the PC.

What the clock does is send pulses through the circuitry, to make things click and whirr, much like a heartbeat. The PC needs a few clock pulses to drive it through each complete instruction. Obviously, if the clock rate were increased, then the PC would run faster. In fact, after a while, manufacturers came out with PCs that had faster clocks, like 8MHz and 10MHZ, and called them Turbo PCs. These were usually switchable back to 4.77 if you wanted, because some software had timing dependencies based on the 4.77 clock.

Unfortunately, you can't just keep cranking up the clock rate. There are some fundamental timing limitations inherent in the computer architecture, and the only way to fix that is to change the architecture. This is, in fact, what has transpired.

Now, the CPU has to "talk to", or interact with all the bits that make up the computer. The memory is something that the CPU has to interact with all the time. The memory holds all the instructions. It also holds all the data that the instructions are to operate on. So, the CPU is constantly fetching stuff from the memory, and storing stuff back in there. It also has to interact with all those I/O devices (I/O = Input/Output), like the keyboard, the screen, the disks. So, the CPU does this over a thing called a bus. A bus is just a data path. Data travels up and down the bus between the CPU and all the things it has to talk to. Obviously, the more data you can send up and down the bus, the faster your program is going to run. The 8088 has an 8-bit wide bus, so that it is limited to one byte at a time. The 8088 is actually the 8-bit bus version of the Intel 8086 (which has a 16-bit bus). IBM probably chose the 8-bit bus because it tended to make lots of the other electronic components simpler in design - so it was a trade-off.

Having all this doesn't make the computer actually do anything. When you switch your PC on, the CPU starts fetching instructions from a predetermined location in memory, and starts to execute them (that means make the computer do what the instruction requests). If the memory is empty, it's going find nothing, and stop. To overcome this, IBM wrote some fairly basic software that was given control when the machine was switched on. Added to this software were some, again, fairly basic routines for low-level access to things like the keyboard, the screen, the disks and so on. This software is called the BIOS - Basic Input/Output System. The BIOS has been the root of a lot of problems in the industry, because it's copyrighted. This is why the word "compatible" has popped to the surface and been abused and overused. A compatible PC is a non-IBM machine that has a BIOS that isn't exactly the same as the IBM BIOS, but that provides (almost) the same functionality. The degree to which it is the same is the degree to which the machine is compatible. It can't be the same, but it can be similar, or compatible.

No one is going to buy a PC that comes with only a bit of BIOS. That's extremely low level software, and if we wanted to do anything at all, we'd have to get involved in some assembly language programming.

What we need is for somebody else to do all that, and present us with some higher level services - just simple things like being able to create files, copy them, delete them, print them etc. Well, somebody else did. A few people did in fact, including IBM (through Microsoft) and Microsoft itself. These higher level services are offered by way of a suite of software routines, and called the "Operating System". IBM called its operating system PC-DOS and Microsoft called its operating system MS-DOS. DOS stands for Disk Operating System.

Now the designer of an operating system is faced with several problems, such as "Well, I've got to set aside some of the memory for the BIOS, some for the screen image, some for working areas, some for...". Thus, a goodly chunk of the available memory is gobbled up just to provide the operating system. So, it was decided to pre-assign all of the memory above the 640K mark for system functions, which left only 640K (maximum) for user programs (less actually, because some of the lower memory was needed as well).

Most early PCs, however, didn't come with 640K - 128K and 256K were common. More memory was added by supplying it on a board, and plugging this into the bus. If you buy a PC today, you'll more than likely get the 640K thrown in.

Even having an operating system still isn't enough. The bare bones PC comes with the operating system, and some version of BASIC, so if you want to do anything, you still have to start programming, except that now we have BASIC as well as assembly language. We do have something a bit more, because MS-DOS also provides assembly program interfaces to some high level functions that the BIOS doesn't provide (e.g. string print, string input, search directories and so on). What we need is for somebody to write some useful software for us using these interfaces, so that we don't have to be bothered. Well, of course, this has happened. Oodles of software has been written to run under MS-DOS. Sometimes the software writers write "ill-behaved" programs, because, instead of using the MS-DOS interface for some things, they make direct calls to the BIOS, and as we discovered above, the IBM BIOS is copyrighted, so such software won't necessarily work properly on compatibles.

The 8088/8086-based machine has enjoyed a tremendous career. Because of the wealth of software written for it, it is still a useful purchase for some businesses. Soon, though, Intel developed the 80186 processor. This was really only a faster version of the 8086. It integrated a lot of the 8086 chip set into the one processor, and added only a few extra instructions. The 80186 didn't find its way into many PCs, but still survives as the processor in some intelligent workstations, communications controllers and real-time systems.

The real change came with the 80286 processor, with added instructions, and most notably, the ability to address up to 16 megabytes of memory. So, the PC/AT (Advanced Technology) was born. Now we even have the 80386 & 80486 processors. None of this helps too much (apart from speed) when DOS can still only give you 640K of usable memory. This can be overcome with 3rd party EMS (Extended Memory Systems).

PCM

Pulse Code Modulation. A common way of converting analogue signals to digital signals for voice applications. It usually takes 8,000 samples per second, each sample being 8 bits (giving 256 levels for each sample). This then requires a transmission bandwidth of 64Kbps.

peripheral

A general term used for the pieces of computer equipment that are outside of the main CPU, such as printers, disk drives, some communications equipment, terminals, and the like.

pixel

An image is a computer representation of a picture. Images are displayed on a computer screen by representing the screen in memory as a fine grid, or matrix, of values. Each point on the grid is referred to as a pixel (picture element), and the value of the pixel determines the intensity and colour (or grey scale value) of the corresponding point on the screen. In its simplest form, a pixel is a single number with a value of 0 (for white) or 1 (for black). A more complex pixel could be a vector (r,g,b) of three numbers, representing intensities of red, green and blue, respectively. See "image processing".

port

A generic term for some point of input/output between a computer and the outside world. PCs have several ports, and under MS-DOS they have various names. For example, the COM1: port is a connection at the back of the PC for connecting to what is called a serial line - e.g. a plotter, or the telephone system (through a modem).

"port" is also used as a verb, and refers to the process of taking an application that was built on a given computing platform and making it run on another platform. Quite often, the term is used for porting applications between versions of UNIX, highlighting the fact that not all UNIX versions are compatible.

POTS

Plain Old Telephone System.

program

What programmers write. A series of instructions to a computer to tell it how to make some sort of calculation, or how to manipulate some piece of data.

protocol

A protocol is simply an agreed set of rules with which consenting parties can exchange information.

prototyping

Refers to the development of a (probably scaled-down) "look-alike" version of a proposed computer system, so as to determine as quickly as possible the feasibility of the system, and to refine the system requirements and man-machine interface. In some instances, the prototype system then grows into the actual system. In other instances, the final system might need to be developed for a mainframe, but the prototype system is developed using PC tools (e.g. using a PC database system such as Dbase IV). Some mainframe development products actually include facilities for doing development work on a PC, so that the work done to develop the prototype can be used to move up to the final system.

PSTN

Public Switched Telephone Network.

PU

Physical Unit. This is IBM terminology in its SNA world (SNA = Systems Network Architecture). It generally can be considered as referring to a communications controller of some sort (e.g. a front-end or a terminal controller).

Q

QDU

Quantization Distortion Unit. A measure of the distortion introduced by digitizing a voice signal. Please refer to the entry under "digitized voice".

QMF

This is an IBM product - Query Management Fac%
ty - a facility for making it easy to query databases.

QPSX

Queued Packet Switch eXchange. A fast packet switching technology. See "MAN".

queue

Queues are used a lot in computing. This is generally because there is only one (or at least only a few) process available to perform certain tasks, so the inputs are queued to the process. There are different types of queues. A FIFO queue is the most familiar - First In First Out. Then there's the LIFO queue - Last In First Out. This is referred to as a "stack". A LILO is a FIFO and a FILO is a LIFO.

query

A general term for requesting information from a database.

R

RAM

See "MEMORY".

raster

A horizontal row of pixels. See "pixel" and "image".

remote sensing

The study of the earth using sensors on remote platforms (e.g. satellites). The data is usually presented as images.

RISC

Reduced Instruction Set Computer. By reducing the scope of the instruction set, the electronics can be simplified, with a concomitant improvement in the computer's speed. The trade-off is that the instruction set is then only suitable for a particular area - e.g. the computer might be suitable for intensive computations, but at the expense of efficient text manipulation.

RJE

Remote Job Entry. The ability to submit batch jobs from, and receive output at, a remote site.

Robotics

An area of Artificial Intelligence sometimes referred to as smart robots. This area addresses the use of robots in a continually changing environment. Robots are programmed to "see" and "feel" changes in their environment as they move about, and are able to respond to these new spatial references.

ROM

See "MEMORY".

RS-232

A definition of pin functions and circuit functions for serial (bit-by-bit) communications between two devices. Limited to 20Kbps.

S

Satellite

A communications processor in (sometimes geostationary - i.e. its position remaining fixed relative to a fixed spot on the earth) orbit around the earth that permits physical communication over large distances. Good for some things (e.g. television), but because of the inherent delay in travelling such long distances (of the order of a quarter of a second), it's not so good for other things (e.g. interactive character-mode traffic).

SCSI

Small Computer System Interface. A standard interface for connecting Input/Output (I/O) devices to a computer (e.g. magnetic disk drives, optical disk drives, magnetic tape drives, engineering devices). The interface conforms to ANSI standard X3.131-1986.

SDLC

Synchronous Data Link Control. This is a Layer 2 protocol for sending lumps (frames) of data around. This is the protocol used by SNA. Its international standard companion is HDLC (H=Higher), as used by X.25.

sensor

A device for gathering data from the real world, for example radar, kinetheodolites (optical trackers). The word is usually used to refer to a technological device that has an electrical output, as opposed to such things as barometers and thermometers.

Sensor Fusion

Refers to the joint processing of data from different sensors or different types of sensors so as to produce more accurate and reliable data about the object or process being sensed. For example, a rocket might be being tracked by a combination of sensors including radar and optical-tracking devices. Moreover, it might be moving out of the range of some sensors into the range of others. The processing of all of this sensor data so as to produce an accurate data track of the rocket is the sensor fusion process. See also "Data Fusion" and "Information Fusion".

There is some Con Fusion about all these different fusions. One could imagine, though, a hierarchy where the Sensor Fusion process feeds data into the Data Fusion process which in turn feeds information into the Information Fusion process. See also the notes under "data".

serial

Used in communications to indicate that each byte of data is transmitted bit by bit. Most remote communications are serial, but some local communications (e.g. printers attached to PCs) use parallel communications, where each byte is sent in one fell swoop. Parallel communications requires a lot more wires in the cable (or bandwidth in the channel), but is faster.

shrink-wrapped

A term used to indicate the amount of packaging that a software (or hardware) distributor must provide. The larger the number of platforms that the software will run on, the lower the number of particular packaging (wrapping) needed. This saves the distributor money, and so reduces the consumer cost of the software. The trend to Open Systems is a trend to shrink-wrapped products.

simplex

In communications, it means a one-directional transfer of information.

SNA

Systems Network Architecture. SNA was introduced by IBM in September 1973 as its major commitment to communications systems and networks. SNA is a specification describing the architecture for a distributed data processing network. It defines the rules and protocols for the interaction of the components (computers, terminals, software) in the network.

SNA is organized around "domains". A domain comprises a managing host plus the managed network. The host node contains a software set called VTAM (Virtual Telecommunications Access Method) that embodies the System Services Control Point (SSCP). The SSCP is the focal point in the network for managing the configuration, operation and sessions of components within the domain. "Sessions" are logical connections between components in the network. One SSCP looks after one domain.

A domain comprises one or more "subareas". One local subarea comprises the host plus any locally attached terminal equipment. Remote subareas comprise one Communications Controller Node (CUCN, often referred to as a Front End Processor) with cluster controller nodes (CCN) connected in a star fashion to the CUCN. Connected to each CCN are several terminal devices (screens, printers), again in a star fashion. The CUCN software is referred to as an NCP (Network Control Program). Multiple remote subareas are connected in the domain by connecting CUCNs over high speed point-to-point links. A fairly typical domain would consist of only two subareas - the local subarea that contains the host plus a remote subarea where the CUCN is locally attached to the host (over a computer channel) and connected in a star arrangement over a wide area to the CCNs.

End users in SNA networks are individuals and application programs. End users are not considered part of SNA. Instead, a logical unit (LU) acts as an access point into the network. The LU is software or microcode. When one end user (e.g. an individual) wishes to establish a session with another end user (e.g. an application program), and LU-LU session must be established. The LUs provide for any buffers, processor capacity and software required to satisfy the end user requirements.

There are different types of LUs for different types of devices. For example, LU Type 2 is suitable for a screen and keyboard combination, and LU Type 1 is suitable for a printer. Traditionally, such LUs are secondary, the primary LU being the application program. Further, these secondary LUs cannot establish sessions between themselves - for example directly between a screen/keyboard and a printer. This has led to the typical SNA network of terminals connected in a star fashion to a controlling host. LU Type 6 is aimed at providing a general purpose program-to-program distributed capability.

In order to support LU-LU sessions, other underlying sessions must be established. The IBM communications architecture identifies a physical unit (PU) as a hardware control unit such as a CCN or a CUCN, and indeed the host itself. CCNs are PU Type 2, CUCNs are PU Type 4 and hosts (SSCPs) are PU Type 5. An LU-LU session requires the establishment of an underlying PU-SSCP session.

SNA is a layered architecture, and from the standpoint of functions provided to an end user, the OSI model and SNA have many similarities. However, the manner in which the functions are implemented are quite different, so that it is not possible to carry OSI traffic over an SNA network.

software

Computers are capable of carrying out a variety of instructions, such as adding numbers, multiplying numbers, manipulating strings of text and the like. This capability, however, doesn't cause it to actually **do** anything. Someone has to instruct it what to do in order to solve a particular problem. For example, a problem might be only to add up a list of numbers. A more complicated problem might be to analyse a set of numbers and try to determine a formula that adequately represents the set of numbers as a mathematical function. Yet another might be to find all occurrences of the string "ise" in a long file of data. The process of putting together the sequences of instructions that the computer must follow in order to solve a particular problem is called **programming**. The result of the programming effort is a **computer program** that, when run on the computer, performs the desired calculation, or lists out the desired results. Such computer programs are called **software.**

In essence, then, software is the human-generated (and sometimes machine-generated) sequences of instruction that a computer is to follow in order to solve some problem, or to provide some function or service. Software needs to be loaded into the computer, and the computer told to run it. Once this is done, another piece of software can be loaded into the **same** computer, and run to perform an entirely different function. Thus, we can use the same hardware (computer) to run different software packages.

Software Engineering

A disciplined approach to software development. Please refer to the discussion under "CASE" (Computer Aided Software Engineering).

spreadsheets

The advent of the spreadsheet has changed the world. Interestingly, spreadsheets needed the microcomputer (or PC) to be universally accepted. The idea is that there are lots of times when we could organize a particular problem on a big sheet of paper - a worksheet, or, latterly, a spreadsheet. Across the top we would put headings in columns, and down the side we would put particular instances of whatever it was that we were doing. A simple example might be tracking our expenditure. Across the top we would put headings for rent, food, electricity, gambling, rates, and other necessities. In each row (line) we would put how much we spent on these things each fortnight.

Along one row, we might want to add up all the numbers to find out how much we needed for all of those things for one fortnight. Down one column, we might want to add up all the numbers to find how much we have spent on, say, electricity for the last 12 months. We might want to determine the average cost of electricity per month over the year. Perhaps even draw a graph of how our food bills have risen over time.

This is a simple example. Some spreadsheets can become quite complicated. Spreadsheet software makes all of this very easy. Of significant importance is the ability to carry out a "what if" analysis. For example, in the previous illustration, we might want to find out how broke we would be if we increased our food intake (or quality) by about 30%. We can set up the spreadsheet to recalculate all the figures in a few seconds. This ability is of fundamental importance to the value of spreadsheets.

Possibly the first instance of a micro-based spreadsheet was VisiCalc, a product created by one Dan Bricklin in the USA. VisiCalc was soon enhanced with other products like VisiTrend (forecasting - i.e. "what if") and Visi-Plot (i.e. graphs of the numbers). Far and away the most popular spreadsheet package over the last few years has been Lotus 1-2-3. This product combined a lot of the separate "Visi" functions into one integrated software set. Things can go overboard, and a product was released (called Symphony) that was more than just a spreadsheet. It was more of a splatter-it-all-over-the-place-sheet. Things can become too complicated, and Symphony has not enjoyed the same historic popularity as 1-2-3.

SQL

Structured Query Language. An industry-standard language for framing queries to different proprietary database systems. One often hears of the ability to "embed" SQL queries in a standard programming language. This allows a computer program written in COBOL, say, to include database accesses (through standard SQL queries) as part of its processing.

stack

A linear list where the elements are accessed, added and removed from one end only, called the *top*. This implies access on a last-in-first-out (LIFO) basis. The operations push and pop refer to adding and removing elements, respectively.

statistical multiplexer

A Time Division Multiplexer (TDM) splits up the bandwidth into separate time slots and allocates every nth slot to a communications channel - e.g. a 9600bps line can provide 4x2400bps channels by taking data from the first quarter of a second to be for one channel, data from the second quarter to be for the second channel, and so on. The **statmux** is a clever TDM that shares these time slots between the channels, thereby utilizing the quiet periods of channels to service busy channels. In this way, it is possible, for example, to give the appearance of say 4x9600bps channels over a single real 9600bps. A statmux relies on there being quiet periods available in each channel.

statmux

This is short for "statistical multiplexer", and is discussed under that heading.

Supercomputer

A computer that is specifically designed to perform fast computations. Most supercomputers are geared toward "vector processing", which involves floating-point calculations performed on data arrays. If the same operation is to be performed on all elements of an array, then these can be performed in parallel by using multiple processors. For example, adding two 10-element vectors together can be done in the same time as it takes to add two numbers together if 10 ADD processors are used at the same time.

Applications that require complex matrix computations, such as fluid dynamics and visualization, will make good use of highly optimized vector machines.

Many supercomputers use what is termed a "moderately parallel" architecture, with between 2 to 8 processors. Often the multiple processors of a moderately parallel system are used to run whole subroutines of a program in parallel, or even to run multiple programs for different users. A relatively new alternative to moderately parallel systems are the so-called "massively parallel" systems, which can have hundreds or even thousands of microprocessors, each performing a single step in a vector operation.

switched network

Generally means a network that makes a connection between two parties at the time the connection is requested, much like the telephone network.

synchronous

This is where the sender and receiver are synchronized for the transmission of bits and bytes. Asynchronous means that the sender sends bytes at arbitrary intervals.

T

technology

Can represent a particular type of hardware. However, it is more generally used to refer to a process, or set of processes, that use particular techniques, and these techniques could be based on particular hardware items.

The Oxford English Dictionary defines technology as the "(science of) practical or industrial art(s); ethnological study of development of such arts; application of science", which requires another search of the dictionary for "ethnological". The definition "application of science" is probably the most widely used meaning.

The most general meaning in common use is simply *applied science*. The term is frequently used in a more restricted sense:

a) to label different fields of applied science; or

b) to label different phases of development in a field of endeavour.

For example, the current capabilities in ceramics or computer engineering could be referred to as *ceramics technology* or *current computing technology*. In this sense, the word is used to embrace and distinguish a field of endeavour. Alternatively, historical phases of technological development in a particular field could be differentiated by referring to them as "different technologies", meaning different processes or techniques. For example, we talk of the older *small scale integrated circuit technology*, or the newer technologies of *VLSI* or *VHSIC*. Thus, the word is used in various senses, frequently to delimit a particular level of manufacturing sophistication, fabrication style or industrial process.

Telematics

Refers to the area of overlap between the converging technologies of computing and communications. EDI is a telematic technology.

time division multiplexing

A technique for using one high speed line to transfer information from several smaller speed lines. The higher speed line divides its bandwidth into time slots, and each lower speed line is allocated its share of slots.

timer

Timers are used a lot in computers. A counter is set that decrements by one every time it is pulsed by the electronic circuitry. When it reaches zero, the computer generates what is called an interrupt. The interrupt causes the computer to stop what it is doing, and to start running a different program. When this different program is finished, the computer resumes from the point at which it was interrupted.

timesharing

Several programs or processes are made to appear to be running simultaneously in a computer. In reality, the computer is allocating tiny slices of time to each, and so is sharing the time between them. Refer also to "multiprocessing" and "multitasking".

Token Ring

Token Ring is the name given to a scheme, or set of procedures, for connecting nodes into a LAN. Nodes can be anything from PCs to mainframes. It is essentially an IBM development, but is a recognized international standard (IEEE 802.5).

This is a fairly low level scheme, and only covers the bottom two layers of the OSI model. Higher level software is needed to use it. For example, Novell LANS can be based on Token Ring technology.

The scheme is fairly simple. Nodes are connected in a ring. A special piece of data, called the "token", whizzes around this ring. When a node wants to send something to another node, it has to wait until the token comes by. Not only that, the token has to be empty. If the token isn't empty, then two other people are blabbering, so the node has to wait. Naturally, there are rules for relinquishing the token.

Anyway, when a node wants to talk to somebody, it waits until the empty token arrives, then grabs it, and sends a message to whomever it wishes to strike up a conversation. This all sounds a bit tedious, but the thing's running around at 4 Megabits/second (and there is also a 16Mbps version), so a lot of talking can be going on.

There's an ongoing argument about whether the token ring technology is better than Ethernet, or anything else, but they all have their good and bad points. It might seem bad, for example, to have nodes connected in a ring. When one of the nodes crashes, the token can't whiz around, and the whole LAN disintegrates. This problem is largely overcome by connecting the ring physically as a star, through a device called an MAU (Medium Access Unit). Thus, the token always goes back to the MAU after visiting each node. If a node crashes, the MAU simply doesn't forward the token to the node.

Then there are distance considerations. With the bus-based Ethernet, every node has to be able to be heard by all the other nodes, and this imposes distance limitations. With the token ring, each node only needs to send the signal to the next one.

One other important feature of token ring is that it implements traffic priorities. Ethernet has no such thing. This means that there is no way on an Ethernet to differentiate between interactive traffic and batch traffic.

On the other hand, the cabling for a token ring is not as neat as that for Ethernet, because of the physical star topology.

So, there are pros and there are cons. The important thing to realize is that token ring is simply a low level technology for effecting communications in a LAN, and it has some desirable features and some undesirable features.

top-down

This refers to the process of designing a system beginning with the first major component, and splitting this into smaller parts. These smaller parts may then be further split, until arriving at manageable components. This contrasts with "bottom-up", which attempts to design the detail, and then use these as building blocks.

tracing

Refers to the process of recording in a log file the sequence of events that are taking place during the running of a program, or during a communications transmission.

transaction

The entire process from the time an operator enters a request and receives a reply. This usually involves an update of a database. A transaction can involve quite a number of steps, and can even trigger other transactions.

transducer

A device that converts variations of one quantity into those of another (e.g. pressure into voltage).

Trusted Systems

A Trusted System is, essentially, one that you can trust to do what it is supposed to do. We are all familiar with the catch-all excuse from the service-provider "Sorry! The computer made a mistake!". Of course, the computer didn't make a mistake. The software had a bug in it, or was inadequately designed to cope with the situation, and the problem manifested itself in, say, sending you a bill for somebody else's goods.

Quite probably, such errors cause inconvenience, but are tolerable - the mistake was made, but the situation can be fixed. However, if the bug or inadequate design caused a plane to crash and kill hundreds of people, then this is a bit more than an inconvenience, and the results are irreversible.

Therefore, in situations where safety or security are concerned, we want to feel a little more confident that the system is going to work, not only reliably, but also adequately and completely - i.e. with a high degree of "trustedness". The amount of confidence we have in the system can be related to the degree of rigour used in developing the system. This requires the adoption of procedures and practices for the system development.

Clearly, however, the more rigour we impose in the system development process, the more lengthy and costly that process becomes. There needs to be a balance between the safety or security requirements for a particular system and the amount of effort needed to ensure those requirements.

Thus, it is possible to define categories of trustedness. In the US, levels have been defined ranging from D (the lowest level of trustedness) up to A (the highest level). Within these levels are also sublevels, e.g. B1, B2 and B3 (note that B3 is more trusted than B2 which, in turn, is more trusted than B1). In Europe, the levels all begin with E followed by a number, but here, a low number implies a low assurance level, and a high number implies a high assurance level. The levels range fro E0 to E6, and are as follows:

E0: inadequate assurance
E1: informal description of architecture; some testing
E2: E1 + informal description of detailed design; test evidence; configuration control; controlled distribution
E3: E2 + detailed design evidence; check source-security functions
E4: E3 + formal security policy; rigorous architecture; detailed design; vulnerability analysis on design
E5: E4 + rigorous design refinement; check source code against design; vulnerability analysis against source code
E6: E5 + formal methods in security policy model and security architecture with consistency proof

twisted pair

Two copper wires twisted around each other, and used for electronic communication. The wires essentially close an electrical loop between the communicating parties. Twisting them around each other helps to minimize the interference caused by the proximity of the electrical conductors carrying fluctuating currents.

U

UART

Universal Asynchronous Receiver/Transmitter. It is a bit of circuitry for converting bytes to strings of bits for serial transmission, and for the reverse process at the other end of the line done by another UART. There is another thing called a USART, which is a later invention for doing the same thing to synchronous and asynchronous traffic.

UHF

An acronym for Ultra-High Frequency. UHF is a band in the radio frequency spectrum (300MHz - 3GHz) utilized for radars and radio relay microwave communication systems.

UNIX

Just as MS-DOS is an operating system for PCs, so UNIX is an operating system that is very popular in the minicomputer environment. These days, UNIX is available across the range (micros to mainframes). They are not necessarily all compatible, though.

UNIX was developed by AT&T's Bell Laboratories in the late 1960s as a simple multi-tasking, multi-user operating system that emphasized a modular approach to its design. UNIX is mostly written in the C programming language, as opposed to a lower-level machine language, thereby making it easier to port to (i.e. implement on) different kinds of computers (or different "platforms"). The main requirement is that the computer system have a C compiler. However, low-level device drivers still need to be written in a lower-level language.

At first, UNIX was only used internally at Bell Labs. Its initial commercial acceptance was slow. It was given away to universities, however, and it quickly became popular in the academic world. AT&T developed several versions, including UNIX Versions 5-7, UNIX System III and the current UNIX System V.

Perhaps the key to the commercial acceptance of UNIX was the work done on it at the University of California at Berkeley. Berkeley provided network support for UNIX, extensive support for peripherals, and a good software development environment. Berkeley's versions are labelled BSD (for Berkeley Software Distribution), followed by the revision number (e.g. 4.2, 4.3).

These Berkeley enhancements made the BSD version popular on engineering workstations from groups such as Hewlett-Packard and Sun Microsystems. In 1985 AT&T, noting the increasing popularity of BSD UNIX, tried to regain control of the market by establishing the System V Interface Definition (SVID), the goal of which was to provide a common base for developing UNIX applications. The US Government then stipulated that any UNIX-based computers it bought had to meet this standard.

AT&T then entered into an agreement with Sun Microsystems, and formed a group called UNIX International in order to steer these two versions along a common path. However, a consortium of major computer companies, including IBM and DEC, were worried that such a major standard operating system be controlled by one group, so the consortium formed its own group called the Open Software Foundation (OSF), its objective being to develop a standard version of UNIX and an open platform for distributed computing. This has resulted in the development of OSF/1, the OSF version of UNIX.

Fortunately, there are signs that this fragmentation of versions will gradually come together with the major players beginning to agree on operating system interface standards (POSIX, Intel Binary Compatible Standard (iBCS), X/open Transport Interface (XTI)). It is through these agreements that UNIX will build up its strength, and that the nascent "shrink-wrapped" UNIX applications software market will flourish.

update

Bring up to date. Tends to be used a lot for "make a change" rather than "bring up to date".

UPS

Uninterruptable Power Supply. A power supply with battery backup to cater for mains power failures.

user-centred design

Please refer to the entry under **Human Factors**.

utility

Refers to a useful piece of (usually housekeeping) software. See also "Decision Support Systems".

V

V&V

See "**Verification & Validation**".

vapourware

Refers to products or services claimed to exist by a provider, but which either don't exist, or have not been completely developed.

variable

Usually applied to the name allocated to a memory cell by a program. The name stays the same, but the contents vary. Loosely related to a mathematical variable, but is more specific in that it tags something fairly tangible inside a computer.

VDM (Vienna Development Method)

An extremely precise specification language used to generate trusted systems. VDM is a rigorous system specification technique based on the formal mathematics of set theory and predicate calculus. Using VDM, the essential features of a system can be abstracted into mathematical formulae. Once in this form, prototyping can be performed in the laboratory, and theorems can be proved about the system. At each stage, the specification can be proved to be a valid statement of a given set of requirements.

The language and method are both in the public domain, and have been described in at least four books. There is now a substantial body of real industrial experience in the use of VDM, and this experience probably exceeds that of any other formal specification and design technique (cf. "Z").

VDU

Visual Display Unit. Sometimes also VDT - Visual Display Terminal.

Verification & Validation

A generic term for the complete range of checks that are performed on a system in order to increase confidence that the system is suitable for its intended purpose. This range might include a rigorous set of functional tests, performance tests, reliability tests, and so on. Although a precise distinction is not always drawn, the *verification* aspect normally refers to completely objective checking of conformity to some well-defined specification, while the *validation* aspect normally refers to some assessment of likely suitability in the intended environment.

VHF

An acronym for Very High Frequency. VHF is a band in the radio frequency spectrum (30MHz - 300MHz) used for mobile radio telephone, television, FM broadcasting and line-of-sight communication systems.

virtual

This is a much used word in the computing world. Its main use is for something that is emulated. So, we have virtual circuits, for example. These appear to be dedicated circuits to the end points (the two devices that want to communicate), but in fact share a communication line with lots of other "circuits". Then there's the virtual terminal. This is a piece of software that appears to be a terminal to an application program. The application program doesn't know - as far as it is concerned, it is talking to a real terminal.

Virtual Reality

This term began in a context similar to that of virtual memory. Virtual memory extends the real memory by using disk space. The program that is running is not aware of this, except that it probably runs more slowly. Similarly, virtual reality extends reality. For example, we could capture an image of a bald-headed person from a photograph into a computer graphics system, and, using the graphics software, enhance the image by adding a full head of hair. An outsider looking at the original captured image and the enhanced image would not be able to distinguish whether one image had been enhanced or not (presumably the original could have had hair, and the "enhancement" process could have removed it).

The term "virtual reality" now includes the so-called three-dimensional multimedia technologies. A research leader in this field is NASA's Ames Research Center, which is developing head-mounted equipment that places the operator into a 3-D virtual environment of computer-generated images, the view changing according to the operator's voice, orientation and gestures. Thus, the operator enters and controls a "virtual reality".

An interesting sidelight emerging from this technology concerns ethical practices. With virtual reality, people can be put into places where they have never been, and objects can be made to take on lifelike qualities they could never actually have. If we develop an application based on virtual reality, should users be told which parts of what they are seeing are contrived and which are real? What standards of behaviour should be applied to two or more users interacting in an artificial space? The rules of law and the traditional rules of nature are not clearly defined, partly because computer technology allows computer users to do things that are impossible in reality. The potential for deception is considerable.

VLSI

Very Large Scale Integration. VLSI is a modern printed circuit assembly design technology that enables very high component packing densities to be achieved in manufacture, and contributes to the progressive miniaturization of electronic hardware.

VMS

The name of Digital's premier operating system on VAX computers.

volatile

This refers to the characteristic of a computer's or device's memory not retaining what was stored in there when the power is switched off. The alternative is "non-volatile" - i.e. the information is retained after power loss.

volume

Usually means the physical thing that holds data, for example a magnetic tape, or a disk pack. Thus, we can have multi-volume files, or multiple files per volume. We can also have virtual volumes (see above). A volume is really the collection of information on a particular medium. Thus, what's on one magnetic tape is one volume, and what's on another is another.

V.24

One of several CCITT V-series recommendations specifying the electrical and physical interface between data terminal equipment and data circuit-terminating equipment. V.24 is used for connections up to 19.2Kbps. V.35 lets you go faster.

W

walkthrough

A term used for stepping through a program or procedure (not necessarily computer oriented) in a group situation to highlight any errors or improvements.

WAN

Wide Area Network. As a general rule, a LAN (Local Area Network) would exist within a building, a MAN (Metropolitan Area Network) would exist within a city or town and a WAN would traverse a country (and beyond). LANs would be interconnected using MANs and MANs would be interconnected using WANs. Again, as a general rule, LANs would operate at a higher speed than MANs and MANs would operate at a higher speed than WANs.

These distinctions are not at all clear cut. For example, one could interconnect two LANs across the country using bridges and point-to-point high-speed links. This "extended" LAN is seen as a single LAN to the users, even though it spans the nation. Possibly a more meaningful distinction can be made by considering the protocol used to implement the network. For example, if the protocol is one usually limited to LANs (e.g. Ethernet or Token Ring), then it should probably only be considered a LAN, irrespective of the actual distances involved. Similarly, MANs might be identified by their use of specifically MAN-oriented protocols (e.g. fast packet-switching), and WANs might be identified by their use of specifically WAN-oriented protocols (e.g. X.25)

Whetstone

A benchmark loop program originally developed in ALGOL in the 1960s, and now available in FORTRAN, C and other languages. It is a generally fair benchmark of computation-intensive performance, but one that has been a favourite target of optimizing compilers.

white box

You've heard of the black box. Well, a white box is a black box, but you know how it works internally. For example, a subroutine can be both a black box and a white box. It is a black box to the routines that use it. They do not know, or need to know, how it performs its function, only that it does so unfailingly and according to the specification. It is a white box to, for example the maintenance staff, as its precise operation is documented.

widow

It is the last line of a paragraph that appears by itself at the top of the next page. When the first line of your paragraph appears by itself at the bottom of a page, that's an orphan. Some word processing programs have the option of avoiding these with no further effort.

wildcards

Symbols that you key in as part of your command, that represent any substituted character or string of characters. For example, in MS-DOS, the asterisk (*) is used for any string, so that a command such as "DEL DIC.*" will delete all files whose name is DIC with any extension.

window

A (usually rectangular) section of a computer screen that is used to display information from another application, or another section of a program. "Windows" is the name of a Microsoft computer operating environment that sits on top of DOS, and that allows the operator to perform functions on multiple applications using windowing techniques.

word

A very early computer term that has stuck. Earlier computer architectures were constructed around the "word", which was a collection of bits that the computer operated upon as an entity. The bits were split up in various ways to represent numbers. Nowadays, most computers operate to manipulate bytes. Several bytes are concatenated into words for arithmetical operations.

worksheet

See "spreadsheet".

WYSIWYG

Means "What You See Is What You Get". A term used by word processing packages to indicate that what you see on the screen is exactly how it will appear on the printed page.

X

Xmodem

A computer program for transferring data between computers using a protocol that includes some error checking.

XON/XOFF

A type of flow-control mechanism for data transfers where the receiver (e.g. a printer) tells the sender (e.g. a computer) to XOFF, because its buffer is jam packed. When there's some room available, the receiver tells the sender to XON again.

X-Windows

X-Windows is an operating environment between an X-terminal and one or more X-applications. It comprises a system for providing a windowing environment (including mouse support) for terminals attached to major shared applications (as opposed to one-to-one use on a PC).

X.25

X.25 is one in a series of recommendations, the X-series, as promulgated by the CCITT (International Consultative Committee for Telegraph and Telephone). The 25th one describes the protocols and standards for implementing what is called a Packet Switching Network (PSN). There are others, like X.21 referring to a particular electrical interface and X.121 referring to an addressing scheme.

The rationale is that if everybody implemented their PSN according to the recommendations set out under X.25, then they would all be able to talk to each other. By and large, this is the situation today.

The standard telephone network is termed a "Circuit Switched Network". When you ring someone up, the Telecom network finds a path between your telephone and that of the number you dialled. This is called setting up a circuit through all of their switching gear. Once connected, that circuit is used exclusively for your conversation. Nobody else can use it. The telephone network gets used for data transmission between computers in much the same way as it is used for voice.

Packet switching is a bit different. There is only one actual physical path or circuit between two places, but lots of pairs of people can set up their own "virtual circuits (VCs)" between themselves, using the same single physical circuit. This is done by splitting up each pair's data into small packets (usually about 128 bytes each), and attaching a label to each packet to identify the virtual circuit (or conversation) to which it belongs. (This label, incidentally, is called a logical channel.)

At each end of the physical circuit (called an X.25 trunk) is a small computer called a packet switch. Lots of people can be connected to the trunk. The packet switch looks at each packet and says "right, this one's for you, this one's for you, Oh and here's two for you over there, this one's for - I don't know who this one's for, so I'll send it back, this one's for you", and so on. In this way, the single physical circuit can be used to carry on conversations with several pairs of parties using virtual circuits. Obviously, you can extend this concept to having lots of packet switches at various centres (capital cities, for example), with X.25 trunks between them. This is exactly what Telecom has set up in the form of AUSTPAC.

You will often hear the terms "Switched Virtual Circuit (SVC)" and "Permanent Virtual Circuit (PVC)". A PVC occurs when you assign a label, or logical channel number (LCN), permanently to a pair of conversing parties. Nobody else can use it. When you switch on your terminal or computer, you are automatically connected to the other party. An SVC is set up at the time you want to carry on a conversation. Here, you are allocated an LCN at the time you make the call. When you have finished, that LCN is free to be used by other SVCs.

X.25 networks are intended for low to medium volume usage. This enables single lines to be shared by numbers of people. For high volume use, there isn't any spare capacity to share with other people. They are also not very good for dumb terminal interaction where the host must echo each keystroke (as is required by some word processing packages, for example). Every time you hit a key, the network has to wrap that single character up into a packet, send it across the network, wrap the single character response up into a packet and send it back to be displayed on the screen so the user knows that the system has accepted the input.

X.400

The term "X.400" is used generically to refer to a **series** of CCITT recommendations for what is referred to as a Message Handling System (MHS). The X.400 recommendation itself is only the System and Service Overview. Other recommendations in the 400 series flesh this out. For example, X.402 covers the overall architecture, X.403 covers conformance testing, X.407 lays out some conventions for specifying the recommendations, X.420 covers the Interpersonal Messaging System (IPMS), and there are others in between.

In a similar way, the term "X.500" refers to a series of recommendations concerned with directories, and how they are structured and administered. X.500 itself is only the overview. X.501 covers organizational and security models, X.509 deals with authentication, X.518 covers procedures for distributed operations, and so on.

Two aspects of X.400 are of significance:

_X.400 is not just an electronic mail system. X.400 describes an MHS, and, whilst an MHS can be used to implement an electronic mail system, it can also be used to implement a wide variety of other types of systems. These implementations are done through the development of specific pieces of software called User Agents (UA). In fact, included in the current X.400 recommendations is a specification of a UA for the interchange of what are termed "interpersonal messages". The system is referred to as the Interpersonal Messaging System, and its early appearance in the recommendations probably accounts for X.400 being labelled as an electronic mail system. An IPMS is a particularization of an MHS.

_X.400 describes a series of OSI applications and OSI application services. This point is important, because an X.400 implementation needs an OSI network for communication, and this includes not only a lower level OSI network (e.g. X.25), but also the upper layer OSI services as well (Transport, Session, Presentation, Association Control Service Elements - the reader is referred to the entry under "OSI"). This fact is in contrast to the EDI (Electronic Data Interchange) standards, which do not stipulate the actual document interchange mechanisms, only the structure and sequences of exchange. Thus, an EDI system could be implemented over SNA, which is not OSI. Clearly, however, sending EDI messages over X.400 is an attractive alternative, and future X.400 developments will help to facilitate this. For example, the X.435 standard is being developed to define a new protocol (dubbed "Pedi") for sending EDI messages over X.400. Pedi supports end-to-end acknowledgement and authentication of sender and receiver.

An MHS comprises two fundamental components - User Agents (UAs) and Message Transfer Agents (MTAs). A UA is a piece of software that interacts with the user for the purpose of exchanging messages with other UAs (of the same **class** - e.g. IPMS-UAs). This software could be implemented in a PC, or could be provided as part of a larger system's X.400 capability, and accessed through a terminal. Essentially, the UA helps to prepare a message for "posting", including putting the message into an envelope. An MTA is akin to the Post Office. It accepts the message from the UA for delivery to the recipient UA. This might require passing the message on to other MTAs (Post Offices) for delivery. A group of such MTAs is referred to as a Message Transfer System (MTS), and when the set of UAs is added, we have a complete MHS.

The concept of **management domains** is a significant one when considering the implementation of an MHS. The current MHS standards distinguish between domains managed by public authorities (e.g. Telecom and/or OTC) and those managed by private organizations. A public management domain is referred to as an Administration Management Domain (ADMD) and a private management domain is referred to as a Private Management Domain (PRMD).

Y

Y

Short for "YES". Often, in the interaction between a computer and a user, the computer needs a simple YES or NO answer to a question. The user responds by hitting the "Y" or "N" key (or just by hitting the "ENTER" key to accept the default).

Z

Z

An extremely precise specification language used to generate trusted systems. Z is a rigorous system specification technique based on the formal mathematics of set theory and predicate calculus. Using Z, the essential features of a system can be abstracted into mathematical formulae. Once in this form, prototyping can be performed in the laboratory, and theorems can be proved about the system. At each stage, the specification can be proved to be a valid statement of a given set of requirements.

Z is in the public domain, and is still under development. It is similar in many respects to VDM (see entry under "VDM") but is a later development. Z is suitable for data-oriented applications, but not control applications since it cannot deal easily with concurrent tasks.

ZAP

ZAP is often used to indicate a clearing out of something (e.g. a database), or an erasing of something (e.g. a file). One other well-known use is a binary patch to an object or load module, pending the formal software change that fixes the problem. (A "binary patch" occurs where the machine language version of a program - which is just a lot of 1s and 0s - is modified before the program is run.)

Department of Defence

DOCUMENT CONTROL DATA SHEET

Page Classification	UNCLASSIFIED
Privacy Marking/Caveat	N/A

1a. AR Number	1b. Establishment Number	2. Document Date	3. Task Number
AR-006-927	ERL-0594-GD	January 1992	

4. Title

INFORMATION TECHNOLOGY IN PLAIN WORDS

5. Security Classification

Document	Title	Abstract
U	U	U

S (Secret) C (Confi) R (Rest) U (Unclass)

* For UNCLASSIFIED docs with a secondary distribution LIMITATION, use (L)

6. No. of Pages: 70

7. No. of Refs.: -

8. Author(s)

A. RATTRAY

9. Downgrading/Delimiting Instructions

N/A

10a. Corporate Author and Address

Electronics Research Laboratory
PO Box 1500
SALISBURY SA 5108

10b. Task Sponsor

-

11. Officer/Position responsible for

Security..................N/A..

Downgrading...........N/A...

Approval for Release........D/ERL.................................

12. Secondary Release of this Document

APPROVED FOR PUBLIC RELEASE

Any enquiries outside stated limitations should be referred through DSTIC, Defence Information Services, Department of Defence, Anzac Park West, Canberra, ACT 2600.

13a. Deliberate Announcement

NO LIMITATION

13b. Casual Announcement (for citation in other documents)

- [√] No Limitation
- [ ] Ref. by Author & Doc No only

14. DEFTEST Descriptors

Information technology, Computers, Word lists
Terminology

15. DISCAT Subject Codes

1210

16. Abstract

This document is a glossary of IT-related (Information Technology-related) terms directed specifically at the non-specialist reader. It is not for scientific professionals in any specific area, although their input has been invaluable in compiling this work. Further, it is not a formal Defence glossary in the sense of providing approved formal definitions of terms, and does not replace such definitions. This is a supplementary, and to some extent complementary, work.