Chapter 31 - Data communications

31.1 Introduction to I/O ports

A port is a place in your computer where you can attach peripheral devices such as printers, external CD RW devices, speakers, microphones, modems and so on. They are the places where data can enter and leave the computer system.

31.2 The serial port – an overview

PCs commonly have 2 serial ports. Look on the back of a computer for an elongated D-shaped socket with 9 pins in it, about 2 cm long. That is a serial port. You can connect devices that use serial communication to these ports.

Examples of devices you might typically connect include modems, mice and keyboards. Serial ports send and receive data **one bit at a time** so you only need two wires for communication (one wire to send data in a serial manner and a second wire to receive data in a serial manner). There are, however, some other wires to control the way two devices communicate, to control the 'handshaking'. By having 2 wires for communication, you can communicate in two directions, in and out of the computer, at the same time (known as **duplex transmission**). If you only had one wire then you could send and receive data but you couldn't do this at the same time (known as **half-duplex transmission**). If you want to send a byte of data serially, then it would take 8 times as long as doing it using parallel transmission. This is because you send each ASCII character one bit at a time. With parallel transmission, you send each bit that makes up the ASCII character in one go.

31.2.1 The RS232 protocol

RS232 is a very common communication protocol (a set of rules that governs how communication will take place) for sending data using serial communications between **Data Terminal Equipment (DTE)**, such as your computer or a phone, and **Data Circuit-Terminating Equipment (DCE)**, such as a modem. The 9-pin D connector serial ports we just discussed use the RS232 protocol to send and receive data. There is also a bigger D connector with 25 pins. This also uses the RS232 protocol to manage the serial transfer of data, although it is not so common now. This protocol is used for two types of serial communication known as '**asynchronous communication**' and '**synchronous communication**'.

31.2.2 Asynchronous communication

In this kind of communication, the data is sent intermittently! You never really know when the next data byte is going to be sent. For example, the computer keyboard sends bytes of data to the computer using serial communication. The computer does not know when the next key press will actually happen. For this reason, asynchronous communication is suited to **low-speed applications** such as in the example just given, mouse to computer communication, computer to MIDI keyboard communication or modem communications.

31.2.3 An example of asynchronous communication

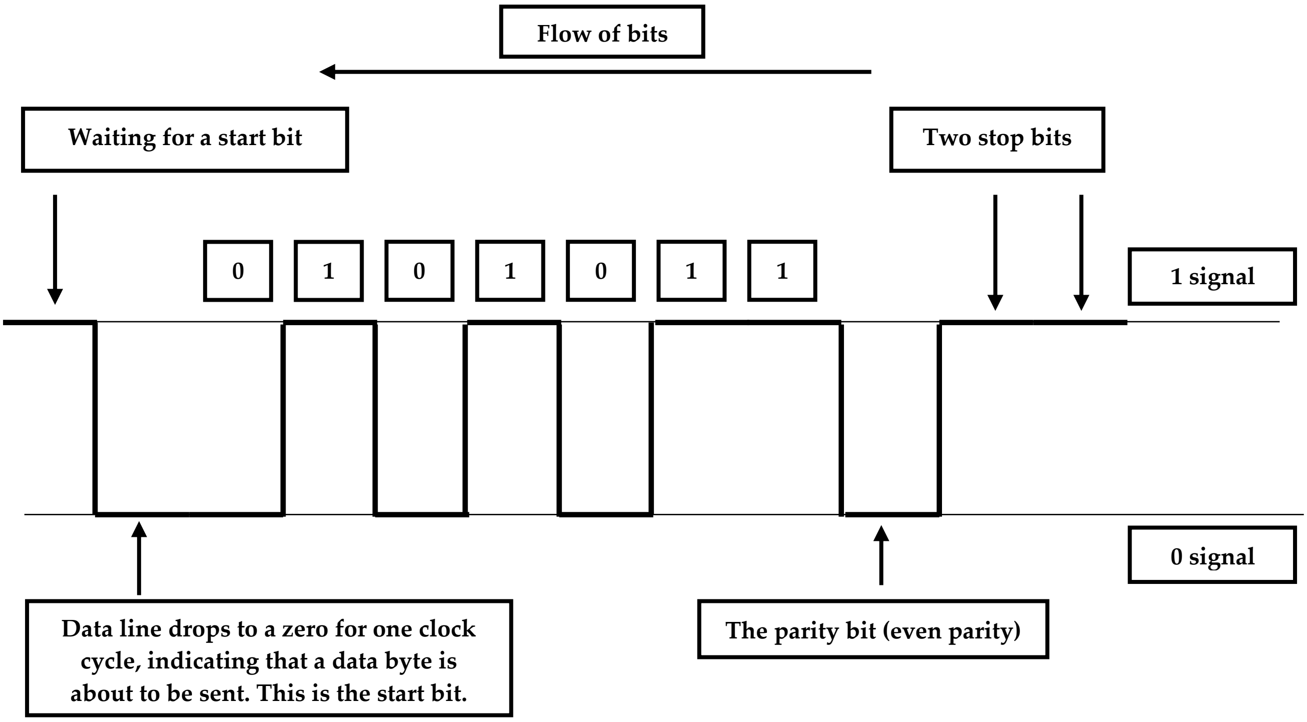
Think about what happens when a key on the keyboard is pressed.

- Because the computer never knows when it is going to receive a byte (another press of the key), it must have a way of 'knowing' that some data is about to be sent (as opposed to it receiving data signals that could be generated by interference, for example).
- To do this, a **start bit** is sent by the keyboard just before the data bits are sent. What this actually means is that the keyboard puts a **0 signal** on the data wire for one clock cycle. The reason it puts a 0 signal on the data communication wire is that whilst **no data** was being sent, **a 1 signal** was on the data line, to indicate that there was no communication taking place.
- When the receiving computer sees the signal on the data wire drop from a one signal to a zero signal for one clock cycle, it knows that (for 7 bit ASCII) the next 7 clock cycles will contain the bits that make up one byte of data. During each clock cycle, the keyboard will either put a 1 signal on the data line or a zero signal on the line, depending upon the ASCII code for the key that was just pressed. The computer is 'collecting' these bits and also counting off 7 clock cycles.
- After the data bits have been sent, the keyboard then sends a **parity bit**. The computer uses the parity bit to check if the data was sent correctly. This is discussed in more detail later in this chapter.
- The parity bit could be either a 0 signal or a 1 signal.
- After the parity bit is sent, a **stop bit** is sent. This stop bit is a 1 signal. It signals to the computer that the byte of data has now been completely sent. The stop bit can actually be either one clock cycle or one and a half clock cycles or two clock cycles in length.

- The computer now waits, watching for when the data line drops down to zero again for one clock cycle and the whole process is repeated.

31.2.4 An example of sending a letter

Consider the example of a keyboard sending the letter j to the computer. The 7-bit ASCII code is 1101010 for this letter. Note that the **least significant bit** is sent first and the most significant bit is sent last!! That means the right hand bit gets sent first! The ASCII code would be sent in the following way:



Sending the letter j using asynchronous communication.

If a computer used 2 bits for a stop bit then to send one character using 7-bit ASCII requires 1 start bit, 7 data bits, 1 parity bit and 2 stop bits, or a total of 11 bits. Each character requires 11 bits to send it!! It is worth noting at this point that there are lots of combinations of bits that could be used for asynchronous communication. For example, you could use one start bit, seven data bits, no parity bit and one stop bit, a total of only 9 bits per character. Two devices that want to communicate must agree in advance whether they will use 7-bit or 8 bit ASCII, whether they will use odd or even or no parity and how many stop bits they will use. They must agree the **communications protocol**.

Clearly, asynchronous communication is not very efficient if you need to transfer blocks of data because of the amount of overheads (the number of start, parity and stop bits) that have to be sent in total. If you need to send large, predictable blocks of data then asynchronous communication is not the best method. An alternative method of serial communication called 'synchronous communication' should be employed.

31.3 Synchronous communication

In this type of communication, data bytes are sent in blocks in a steady stream. There is no need to send a start, parity and stop bit for each ASCII code. You simply collect together a block of bytes, add a small amount of data to the beginning and end of the block to indicate the start and finish of the block, add some error checking bytes (called a 'check sum') and send the data in one go. We can represent how a block of data is made up with the following diagram:

Header	**A block of data bytes**	**Tail**	**Check sum**

Sending a block of data.

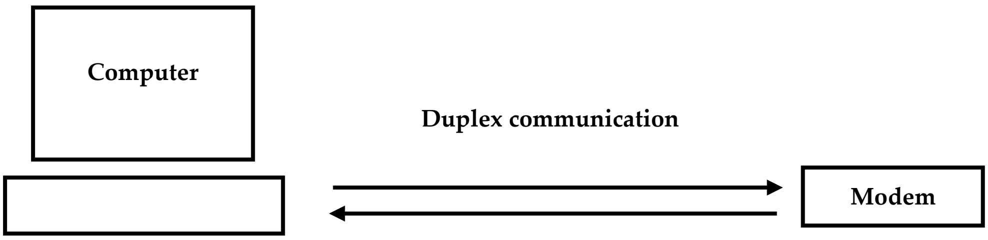
Sending large amounts of data, then, is faster using synchronous communication than asynchronous communication because there are fewer overheads. In addition, you can add 'checksums' to blocks of data. These are carefully calculated bytes, sent with the block of data, which can be used to pick up nearly all of the transmission errors. Compare the rate that errors can be picked up using parity checking in asynchronous communication – only a 50% error detection success rate!

31.4 Serial communication 'handshaking'

Before two devices can communicate, whether synchronously or asynchronously, they must ensure they are both ready to communicate, that the DTE (such as the computer) is ready to send and the DCE (such as a modem) is ready to receive, or vice versa. How they do this is referred to as 'handshaking'. There are two types of handshaking that could be used. The first is **hardware handshaking** and the second method is **software handshaking**.

31.4.1 Hardware handshaking

Consider the following situation that uses duplex transmission.



Sending data using duplex communication.

31.4.1.1 Sending data from the computer to the modem

1) If the DTE (the computer) wants to send data to the modem and is ready to do so, it sends the modem an RTS signal ('Request To Send') and a DTS signal ('Data Terminal Ready'). Each of these signals has its own wire in the RS232 serial communication cable.
2) When the modem sees an RTS signal, it sends a DSR (Data Set Ready) signal to tell it that it is switched on.
3) The modem then sends a CTS signal. This is a 'Clear To Send' signal that indicates to the computer that it is okay to send data. This is another wire inside the RS232 cable, used solely for the CTS signal.
4) The data is then sent one bit at a time down the TD wire, or the Transmit Data wire in the cable.

31.4.1.2 Sending data from the modem to the computer

1) The computer sends the modem a DTR signal. This is a Data Terminal Ready signal on its own wire and signals to the modem that the computer is ready to receive data.
2) The modem sends a DSR signal to the computer when it sees a DTR signal. This is a Data Set Ready signal and tells the computer that the modem is connected and ready to send data.
3) The modem then sends data to the computer on the RD wire, or 'Receive Data' wire, one bit at a time.

31.4.2 Software handshaking (XON / XOFF)

Rather than having lots of extra wires in an RS232 cable (one for the DTS signal, a different one for the RTS signal, and so on), there is nothing stopping you using **software handshaking** for data communication. Special ASCII codes are sent between devices to start and stop data transmission. In fact, you only need two special codes:

- 'Start the data transmission' - **X-ON**
- 'Pause the data transmission' - **X-OFF**.

Having fewer wires to worry about is nice but it is not all good news. You need to send the special characters, which may mean that you have to send more data than if you used hardware handshaking!

31.5 The Centronics Interface parallel port

If you have a printer connected to a PC, the socket where the printer cable is actually connected to is known as the 'Centronics Interface parallel port'. It looks like an elongated D-shaped socket about 4 cm long with 25 'pins' in it. Eight of those pins are used to carry electrical signals from the PC that together correspond to one 8-bit ASCII code. When the computer wants to print a file out, it must send out a stream of these ASCII codes, one after another in a coordinated manner using **handshaking**. The

PC puts the first byte that it has to send on the eight data lines and sends it. The other pins in the 25-pin socket are used to control how the bytes are sent. Once one byte is sent, the next one is sent by putting it on the data lines and using the control lines to effect the transfer. Each set of 8 bits is sent in one go, which is why the Centronics interface is known as a 'parallel communications' interface, as opposed to RS232, which is a one-bit-at-a-time protocol.

31.5.1 Parallel communication 'handshaking'

Consider a typical situation where parallel communication is used, connecting a printer to a computer. One method of handshaking is known as 'polling'.

1) The computer puts a byte of data on the 8 data lines of the Centronics interface..
2) The computer then looks for a signal from the printer (on a control wire in the Centronics cable) that says "I'm not busy so you can send the data now".
3) The computer then sends a signal on another wire to say "Take the data now".
4) The printer takes the data and sends (on yet another control wire) a "Thank you. I have the data now" signal to the computer.

31.6 Drivers

Of course, you cannot just connect any printer to your computer and expect it to work straightway! You must first load the **printer driver**. A driver is a special piece of software that helps your computer successfully communicate with the printer. In fact, when you buy any new peripheral device, you more often than not get a CD with it. This has the driver on it and you have to spend 2 minutes loading it up before you can use that particular piece of hardware.

31.7 USB ports (Universal Serial Bus)

Most peripherals come a USB port or at least give you the option to have one rather than a serial or parallel interface. This type of interface makes connecting a new piece of hardware to a computer incredibly simple. PCs traditionally came with a very limited number of parallel and serial ports, often just one parallel and two serial ports. These days, they now have many USB ports and you can add extra ones easily if needed.

31.8 PS/2 mice

Although some mice (and keyboards) still use the serial port for connecting to the computer, most now either use a USB port or a PS/2 port. The PS/2 port was designed specifically for the job of connecting mice and keyboards to the computer and thereby freeing up the serial port for more important jobs, such as connecting up a modem. In fact, the PS/2 is often called the 'mouse port'). You may have these connections at the back of your computer. They look like round sockets, about 1 cm in diameter, and each port contains 6 holes in a circle. You would usually have two PS/2 ports, one for the mouse and one for the keyboard.

31.9 Infra-red ports

One problem associated with all of the peripherals that computers users use now is the clutter of the cables. Users also want more and more to be totally portable. One solution that has become more commonplace recently is the use of infra-red ports. By using these ports, you get away from the need for cables. However, there are some problems associated with this technology. Both the sending device and receiving device must be pointing at each other, in 'line-of-sight'. Not only that but the distance the devices can be apart is limited to about three feet for successful transfer of data. For these reasons, infra-red is often limited to providing a wireless connection between computers and printers.

31.10 Bluetooth

This technology is another wireless technology that is taking off! It allows users to connect one device to another device without physically connecting them together with cable. It does not require line-of-site between the transmitting and receiving devices and the range is about 10 metres, although this can be extended to 100 metres with special equipment. For these reasons, this technology has some advantages over infra-red connections. It can be used, for example, to pick-up email automatically into a personal organiser from a computer as soon as you get within range of the computer. This will happen automatically because a Bluetooth-enabled device is constantly scanning for other Bluetooth devices. When it finds one, they both start communicating automatically! The technology can be used to send a document for printing to a printer on the other side of the room, can be used to connect up a scanner, mouse or FAX machine. In fact, the possibilities are endless!

31.11 Firewire

This is a very fast standard! It is used to connect peripheral devices to a computer, especially when a lot of data has to be transferred, for example, when using camcorders for video editing. It is also useful because the connection cable is much thinner than other types of communication cable and this standard provides for using a simple common plug and socket arrangement on peripherals and computers. You can also connect up devices without disrupting the computer as well as connecting up more than one device to only one socket.

31.12 Data transmission definitions

When data is moved from one place to another, we talk about 'data transmission'. You can classify data transmission into different categories known as simplex, half-duplex and duplex. These terms are described below.

31.12.1 Simplex data transmission

When data transmission can only take place in one direction, we talk about 'simplex transmission'. Teletext is a classic example of simplex transmission. Data is broadcast by television companies at the same time as TV pictures and picked up via aerials in people's homes. People do not send back signals from their television to the aerials. Of course, you can use a handset to send requests for pages to your TV's special Teletext adapter and the special Teletext adapter then captures the requested Teletext pages when they get broadcast. The requests don't get sent from the TV back to the transmitters.

31.12.2 Half duplex transmission

Simplex data transmission is one-way communication. 'Half duplex communication' is the term given to communication that can happen in both directions but not at the same time. The classic example of this is a set of walkie-talkies. Each handset can be used to either send or receive but cannot do both at the same time.

31.12.3 Duplex data transmission (sometimes called 'full duplex')

Duplex data transmission is the term used to describe any communication that takes place in both directions at the same time. The classic example of this is the telephone. You can both talk and hear simultaneously.

31.13 An introduction to data transmission errors

When data is transferred from one place to another, it can become 'corrupted'. For example, electrical cables and electrical devices generate a magnetic field. This field can interfere with the electrical signals that make up transmitted data, resulting in bits being changed from a 'one' to a 'zero' or from a 'zero' to a 'one'. It is necessary for computers to check that data has been sent correctly, that it hasn't become corrupted. There are four ways that we will discuss. These are parity checking, echo, check sum and self-correcting parity blocks.

31.13.1 Parity

When you send a byte of information using 7-bit ASCII, you have one bit spare. This can be used to check for errors in transmission. We have already said that when bytes are moved from one place to another, particularly over long distances, the bits that make up a byte can get corrupted by electrical interference. By using an error-checking method known as 'parity checking', ***half*** of these types of errors can be detected. Parity checking involves both devices deciding in advance whether they are going to use even parity or odd parity. There is no advantage of one method over the other. With even parity, the number of bits in every byte must always be even. With odd parity, the number of bits in every byte must be odd.

31.13.2 An example of parity

For example, you are sending the 7-bit ASCII code 0001001. The byte we need to send is ?0001001. The question mark is the parity bit, which we have to decide whether to make a zero or a one. When we look at the ASCII code, we can see only two bits are set (set means a 'one'). If it was decided before transmission to use even parity, we know that the number of bits in every byte must therefore be even. We must make the parity bit in this byte a zero to keep the total number of bits even. If we had decided in advance to use odd parity for data transmissions, then in the above example, we would have had to set the parity bit to a one, to make the total number of bits in the byte odd.

31.13.3 Even parity

Let us assume that we are using even parity, and the byte 00001001 was sent. The computer that receives the byte knows that it is using even parity because it was agreed in advance. It counts the number of bits in the byte and if they are even it accepts the data. If it counts an odd number of bits, however, then one of the bits must have been corrupted and it must signal to the sending computer to send the data again.

31.13.4 Odd parity

Let us assume that we were using odd parity, and the byte 10001001 was sent. The computer that receives the byte knows that it is using odd parity because it was agreed in advance. It counts the number of bits in the byte and if they are odd it accepts the data. If it counts an even number of bits, however, then one of the bits must have been corrupted and it must signal to the sending computer to send the data again.

If you send 00001001 using even parity and two of the bits get corrupted, for example, to 11001001, then there will still be an even number of bits and the data will be accepted, despite their being an error. If 4 bits or 6 bits or 8 bits were corrupted in this example using even parity, there would also be no error reported. In fact, an error will only be reported if 1, 3, 5 or 7 bits get corrupted. The same argument applies to picking up errors using odd parity. Parity checking is a useful way of picking up ***half*** of the errors created during data transmission.

31.13.5 Echo

Another very useful way of checking if a message has been sent successfully is to use a technique known as 'echo'. The way this works is that a message is sent. The receiving computer then returns the message to the sending computer that asks, "Is this what you sent"? If it is, then the sending computer signals to the receiving computer that the message was sent correctly. If it isn't, then the message is re-sent. Echo requires data to be sent twice and therefore takes longer.

31.13.6 Check Sum

Another method used to check data is the check sum method. Data is usually sent in blocks because that is the most efficient way of sending the data. Apart from the data, however, an extra byte is added, known as the 'Check Sum'. The value of the check sum is arrived at by adding up the data bytes and discarding any carry. When the receiving computer receives the data, it does its own check sum calculation, and then compares it with the check sum that was sent. If they're the same, then it accepts the data. If they are not the same, then it can request the data to be sent again.

31.13.7 Self-correcting parity blocks

If we consider sending data in blocks, we can correct some errors automatically, without asking the sending computer to resend the data. If we send 8 bytes of data and consider them in a block, we can see how this works. We will use odd parity in our self-correcting parity block.

0	1	1	0	1	1	1	0
0	0	0	0	0	0	0	1
1	1	1	1	1	1	1	0
0	1	1	0	0	0	0	1
1	1	1	1	0	0	1	0
1	1	1	1	1	0	1	1
0	0	1	0	0	0	1	1
0	0	1	0	0	1	0	0

Each bit in this row is the parity bit for each vertical byte.

Each bit in this column is the parity bit for each horizontal byte.

We transmit the bits as we normally would, but we look at them in blocks, to see if there are any errors. Consider what would happen if the bit shaded, switches to a 1, perhaps because of electrical interference.

0	1	1	0	1	1	1	0
0	0	0	0	0	0	0	1
1	1	1	1	1	1	1	0
0	1	1	1	0	0	0	1
1	1	1	1	0	0	1	0
1	1	1	1	1	0	1	1
0	0	1	0	0	0	1	1
0	0	1	0	0	1	0	0

The parity bit on this horizontal byte is now wrong. Remember, we are using odd parity.

The parity bit on this vertical byte is now wrong. Remember, we are using odd parity.

Where the horizontal byte and the vertical byte cross must be where the error is. The computer which received this block of data now knows which bit is incorrect and can automatically swap it back. It doesn't need to ask for the data to be sent again!

31.14 Packet and circuit switching

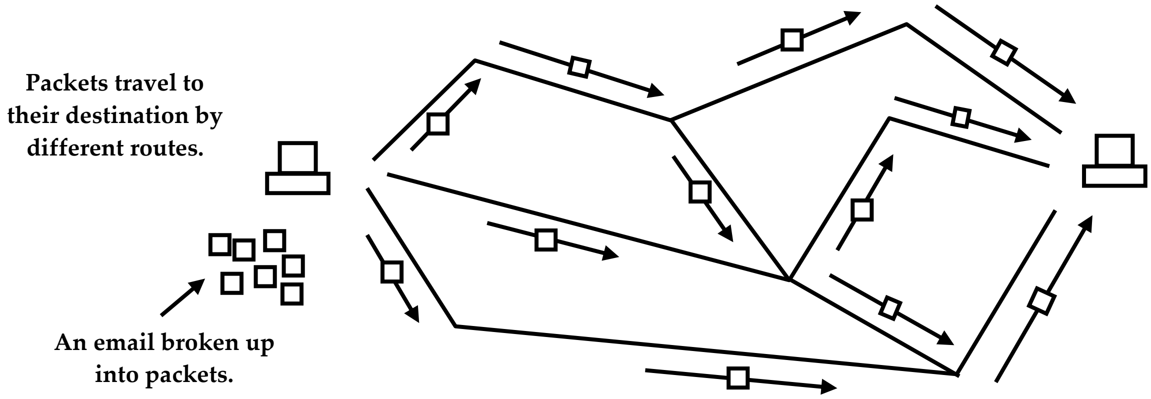
When a message is being sent, it can be sent across a network using either **packet switching** or **circuit switching**.

31.14.1 Packet switching

When an email message is sent across a network such as the Internet using packet switching, the email is split up into 'packets' of data, each of equal size. To each packet is attached some extra information. This includes:

- The address of the sender of the packet.
- The destination of the packet.
- Email reassembling information.
- A check sum for error detection.

When the message is sent, the packets each try to find the best path to the destination. They split up and go by different routes. The packets are passed on from one network to the next, ever forward to the final destination. This means that the network is being used efficiently because no single path is being tied up with just one message. In addition, you are not relying heavily on one particular communication route. With packet switching there may be a delay while each of the packets reaches their destination address. This can happen if, for example, a computer receives lots of packets at once. It will have to queue them all up and then deal with them in turn. When the all of the packets have arrived at their final destination, they need to be reassembled because they will not have arrived in the correct order. One other advantage of packet switching is security. It is almost impossible for someone to intercept all of the packets in a message. If someone does grab a packet, they won't be able to read it because they don't have the other packets. Nevertheless, it is always wise to encrypt data before it is sent, so all the packets are scrambled and unreadable by hackers.

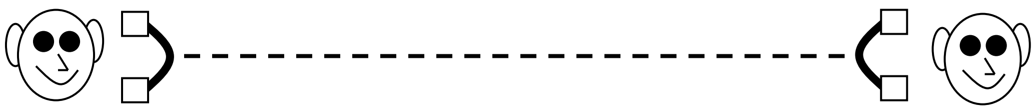


31.14.2 Circuit switching

This is a method of communication that sets up a communications channel just before the packets in a message are sent and then keeps that channel open exclusively for the entire duration of that transmission, until all the packets have been sent. All the packets arrive at their destination in the correct order, because they have been sent down only one channel in the correct order. It might be relatively easy to intercept a transmission because if you have access to the communications line then you can grab all the packets.

It used to be the case that data was sent across networks using packet switching and circuit switching was used for phone calls. However, these two technologies have merged in recent years. This is because phone lines extend everywhere across the globe whilst data networks have become very cheap and efficient at transferring data.

The phone call is the classic example of circuit switching and probably the easiest way to imagine this technology. You dial up a friend. If they are not engaged and answer your call, you then have a direct connection to them until you hang up. Someone can listen in on your call by simply tapping into your connection. Another issue with circuit switching, certainly in older parts of the telephone network (known as the 'PSTN') is that it involves setting up transmission channels by opening and closing electro-mechanical switches. This act causes electrical interference, which can corrupt data (causing interference) if adequate precautions are not taken.



A channel is kept permanently open for the entire communication.

31.15 Protocols

Devices need to communicate. For example, a printer needs to send messages to a computer telling it that it has run out of paper or that it is ready to print while a computer needs to send the data it wants to print to the printer. Computers need to send data between themselves so that, for example, email can be exchanged and the Internet can function. When two devices want to successfully communicate, they must agree to follow some rules about the way they will do it. These are known as 'protocols'.

A **communications protocol** is the phrase used to describe a set of rules that communication equipment adhere to when they send data to each other. If two devices are sending and receiving data but using different rules then the receiving device will not understand what was sent! Because communications protocols are so important, a special organisation known as the ITU, or International Telecommunications Union, exists to lay down the exact detail in the standards that everyone will use. They define the protocols. The two basic protocols used on the Internet are IP (Internet Protocol) and TCP (Transmission Control Protocol). Together, they are referred to as the TCP/IP. TCP/IP controls how information is successfully transferred between computers on the Internet. When web pages are requested and then sent over the Internet, another protocol used in addition to TCP/IP is the http protocol, or Hyper Text Transfer Protocol. When files are transferred over the Internet, the FTP protocol, or File Transfer Protocol, is used. Sending email can be done using the SMTP, or Simple Mail Transfer Protocol.

31.15.1 What rules are agreed to in a communications protocol?

A communications protocol needs to specify a range of things before successful communication can take place. These include:

- What baud rate will be used.
- What parity checking will be used.
- Whether software or hardware 'handshaking' is to be used.
- What character set is to be used.
- How many bits will be used for data.
- How many control bits will be used to control data transfer.

31.15.2 Physical and logical layers

When we talk about protocols, we can divide the rules that make up a protocol into two parts:

- logically based parts
- physically based parts.

The logically based parts apply to the data. These might include:

- baud rate to be used
- what error correction techniques will be used
- how messages will be synchronised
- descriptions of the rules governing the data
- The size of the packet
- compression techniques to be used
- how data will be encrypted
- digital signatures and certificates.

The physically based parts apply to the methods of communication. These might include:

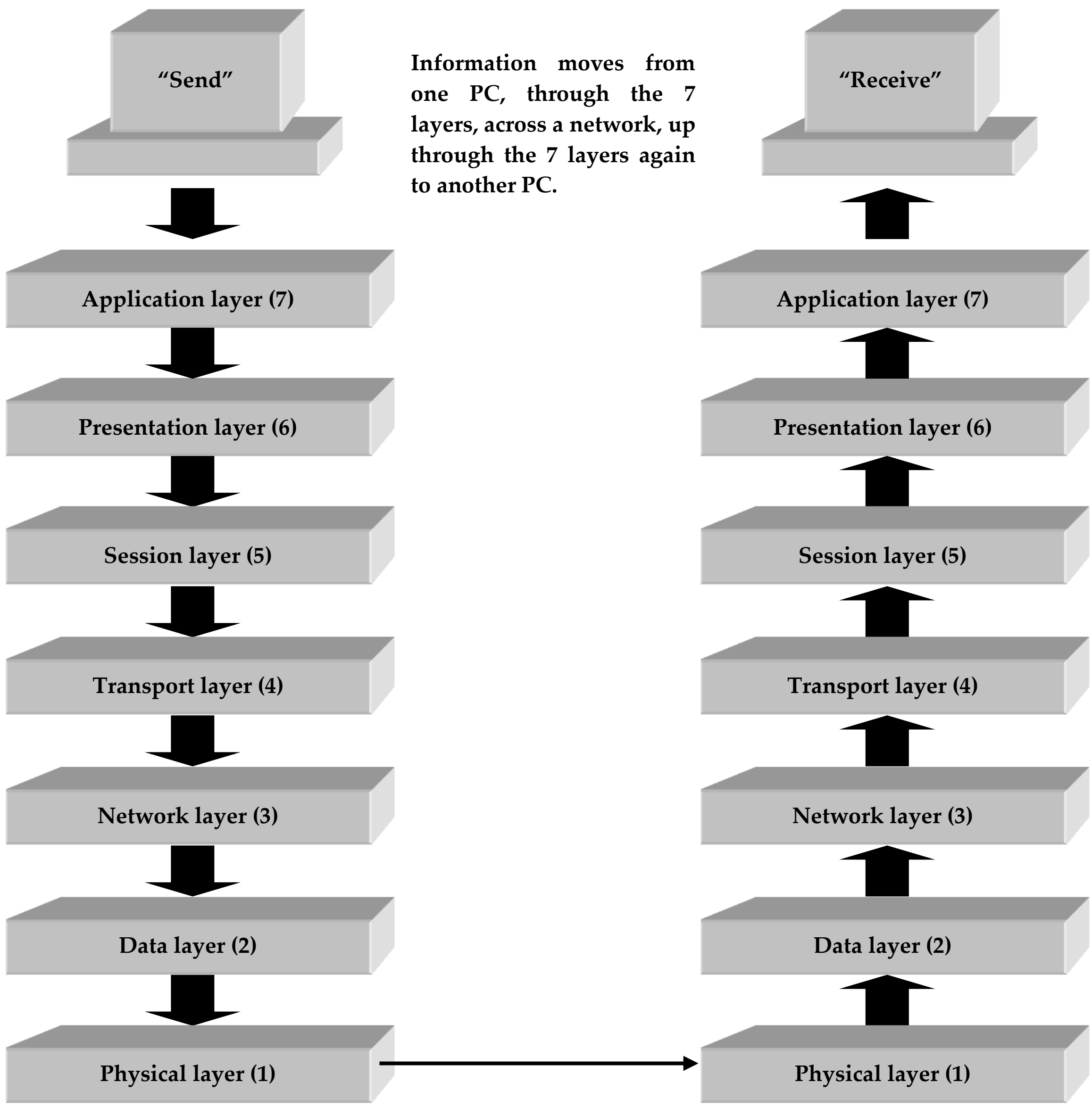
- Whether wireless communications will be used or hard-wiring
- If hard-wiring, whether copper cables or fibre optics be used
- Whether serial or parallel communications will be used.

31.16 An introduction to the OSI model

There are many different parts to a protocol, and many different protocols, both logical and physical, that all need to work together. Because of the complexity of the different parts of the protocol, they need to be organised and this is what the OSI model is all about.

The OSI (Open Systems Interconnection) model is a model that describes what happens to some information when it is being sent from one computer across a network to another. The model organises the protocols and standards that are used into independent 'layers' of protocols and standards. As long as manufacturers of hardware and software products make products that conform to the standards and protocols in any one particular layer, then their product will slot into the model, will interface correctly with the layer above and below it and therefore will successfully enable communication to take place. For example, imagine you have written an email. What happens when you hit "Send"?

- The message is passed to the Application layer, layer 7.
- It then travels through each of the remaining independent 'layers' of protocols.
- At each layer, something more happens to the message, something extra, that prepares it for the next layer.
- Finally, the message is ready to leave the computer and be put on the network interconnections as a stream of packets of data. The packets travel to their destination, where they enter the Physical layer of protocols.
- The packets move up the independent layers in reverse order, until the message has been reassembled.



31.16.1 The importance of standards in communication between computers

Setting standards, rules that all manufacturers of hardware and software will follow, are important for a number of reasons:

- Standards describe accurately and unambiguously how information is transmitted.
- A manufacturer's products will work successfully with other manufacturer's products if they all follow the same standards.
- By defining a set of standards, you are providing a framework within which all manufacturers can design new, successful products.
- Standards break down complex ideas into smaller, methodical, easier-to-understand components.

31.16.2 The benefits of the OSI model

There are a number of key benefits to using the OSI model.

- The OSI model provides standardisation with hardware and software involved with communication.
- Manufacturers of hardware and software can be confident that their products can be used in successful communication if they follow the standards laid down in the OSI model. They may design products that use the standards and protocols in one layer. A new product, therefore, will interface correctly with the layers above and below it in the model, so long as they follow the protocols and standards in that particular layer. They can be confident that their products can be used successfully and so they will have happy customers!
- If there is a communications problem, the model helps someone track down where the error is, in which layer and what caused it. This is because the layers are all independent of each other.
- Protocols are updated from time to time. They only need to be updated in the layers where they occur.

31.16.3 The principle of layering

The term 'layering' is often discussed when the OSI model is being studied. It simply refers to taking all of the protocols and standards and grouping them into one of seven layers. This has a number of benefits:

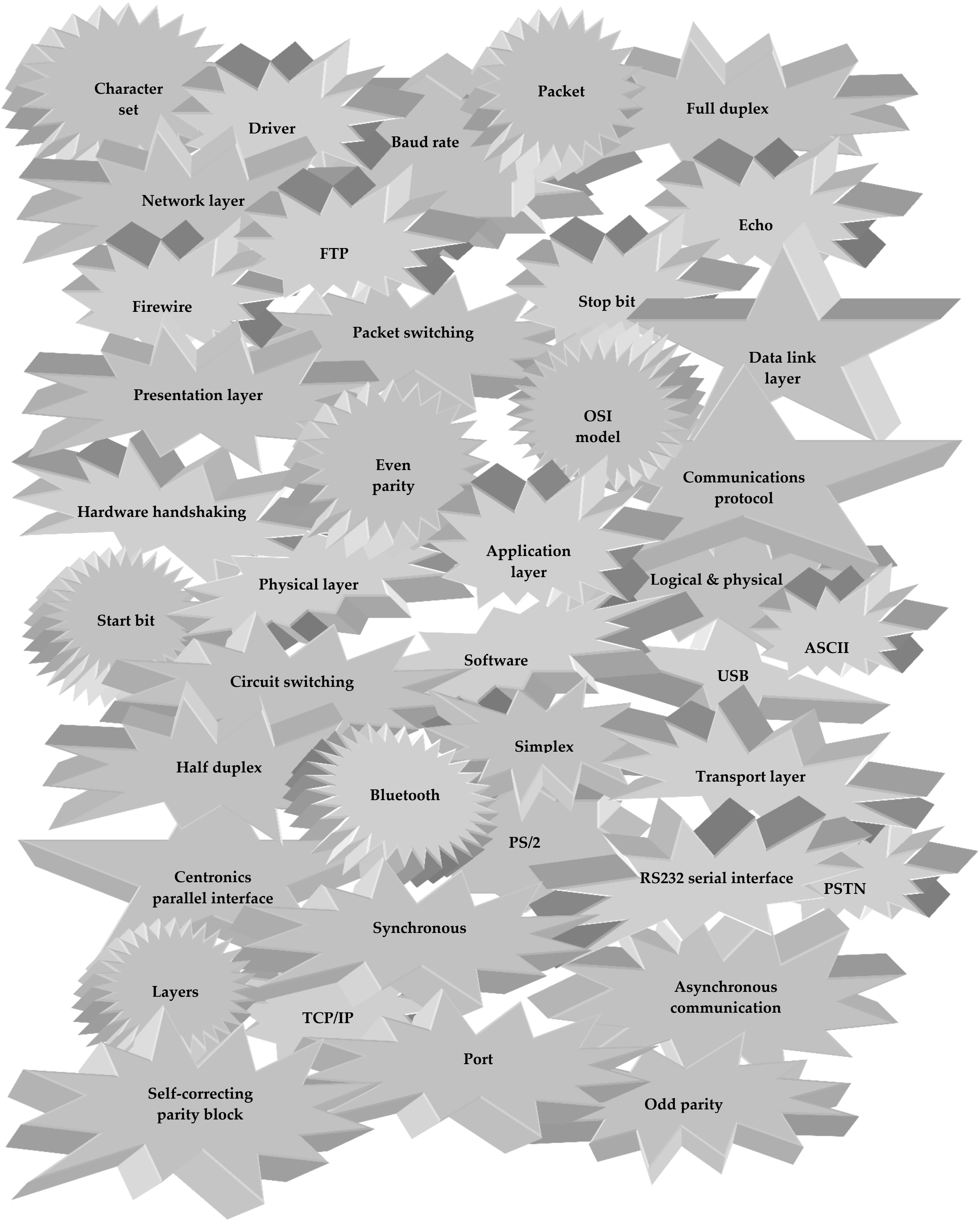
- It means that a product that uses a particular layer can be designed and tested totally separately from the other layers.
- Layering describes the process where each layer of protocols builds upon the output from the previous layer, and provides the input for the next layer.
- As long as the protocols and standards for that layer are followed, a manufacturer can be confident that it will work with other manufacturer's equipment.
- Each layer can be treated independently.
- Layering helps problems to be identified because problems occur inside an independent layer.
- It means that as long as everyone uses the OSI model, successful communication can take place. Before the model was specified, data communication between different applications and hardware was practically impossible.

31.16.4 More on the layers

As we know now, each layer is independent of the other layers. One layer builds upon what it receives from the previous layer and passes it on to the next one. Here is a table that describes briefly what each layer is responsible for.

Layer 7	Application layer	Any	**This layer provides the interface for the user between applications and the network. It accepts information from applications and passes them on to the next layer.**
Layer 6	Presentation layer	Person	**This layer takes data provided by the application layer and converts it into a format the next layer (and subsequent layers) can understand and make use of.**
Layer 5	Session layer	Seeing	**At some point, you have to make a connection with a piece of equipment that will receive the data packets. This layer's job is to set-up, maintain and end a communication session.**
Layer 4	Transport layer	Treble	**This layer is responsible for the ordering and reassembly of packets of data. It may have to take packets from multiple applications and order them into a stream of packets.**
Layer 3	Network layer	Needs	**This layer is concerned with the delivery of data packets. It is in charge of working out addresses and how packets will get to a particular location (known as 'routing').**
Layer 2	Data link layer	Double	**This layer is concerned with getting error-free connections across networks. It identifies errors and corrects them and coordinates the transfer of streams of data.**
Layer 1	Physical layer	Prescriptions	**Concerned with the physical characteristics of the hardware such as voltages, pin assignments in plugs and so on.**

Look at the following diagram. Explain clearly what each term means.



Chapter 32 - Systems Development Life Cycle

32.1 Some words of advice....

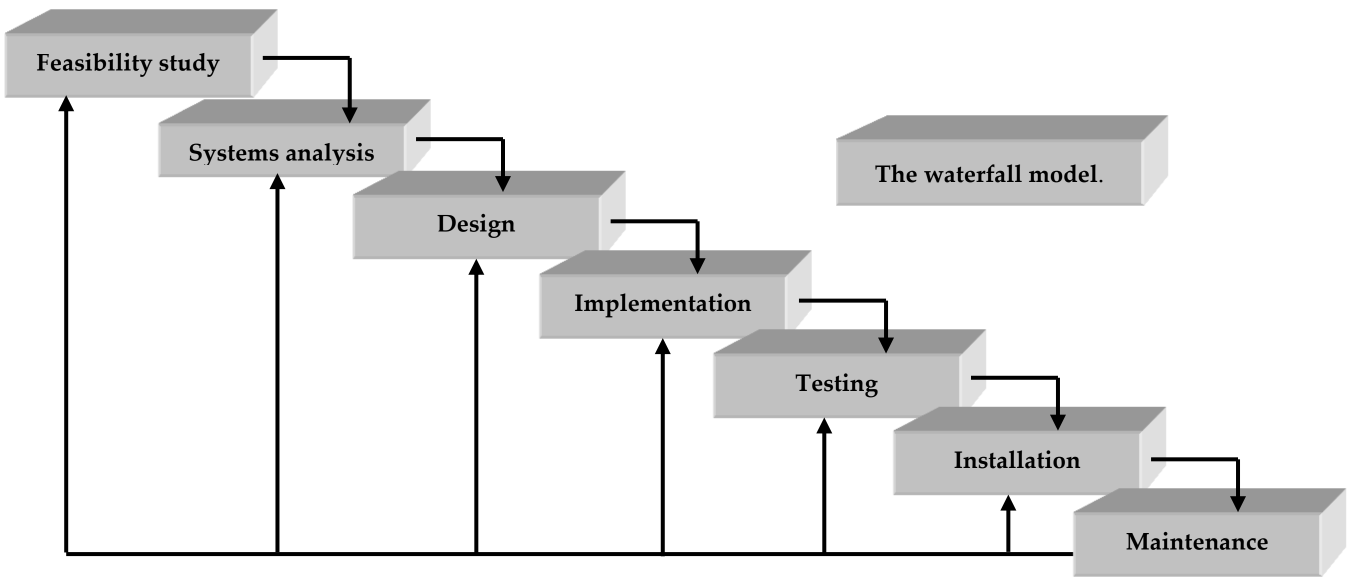
The systems life cycle (sometimes called the 'project life cycle') can be confusing! This is because every author has his or her own idea of exactly what it means, what jobs should be done in each stage and what documents should be produced. On top of that, people will combine or split certain stages in the life cycle and then use different words to describe the same thing, or the same words to describe different things!! There is no getting away from this problem but you do need to tackle it head on. You must read around the subject. Read four or five different descriptions of the systems life cycle and then make your own mind up. Remember as well that if you are really not sure which description of the systems life cycle to use, then use the descriptions and definitions found in the British Computer Society's 'A Glossary of Computing Terms'. It is the benchmark against which other ideas about what terms mean should be judged. It is also used as the reference for answers by Exam Boards!

32.2 Introduction - what is a 'systems life cycle'?

The phrase 'systems life cycle' simply describes the steps that are taken in a project, from the time that the project is started to when it is finished. When any computer-related project is initiated, a number of distinct steps, or stages, can be identified in the life of the project. Each of these stages will involve people doing jobs and producing 'things', for example, a design document, a test plan or a piece of program code. Each of these things takes the project a little further towards completion. Things that have to be produced for each stage are known as 'deliverables', for example, a report.

The idea behind the systems life cycle is that the deliverables associated with each stage in the project must be produced and checked off by the Project Manager before the next stage can begin. A stage cannot be started until the previous stage is finished. This stops a project getting ahead of itself. For example, it will stop someone trying to start the stage called 'implementation' (the stage where you actually make the project using a database application or code) before all of the design documentation has been completed. You may have had some experience of this scenario yourselves with coursework - you don't want to do the paperwork or a detailed design, you just want to get on and do the project! This, however, is the road to potential disaster! For example, how can a project be designed if it is not clear what the problem is? How can a project be built if it is not designed? How can it be installed if it is not properly tested? What happens if a key project member leaves - how can someone new pick up where they left off if half of the paperwork is missing or incomplete? How can a Project Manager accurately manage a project if they can't clearly see that deliverables are being completed on time and within the budget? How can someone make changes to the product in the future if the documentation is incomplete?

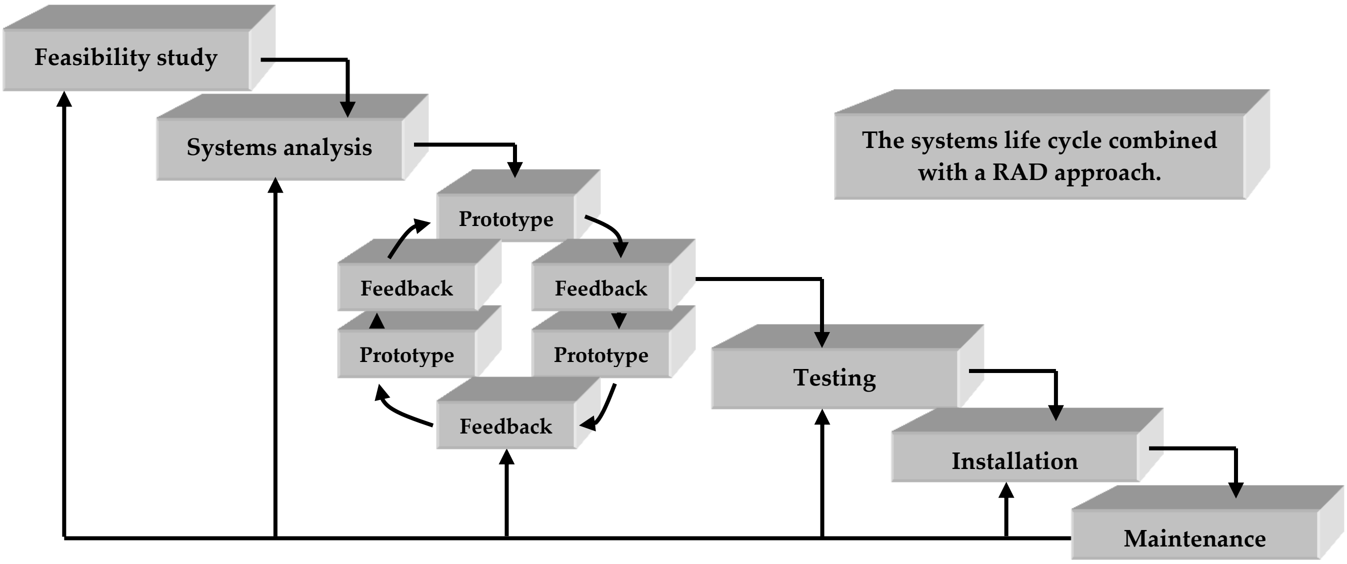
The list of potential problems goes on and on. A systems life cycle gives a project a structure and therefore allows a Project Manager to manage the project rather than reacting to things when they go wrong! We can summarise the systems life cycle with the following diagram (called the 'waterfall model'):



Notice with this model that there is nothing stopping a stage feeding back into an earlier stage, so that the project can constantly loop around until it is perfect!

32.3 An alternative approach - Rapid Application Development (RAD) - the Spiral Model

We will briefly mention here the RAD approach to project development as an alternative to the traditional systems life cycle approach because it has become very important recently with the rise of something called 'Object Oriented Programming'.
The traditional waterfall approach has some drawbacks. For example, it can take a while for customers to actually see the final product. Another concern is that software produced using this approach in theory is easy to change but in practice is difficult.
The RAD approach is different to the classic systems life cycle. It involves designing and building a series of **prototypes**. After each one is built, the user is involved. They are asked to try out and comment on features and test some of the functions. Their comments are then fed back into the next design and prototype and a better one is produced. This process is repeated until the product is finished. After each round, the product spirals closer to the desired product. This process is sometimes called the **spiral model** as a result. The product is in effect developed a little bit at a time but constantly involves the customer throughout the development process. It is very common to combine the classic waterfall model with a prototyping approach; the waterfall model is followed, but the design and implementation stages involve the user commenting on a series of prototypes.



32.4 How are projects initiated?

Projects are initiated in a number of ways.

- A system that is currently in place in an organisation is not doing its job efficiently. Perhaps the business has grown and the system can't cope anymore. Perhaps too many problems keep arising.
- Perhaps it is a new business that needs a way of solving a problem, e.g. it needs to take on-line payments.
- Perhaps an old system isn't flexible enough, or costs too much to run.
- Perhaps there is a change in legislation that requires an existing computer system to be modified in some way.
- Perhaps the company has heard of some new technology that they think may help them make their business more efficient and give them an advantage over their competitors.

Whatever the reason, a manager will notice that a system needs to be looked at. In the first instance, they will arrange a meeting with a Systems Analyst.

32.5 The first meeting

A Systems Analyst's job is to take charge of assessing a problem and to see whether or not it is suitable for computerisation, or whether an existing computer system can be upgraded. They will be expert in assessing situations. They will have a wide knowledge of existing computer technology and will be able to visualise what can be achieved with this technology.

Before a Systems Analyst solves a problem, however, they need to be clear what the problem is! This might sound like common sense but can take a lot of time to get clear and it must be clear before any real work begins. When company X has a problem, they will call in a Systems Analyst to look into it. We can assume that the company will be expert in whatever area of business they are in. They will not be expert in computer systems. They will not know about the latest technology. They will not necessarily know all that can be achieved using computers in their business area. This is where the potential problem lies. The Systems Analyst is an expert in computer systems, but not company X's business. Company X is an expert in their business but may at best, have only a vague idea about computer technology and what it can do for the business. Yet they have to talk together at the beginning of a project to decide what the project aims are. It is a little like two people who speak two different languages meeting together. They have to find a common way to communicate.

When company X calls in the Systems Analyst for the first time, they will sit down together and company X will explain why they have contacted the Systems Analyst. The Systems Analyst will listen carefully and will also use this opportunity to ask any questions, perhaps clarifying some of the jargon that company X has used during the discussion or double-checking that they have understood what they have been told by the company. Together, they will try to define the 'scope' of the problem. They will try to flesh out what the problem is in language that both company X and the Systems Analyst understand.

32.6 The Feasibility study

After a company has made contact with a Systems Analyst, the Systems Analyst will go into the company and make a provisional study of the problem. They will do this to decide if a solution is possible, how it can be done and whether it is a good idea to implement. This study is known as a 'feasibility study'. The deliverable for this stage in the systems life cycle is the Feasibility Study Report. The Systems Analyst will carry out a number of tasks.

- They need to find out about the company and how it functions so that they can better understand and define both the problem and the context of the problem. Remember, the Systems Analyst doesn't know anything about the company yet, other than what they learnt in the first meeting. They need quickly to understand the business and its operations so that they can fully understand the nature of the problem.
- They need to investigate and then document the nature of the problem in some detail. This means that the Systems Analyst will do a scaled down 'systems analysis'. They might do some of the systems analysis tasks in 'outline detail'. You can read more about these actual tasks later.
- They need to identify, suggest and justify possible solutions. Often, a number of possible solutions are put forward to allow a choice to be made. The Systems Analyst will need to identify the benefits to the company of each possible solution put forward and how much each solution will cost the company. This is sometimes known as a 'cost-benefit analysis'.
- In addition to a cost-benefit analysis, the Systems Analyst will also identify a time scale for each proposed solution - how long each one would take to implement.
- They need to identify any legal, moral or ethical considerations in the possible solutions.
- They need to identify any implications for the company. This could include identifying any new skills that the workforce will need, identifying any changes in the way the company will do business, identifying any changes in the way customers will interact with the business and Health and Safety issues, for example.

The Systems Analyst will carry out the above tasks and produce a report called the Feasibility Report. They will then present this written report to the company and may also do a presentation of their findings. This report will be the document that the company uses to decide whether or not to proceed with the project.

Two outcomes are possible. They can decide not to go ahead with the project, perhaps because a solution is more expensive or more disruptive than they thought. Or they can decide to go ahead with a solution. If they decide to go ahead, they need first to select the solution that they want to go with from the range of possible solutions put forward in the Feasibility Report. They then need to agree other details such as the budget for the project and a start date for the next stage in the project. If you refer back to the systems life cycle diagram, you will see that the next stage is known as the Systems Analysis stage.

32.7 The Systems Analysis stage

A company may decide to go ahead with a project after considering the Feasibility Study. They will agree a date for the next stage in the project life cycle to begin, the 'Systems Analysis' stage. During this stage, the Systems Analyst will investigate the problem in much more detail than was done for the Feasibility Study. They must find out a number of things:

- Find out in detail how the current system works and identify problems.
- Identify what is required from the new system by all of the 'stakeholders'.
- Identify the data to be stored, the volume of data and the data about the data!
- Identify the current hardware and software and any necessary upgrades required.
- Identify all the data that will enter the system and all the data that will leave it.
- Produce data flow diagrams that describe the flow of data around the system.
- Produce written descriptions and diagrams that aid understanding.

Where does the Systems Analyst start? They need to use a range of information-gathering techniques to collect the information that allows them to do the above.

There are six fact-finding methods that could be used. These are:

1. One-to-one interviews
2. Group interviews
3. Collecting documents
4. Observations
5. Questionnaires
6. Letters, emails and the phone

Six fact-finding methods

32.7.1 One-to-one interviews

A well-planned interview will involve:

- preparing questions
- making an appointment
- giving the person you are interviewing the questions in advance
- carrying out the interview, ideally recording it if possible
- writing up the interview for the appendix of the Systems Analyst's report
- clarifying with the interviewee any points the interviewer is not sure about, then updating the write-up
- summarising the interview's findings for the body of the Systems Analyst's report
- thanking the interviewee.

Interviews should ideally be held with the managers first, to get an overview of a system, followed by interviews with the users of a system, to find out the day-to-day operational problems. During interviews, the Systems Analyst can start getting a feel for what people want from any new system, what problems they want solving, what reports they want generating and so on. Interviews can be quite time-consuming to organise, especially if you need to do more than a few. They are important to do, however, because details of the way things are currently done can quickly be ascertained.

32.7.2 Group interviews

We have already said that doing lots of one-to-one interviews can be time-consuming. A compromise would be to hold a group interview. This could be done with the employees from a department, for example. They could all be called together and operations, problems and methods could be discussed. The Systems Analyst needs to understand that for many workers, introducing a new computerised system equates to redundancies and they will need to show skill in finding out what they need to know without upsetting anyone! While group interviews can save time, there is a danger that one or two people will dominate these events and that some people's views will be drowned out.

32.7.3 Collecting documents

An excellent way of finding out about the data used in any existing system and the way a company currently does things is to collect examples of documents that are used. These will show the names of data items, give examples of the items themselves and will prompt the Systems Analyst to ask employees questions about the data. For example:

- Where did the data originate?
- How is the data constructed?
- Are there any synonyms used for any data items - different names for the same piece of data? For example, most people would call a car a 'car' but some might use the words 'auto or 'automobile'. You would,

however, only want to set up one field for car, not three! Are there any instances where different names are used to describe the *same* piece of data in the system being investigated?
- What is the range of allowable values?
- What actually happens to the data on a document?
- Who uses the data?
- When is the data removed from a system and who does it?

32.7.4 Observations
Sometimes, what people say they do is different to what actually happens in practice. Sometimes, people simply don't mention things in an interview because they either forget or it seems too trivial to mention. A Systems Analyst can sometimes find out information by spending time watching people go about their job. People can be very aware of the presence of someone that they know is watching what they are doing and behave differently, although with skill and time, this problem can be overcome.

32.7.5 Questionnaires
Questionnaires can be used to collect very focused answers to questions from a large number of people relatively quickly and without using up a lot of manpower (compared to interviews, for example). However,

- Writing good questions in a questionnaire is very difficult indeed.
- Most people do not return them.

An improvement, if resources are available, is to have someone 'interview' people using a questionnaire. The person interviewing asks the questions on the questionnaire and records the answers. If the person doesn't understand a question, it can be explained to them. More questionnaires will be completed this way.

32.7.6 Letters, email and phone
Phone calls can be useful because you can actually speak to the person you need to without having to spend time travelling to see them. Email is immediate compared to letters but whilst everyone can be sent a letter, not everyone has an email address. Even if someone has an email address, they may not use it and you may not realise they don't! Letters are more time-consuming to prepare and write compared to email, which tend to be written in a quick, matter-of-fact way. Both emails and letters give you answers in writing compared to phone calls, where you are relying on your ability to take good notes and understand what someone is telling you. Emails and the phone, and to a lesser degree letters, are especially useful to quickly clarify points made during an interview.

32.8 Making sense of the information gathered in the systems analysis stage
Sometimes, too much information can be as bad as too little information! A Systems Analyst will quickly amass a large amount of data and information, as they carry out interviews, do questionnaires, collect documents and so on. They need to make sense of it all for the Systems Analysis Report. They would typically do the following things as part of the systems analysis stage:

1) Produce written descriptions of current methods and systems.
2) Draw dataflow diagrams (DFDs).
3) Draw systems flowcharts.
4) Draw Jackson diagrams.
5) Design the data dictionary.
6) Write down descriptions of problems.
7) Write down and agree with the customer the Requirements Specification.
8) Write down possible outline solutions that would satisfy the Requirements Specification.
9) Write down a description of the current hardware and software and what needs to be upgraded or replaced.

32.8.1 Written descriptions of current methods and systems
A good starting point for a Systems Analyst when investigating a new problem in an unfamiliar business is to write out how each system in the problem area operates in as much detail as possible. This should be done:

- By describing systems using the present tense.
- By describing systems as though the Systems Analyst is 'a fly on the wall', an observer.

For example, consider a school library that is completely paper-based. A Systems Analyst has been asked to investigate computerising the library. They would arrange an interview with the librarian who would then explain how each of the sub-systems in the library work. One sub-system might be the one used to take books out. The Systems Analyst would listen to what the librarian says and then produce a written description of it. It might go something like this:

A pupil selects a book from the shelf and takes it to the checkout counter. They present the book and their membership card to the librarian. The librarian retrieves the member's details from the 'members card index system'. She then checks how many books the pupil has out already. If it exceeds 3 then the pupil is told that they must return a book before they can take any other out. The librarian also checks to see if the membership has been 'blocked'. This can occur if a fine hasn't been paid for an overdue book or if the pupil has been misbehaving. If the pupil has got less than three books and the account has not been blocked, then the librarian transfers the title and author details to the member's card and the date the book is due back. This card is then filed. The librarian retrieves the 'book details' card from the 'borrowed book card index system' and records the pupil's membership number on it and the return date. This is filed. The book is stamped with the return date and returned to the pupil with their membership card.

Of course, the Systems Analyst will want to go back to the librarian and read back the description after it has been written up, just to check that they have understood everything correctly and that nothing has been missed out. The above example is for the 'Taking a book out' sub-system. The Systems Analyst would also need to write descriptions for each sub-system:

- The 'returning a book' sub-system.
- The 'blocking a pupil' sub-system.
- The 'fine' sub-system.
- The 'adding a new book to the library' sub-system.
- The 'removing a book from the library' sub-system.
- The 'adding a new member' sub-system.
- The 'removing a member' sub-system.
- The 'reserving a book' sub-system.

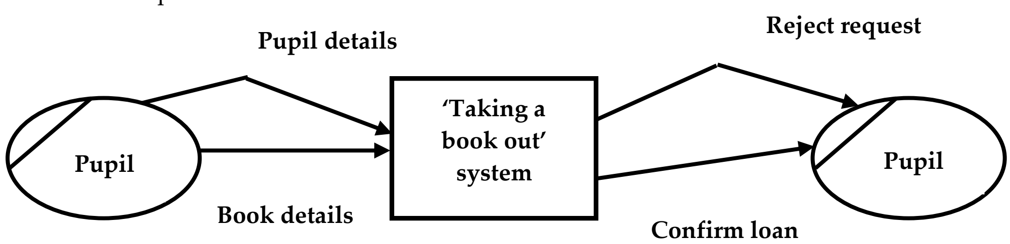
Describing these sub-systems, arranging interviews, writing them up, summarising them and then checking that they are accurate and complete descriptions is not a 5 minute job! A lot of information is going to be generated and, following on from written descriptions, the Systems Analyst will want to use special diagrams to bring the information together.

32.8.2 Dataflow Diagrams (DFD)

Dataflow diagrams, or DFD diagrams, are used to summarise the flow of data around a system. Written descriptions are fine, but pulling a lot of information together in the form of a diagram helps everyone involved 'see' what is going on in the system a lot easier. Because systems such as the library can get quite complicated, they should be broken down into sub-systems, as we have already explained. Each one can then be described and then a DFD can be drawn for each one. There are only four basic symbols in a DFD (not counting the big system box in the Level zero diagram).

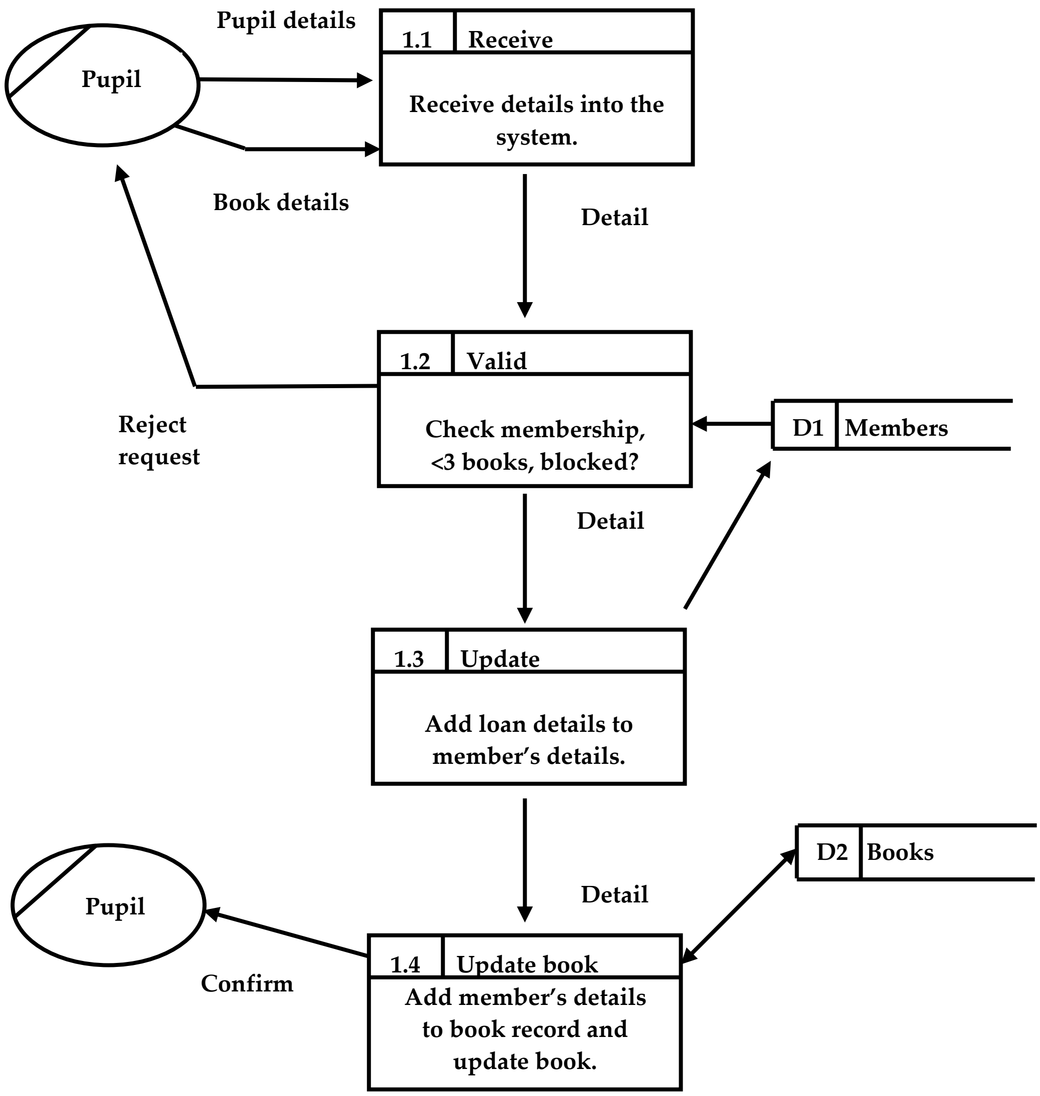
1) An oval, representing an entity. This is someone or something that puts information into the system or receives information from it.
2) An arrow, representing a flow of data from one place to another.
3) A box, which represents an action or processing on some data.
4) A long box, which represents a store of data.

In the next example, we can see how DFDs are produced. The diagram below shows the first step in drawing a DFD. It shows the whole system as a box. It also shows the entities such as people that interact with the system as well as any data that enters the system from an entity and any data that leaves the system to an entity. This DFD is known as a 'level zero' or 'context diagram'. Notice that the librarian is not shown because she is part of the system, not external to it. Also notice that there are two ovals representing 'pupil'. They are the same entity - we could have drawn just one. This would have cluttered the diagram, however, and would have made it less clear. Two ovals have been drawn instead. To show that they are the same entity, a diagonal line is drawn in the top left of each one.



A context diagram.

The context diagram above shows a simplified version of the whole system but doesn't show what actually happens to the data, where it is stored and the processes that happen to it. It doesn't show what is happening inside the box called 'Taking a book out' sub-system. The box needs to be broken down to show some detail. The next stage is to produce a 'level one diagram'. A first attempt is shown in the next diagram.



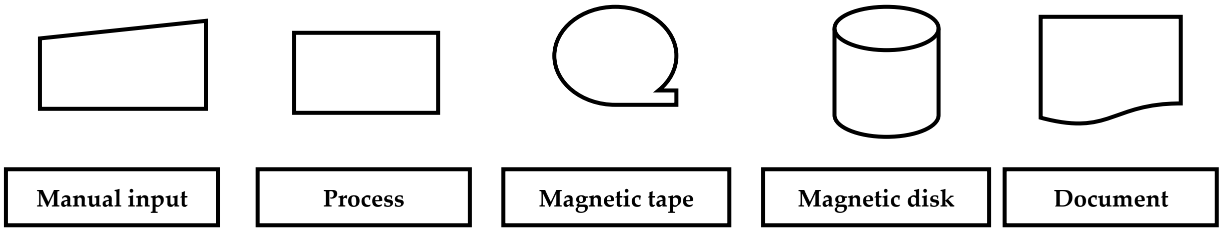
A level one diagram.

See if you can read the description of the 'Taking a book out' sub-system and match it up to the level one diagram. Try to follow what is happening. Do note that a Systems Analyst will take quite a few attempts to get the DFDs right. When you do projects, don't expect to get these diagrams correct at the first attempt. You need to have a go at it, then discuss it with someone, then sleep on it, then have another go, and repeat the process until you get it right! The level one diagram is a big improvement on the level zero diagram because it shows the processes that happen to data and what storage there is. Some of the processes, such as '1.2 Valid' could be broken down further into what are known as **level two diagrams**. These are beyond the scope of this discussion. Interestingly, this diagram can be closely matched to a database. For example, where there is a data store, you need to have a table in a database. Where there is a process, you need to have a query. Where there is a flow into the system, you need to have a data input form and where there is a flow out of the system, you need to have a report. We are starting to move closer to designing a database!

32.8.3 Systems flowcharts

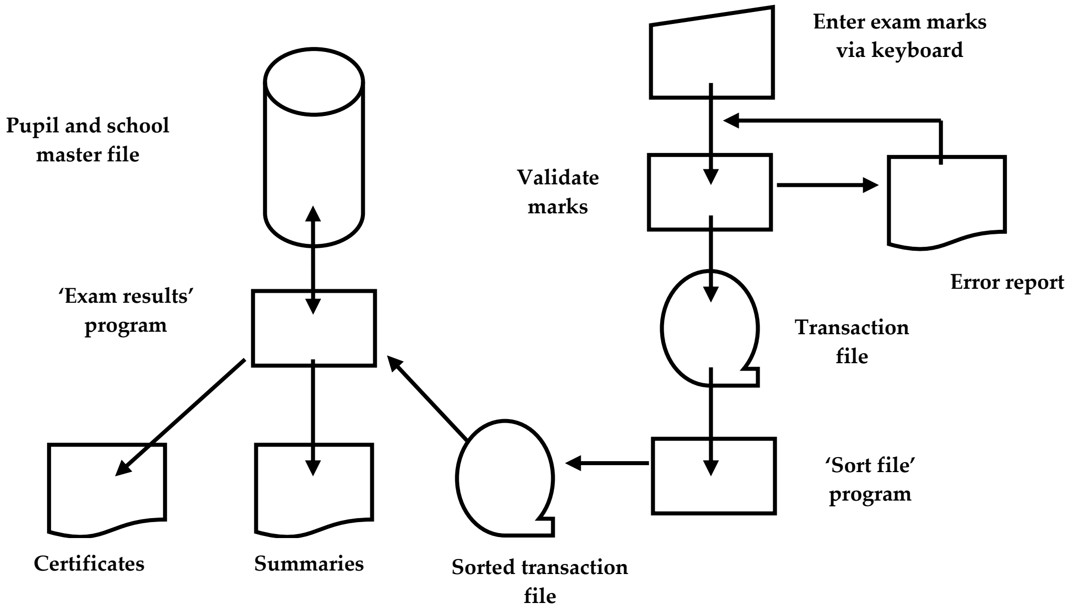
A similar but different type of Dataflow Diagram is known as a 'systems flowchart'. This type of diagram also gives an overall picture of a system. A systems flowchart shows similar information to DFDs in that it shows the processes that happen on data along with the data flows. In addition, however, these diagrams also show what **hardware** is used for input, what hardware is used for output and also the data storage devices. They also show what ***type* of file** is being used, for example, whether it is a master file or a transaction file. Both system flowcharts and DFDs are used by analysts to show whole systems.

A number of symbols are used in systems flowcharts. These include (amongst others):



Some of the symbols used in systems flowcharts.

An exam board marks a set of exam papers for pupils at a school. They enter the marks into their system, validating each mark as it is entered. An error report is generated for any problem marks and the marks are then re-entered. Once all the marks have been entered, they are sorted. Pupils' information is then retrieved from its system, the new marks entered and the records updated. Summaries of the results and certificates are then sent out. A system flowchart might look like this:

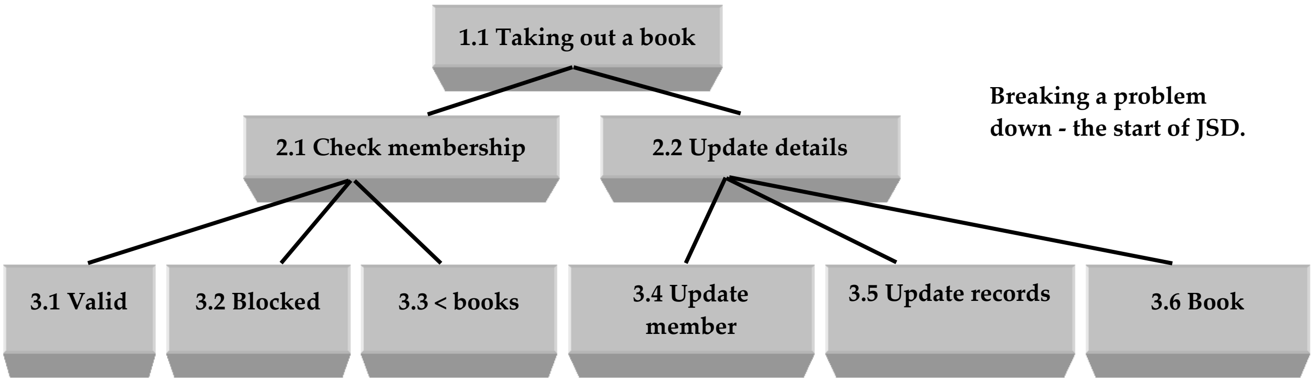


An example of a system flowchart.

Note that processes are shown (' Validate marks', 'Sort file' and 'Exam results' programs), data stores can be identified and flows of data are also on view. In addition, however, we can see some of the hardware needed (keyboard and hard disk) and some of the different types of files (master and transaction files).

32.8.4 Structure diagrams

Structure diagrams can also be used to represent systems and also program designs. They show a different type of information to DFDs. One type of structure diagram uses symbols from the JSD, or Jackson Structured Development way of showing systems. This breaks down a problem into ever-increasing levels of complexity. It is read from top to bottom (using a different number for each level) and from left to right (using a different number for each process in one level). Going back to the 'Taking a book out' sub-system, you could represent the sub-system using the following diagram:



Breaking a problem down - the start of JSD.

This type of diagram can be used to break down complex problems into easier-to-understand blocks. If you were programming, each block could then represent a module or function. We have come across this type of diagram before when we discussed top-down design. Even if you are not programming a system but just want to understand the processes involved, this kind of diagram can pull together a lot of complex descriptions into a form that can be understood at a glance.
We have introduced the idea of structure diagrams above, but we have only just scratched the surface. We will meet these diagrams again in another chapter and take them even further.

32.8.5 Data dictionary

The data dictionary holds information about data! Any system needs data to make the system work. The Systems Analyst must construct a dictionary of all the data items used in the system because this information will be needed by the people who actually build the new system, who write the software. This point of reference for information about data items is known as the 'data dictionary'. They tell somebody the form of the data, how each data item is actually made up. Data dictionaries are often best done as a table, using the following headings:

- The name of the data item.
- What synonyms there are for the data item.
- Whether it is a primary key or foreign key.
- Data type. (Whether it is a real number, an integer, a text, a character, a Boolean, a date and so on).
- Validation rules that apply. (E.g., the range of allowable values for integers, the number of allowable characters for text, the allowable characters for a character, the way that the date has to be entered using an input mask, the number of decimal points allowed, whether it is required or not and so on.
- Examples of typical data entries.
- The origin of the data, where it comes from, how it is generated in the first place, where it is stored.
- What exactly the data item is used for, what happens to it, why it is part of the system at all.
- Specification of access rights – who can view, edit or delete the data item.

The Systems Analyst will start the data dictionary at the beginning of the project and, like the list of problems, will add to it as new information becomes available. Some of this information may come from interviews, but much may come from existing documentation. One reason why collecting documents from an existing system is important is that it shows the Systems Analyst what data is needed in the current system, with examples of the data, where data comes from, how it is used and so on.

32.8.6 Written descriptions of problems

As interviews progress and the Analyst becomes more familiar with the existing system, problems with it will emerge. People using the current system and methods will highlight problems and the Analyst will notice others. For example,

- System users might complain about things not working as it should.
- Users might complain about some tasks being time-consuming or requiring a lot of paperwork.
- Customers might complain that they never get the right information at the right time.
- It might become obvious to the Analyst that some tasks are labour-intensive and could be easily automated.
- It might become obvious to the Analyst that some tasks which should be possible and which an organisation has never thought about doing are simply not possible with the current set-up.

For example, in the paper-based library example, the librarian might during interviews bemoan the fact that she has to use two different card index systems, one so she can search for members and the books they have out and one so she can search for books to see which member has it out! The Systems Analyst might do a questionnaire to pupils to find out what they think of the library service and many might comment on how slow the system is or what features they actually like about the current set-up. The Systems Analyst should write a list of all the problems as they come across them! They should also write a brief description explaining why each problem happens. This is often best done as a table.

32.8.7 Requirements Specification (RS)

When the Analyst has fully investigated the problem area of a business, they should have produced the following deliverables for the Systems Analysis stage of the systems life cycle:

- Written descriptions of current methods.
- DFDs of current methods.
- System flowcharts.
- Structure diagrams of current methods.
- A data dictionary of the current system.
- A table of problems and why they exist.

In summary, the Analyst now understands the business, the problems in detail and the methods used at the moment. Note that they have as yet not started work on designing the detail of a new computerised system. Having said that, the Analyst will have been considering how the problem can be solved. They will have been investigating alternative solutions and weighing them up against each other. They will have a good idea where the solution will come from, but they haven't got down to the detail of designing the new system yet. This is because they have not actually agreed with the customer what the final system will be able to do! This is the next job in this stage. The Analyst should work on producing one of the most important deliverables in the project, the Requirements Specification. This document will form the contract between the company needing a new computerised system and the company who will make the new system. It lists what the customer expects the system to be able to do and how, when the project is complete, the final product will be judged.

Each item in the Requirements Specification should state a way that its success or failure can be measured. When the project is finally finished, it is the Requirements Specification that is used by the customer to check that they have got what they agreed to buy and it is used by the Systems Analyst to check that they have made what they said they would make, to the standard they said they would make it to.

32.8.7.1 SMART requirements

When you are writing a Requirements Specification, one way to help you write excellent requirements as opposed to wishy-washy ones, is to ensure each requirement is **SMART**.

S M A R T

<u>S</u>pecific.	Each requirement must be clearly described and easily understood by everyone.
<u>M</u>easurable.	You should be able to write down tests that can measure the requirement. Requirements such as 'user-friendly', 'easy to use', 'effective' and 'pleasing to the eye' for example, are difficult to measure. 'Fast' is easy to measure because you can set tests e.g. must be able to display a customer's account details in under 0.5 seconds.
<u>A</u>greed.	The customer and Project Manager must agree on each requirement and how it will be measured.
<u>R</u>ealistic.	Each requirement must be realistically achievable, taking into account the resources and budget.
<u>T</u>rackable.	The project manager must be able to monitor and measure the progress of the achievement of a requirement. For this to happen, the requirement must be something tangible, something that they can see being developing, something that they can tell has been finished.

32.8.7.2 The process of producing an agreed Requirements Specification (RS)

The Systems Analyst will use the written description of problems as a starting point to producing the RS. They will produce a '**draft RS**'. In the draft RS, the Analyst will set out what he thinks the customer should expect from the new system, what the new system will be able to do and how success/failure will be measured. He will then arrange a meeting with the customer so that they can **discuss the draft**. Both the customer and the Analyst may well want to make a few changes as a result of the meeting. The Analyst then goes away and **re-drafts the RS**. Another meeting is arranged and the re-draft is discussed. If everyone is happy, both parties **sign** and **date** the RS. If more work needs to be done on it, then it is re-drafted again.

32.8.8 Possible outline solutions

Once the Requirements Specification has been completed, the Analyst can start outlining solutions. They need to think about what software will be used and how the features of the software will be employed to meet the requirements laid down in the Requirements Specification. It is always possible in a piece of software to approach a problem in different ways. The Analyst should document some of these possible approaches. They could also look at different applications to solve the problem and compare and contrast the ability of each application to meet the needs of the Requirements Specification. They may also look at designing a completely new application from scratch. Whichever of the solutions is preferred, it should be justified. There should be a clear statement which identifies the features of the software that will solve the problem as laid out in the Specification.

32.8.9 Hardware and software constraints

The Analyst needs to produce a list of the hardware and software that is being used in the current system (if applicable). When a list has been drawn up of what exists currently, they should then comment on how this may or may not impact on any future solution. They might be able at this stage to identify that certain software is being considered for a solution, but the software cannot be run on the existing machines because they are not of the right specification, for example. It may be that there are only standalone computers at a business but the analyst is thinking of using email and shared resources that require a network. The Analyst should then draw up a list of any new software and hardware that will be needed. They should also justify these proposed purchases to the company so that they are clear about the reasons for spending their money on this equipment. Once the Requirements Specification is completed and agreed to, the real work can begin! The Systems Analyst can start designing the detail of the new computerised system in the next stage of the systems life cycle - the Design stage.

32.9 The Design stage of the systems life cycle

By the time the analyst gets to the Design stage they will understand the business and operations in detail, understand the problem area in detail, have agreed with the customer what they want the new system to do and will have outlined the solution. They now need design the detail of the solution to the problem.

What happens in the Design stage of the systems life cycle?

1 → **There should be a description of the overall design using diagrams and comments.**

2 → **The user interface should be designed.**

3 → **All reports and screens and navigation systems should be designed.**

4 → **Data validation rules and verification procedures should be specified.**

5 → **Data volumes should be calculated and storage and backup defined.**

6 → **Functional descriptions of all procedures, modules, functions etc should be produced.**

7 → **A complete test plan with the data to be used and predicted outputs should be done.**

When the Design stage is complete, it should be possible to give all of the design documentation to any programmer or builder of systems and they should be able to construct the system **from those documents alone**! This statement is important. It tells you the amount of detail that is needed and the clarity of communication that is required by the documentation in this stage.

32.9.1 Written descriptions

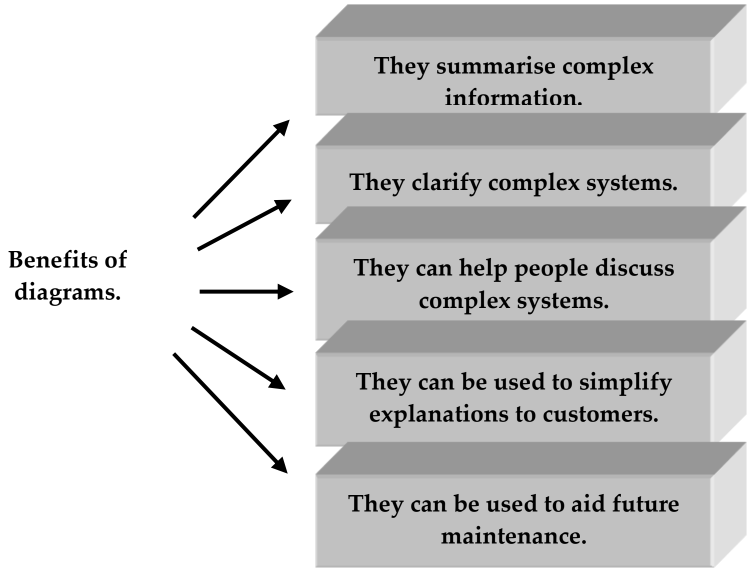
A good starting point for the Design stage is for the Analyst is to write out in the future tense how the proposed system will work. They should try to picture people using the system and describe what they are doing. They should describe each task a user needs to perform and how it is carried out. They need to paint very clear pictures with their descriptions so that they are understood by anyone who needs to refer to them.

32.9.2 Diagrammatic representations

Once they have described the system in writing, they should then produce diagrammatic representations of the design. This will involve producing Dataflow Diagrams and Structure Diagrams of the proposed system. These were described earlier. It is useful to know that there are other diagrams that could be used here as well as the ones already mentioned. Although we will save the detail about these and how they are used for later chapters, we will briefly mention them.

- **Entity-Relationship diagrams**. These diagrams summarise how records in a database are to be organised into tables and how the tables are related.
- **Entity Life Histories**. These diagrams summarise what happens to a particular record in a database. For example, they will show that at some point a new record is created, gets amended, and gets removed from the system.
- **Normalisation**. This is a technique that helps produce an efficient database design.

Diagrams generally are a good thing for analysts, designers and customers:



32.9.3 Input design
When designing the way that data gets from the outside world into a system, some questions need to be asked. These include:

- Will the data be collected on paper forms first or will the data be entered in directly?
- In what form will the data be in if it is entered in automatically? For example, will it be coded up in a bar code, or using MICR, or using OCR, or in some other way?
- What hardware exists, or will exist, to get the data into the computer? Will there be keyboards, a graphics tablet, touch-screens, microphones for voice recognition, for example?
- What skills will the operators have? Do any disabilities need to be taken into account, for example?
- What kind of interface will be appropriate for the system? For example, is a command line interface appropriate, or a menu-driven interface, or a different type of interface appropriate?

32.9.4 Output design
When designing the way that data will get from the computer system into the outside world, similar questions to the design of input methods need to be asked. Who are the users, what are their skills, do they have any special needs, where will the output come out, where will it be used and what will be done with it? The key question to remember here is, '**Who needs what information from the system, when and in what form?**' There are a number of output techniques that could be considered.

- Audio could be used to signal alarms in noisy, busy factories or to signal a break-in in a house.
- Key information on a VDU could be colour-coded or a different font size or style used.
- Important information could flash.
- Pictures could be used to represent or reinforce information.
- Graphs can be used to display information such as temperature trends over a period of time in a factory.
- Printouts could be used so that information is easily transported.
- Plotters can be used to provide large accurate drawings in an engineering design department.
- Dot matrix printers can be used to provide physical copies of key documents.

32.9.5 Data dictionary (following on from the Analysis stage)
A data dictionary needs to be produced for the new system. The information for this will have been collected during the systems analysis stage. For each data item that will be part of the new system, the designer should state the name of the piece of data and any synonyms, the data type, the validation rules for the data, an example, origin and use of the data. It is also helpful to specify how verification of the data items will take place. In addition to this, the designer should calculate the volume of data that is likely to be in the new system. This can then be used to specify appropriate data storage and backup systems.

32.9.6 Program specification
Solutions may require some programming. Part of the design documentation, therefore, needs to describe the functions that any code will perform. These can be described by writing a description of each function in detail, writing pseudo-code for each function or producing diagrams such as a flow chart or a Jackson diagram. The specification will also identify the language to use and program libraries needed along with the routines within them.

32.9.7 Test plan

The test plan should be produced in the Design stage of the systems life cycle. It should be detailed and cover:

- white box testing
- black box testing
- alpha testing
- beta testing
- acceptance testing.

Test plans are written to ensure that the product does what it is supposed to do. Where appropriate, the actual test runs to be carried out should be specified. The actual data that will be used in each test should be stated as well as the predicted outcomes for each test. It is usual for each test, that a typical, atypical, borderline, extreme and silly piece of data is used (so each test is in fact made up of five parts).

32.10 The Implementation stage (building the design)

This stage is concerned with taking the design produced in the Design stage and turning it into a real, actual working system. This is not the same as installation. Installation is taking the finished product from the design company's premises and putting it into the customer's premises. The Implementation stage is all about making a product from the design documentation. If the design documentation is in enough detail, then this stage can be carried out by any appropriately skilled, experienced person.

32.11 The Testing stage

Testing should now be carried out. It should follow the test plan produced in the Design stage. All results should be documented, to prove that they have been carried out. Ensuring that a system works is very important.

- A company that builds systems has a reputation to protect. They do not want to damage this by producing products that get into the news for the wrong reasons! In some cases, it could cause the company to go bust.
- They need customers to say good things about the product they bought because 'word of mouth' will generate new business. This will help the company make bigger profits.
- A system that is unreliable will soon be avoided by users. They will find it frustrating to use and will stop using it. Their company can only suffer because they have invested a lot of money in a system that was supposed to bring benefits and a system that doesn't work often makes matters worse.
- A system that causes accidents will leave a company open to legal action.

32.12 The installation stage

Once a product has been built and tested, it needs to be installed in a way that ensures minimum disruption to the business.

- Staff training on the new system must take place. This should include those who will use the system and those who will support others in the initial phases of installation, for example the Network Administrator or the various managers. This has implications for the business. If staff are training, they are not working! The cost of this should be taken into account when the project is planned.
- When a new system is to replace an old system, the data files kept on the old system needs to be transferred. Someone has to actually do this and this takes money, time and resources. In addition, any data transferred to the new system should be current. It would be of little use to transfer data from the old system to the new system one day but then not use the new system for a week. There would be one week's worth of data out-of-date!
- The hardware and software must be in place. Any additional hardware and software must be bought and stored somewhere until they are installed. They then need to be set-up and checked.

There are 4 implementation strategies that could be used. These are:

- parallel running
- pilot running
- direct changeover
- phased implementation.

32.12.1 Parallel running

The new system is run alongside the old system. Both systems operate together. This allows the new system to prove itself before the old system is abandoned - data generated by the new system can be compared to data generated by the old system. It also means that staff can be trained and gain confidence in the new system. Of course, if you are running two systems together, that means twice as much work for everyone for a short time!

32.12.2 Pilot running

According to the BCS, pilot running is when the new system is run alongside the old system, but only a portion of the data is actually used in the new system. This method is less of a drain on resources. Data from part of the new system can be checked with the old system, but you cannot check how the whole system will react until you have got the whole system up and running. The term pilot running can also be applied when a system is to be installed in a chain of stores e.g. when a new store accounts system is to be installed in each of twenty stores in a chain. The whole system will be put intro one store and piloted. When it has proven itself, it will be rolled out to other stores, one at a time.

32.12.3 Direct changeover (Big Bang)

The old system is stopped and the new system is started. This might happen over a weekend, for example. If something goes wrong with the new system, then it has to be sorted out because you cannot fall back on the old system. Staff training needs to take place in advance with this method.

32.12.4 Phased implementation

Parts of a new system completely replace parts of an old system, whilst the old system continues to be used as required. The part of the new system that has been installed can be used for staff training and can prove itself before the next part of the installation takes place. This method takes longer than the direct changeover method. A company with 10 branches may install a new accounting system in one branch first, for example. They run it in the branch until it has proven itself and possibly bring in staff members from other branches for training. Once the system has proven itself in one branch, it can then be phased into the other branches.

32.13 The Maintenance stage

When a system has been completed and is being used, that is not the end of the life cycle. There will be situations where the designers of the system, or others, have to go back and make changes. This is referred to as 'system maintenance'.

32.13.1 Situations that trigger modifications

System maintenance may be required for a wide range of reasons.

1) A user has discovered a bug in a program. This needs to be rectified.
2) The law has changed. The system needs to be changed to ensure that it conforms to the new law.
3) New technology. Some new technology has come into being which has provided new opportunities. The system needs to be modified to take advantage of these new opportunities.
4) A new function is needed. The company may decide that they would like the system to do something that it cannot currently do so the system needs to be modified.
5) The business has 'outgrown' the system. For example, a business may have expanded and now needs to be able to handle far more data than it used to do.

32.14 The Documentation stage

There are two types of documentation that need to be produced. These are User Documentation and Technical Documentation.

32.14.1 User documentation

With any product, some help needs to be provided for the users to enable them to run the software successfully. With many applications, those facilities are provided as software, as part of the application. To access the on-screen help, a user might click on an icon or press a function key, for example. A pop-up screen would then appear. A typical set of help features of an on-screen help system include:

- An index to allow a user to scroll through help files.
- A search engine to allow a user to type in key words or use natural language to search for solutions.
- A FAQ (Frequently Asked Questions) facility so that users can find answers to common questions quickly.
- Internet links to helpful web sites and support groups so that a user can find further help if they need to.
- Tutorials, to show users how to carry out tasks. These are often animated and interactive.
- Examples to help a user, such as a typical way to construct a formula.
- Provision of a README file to tell the user important information before they get started using the product.

Paper-based manuals are still common. A typical set of features include a contents page and an index, to help users find what they want and annotated diagrams and writing, to explain how to use the functions within the software. This might include how to enter data, print out copies, get on-screen help, for example. There might be examples, to show the user how to do typical things, such as writing formulas. There could be a 'Getting Started Guide', to show users how to load up the software and configure it to their system. This might include extra instructions on how to configure the software for a network. There might be a 'How to get further help' guide, who to contact. The could be a 'Trouble-shooting Guide' for common problems, a description of the licensing agreement for the user and an explanation of the user's right to make a backup copy of the software.

32.14.2 Technical documentation

The purpose of the technical documentation is to describe how the system actually functions. It is not written with a user in mind, but to assist a technical person in the future. The technical documentation is needed by a technical person because software has a 'shelf-life'. Just because it is 'finished', doesn't mean that it will never need to be re-visited again for 'maintenance'. Maintenance is discussed shortly. Just as it would be difficult to maintain a car without the technical manual, so it would be difficult to work on software in the future without the technical documentation for the software. The technical documentation will typically contain the Requirements Specification, the hardware and software specification, all of the documents from the design specification, including any program specifications and how to configure the system.

32.15 The role of the Requirements Specification at the end of a project

When we discussed the Requirements Specification, we said that it was possibly one of the most important documents in a project. The Requirements Specification constitutes the contract between the company buying a solution and the company building the solution. The Requirements Specification is used at the end of the project to see if the project does what it is supposed to do. Each item in the specification is checked to see if it has been done. If all items have been completed and completed to the required, stated standard, then the product is finished! It can be handed over and paid for! If some items are not completed, or completed but not to the required standard, then negotiations need to take place between the company and the maker of the product. They might agree a price reduction. They might agree an extension to fix the problems. They might agree that the company created the situation that has caused a problem and that they should pay more money than was originally agreed. Or they might not be able to agree why the product is not doing what it is supposed to do and they have to go to court to fight it out!! Government IT projects are notorious for being completed late, not to budget and to a poor standard and very often a very public 'blame-game' erupts between the Government and the contractors. They often then end up in court trying to sort out their problems.

32.16 Evaluating a system

After a period of time, a project may be fully evaluated. A contractor needs to learn lessons for the future. They need to see how they can improve future projects. They will want to see how accurate their cost and time estimates were so that they can more accurately quote for projects in the future. The customer will also want to evaluate the project, to see if it was worth doing. Between the contractor and the customer, they can evaluate a system by asking a number of questions. These should be asked after a period of time, to allow the new system to bed in. Questions that could be asked include:

- Does the system do what was specified in the Requirements Specification?
- How many bugs have been reported?
- How are the users of the system finding it and when there is a problem, is it easy to get help?
- How many times has the system crashed?
- Does the new system speed up jobs that used to take a long time?
- Is the company making more money?
- What benefits has the customer identified?

32.17 Hopper Ltd - a case study

Hopper Ltd is a company that makes a wide range of Space Hoppers! These were big in the Summer of 1971, faded away out of fashion but now they're back! (Do a search on the Internet to find out what they are!) It has an office building, a factory site 100 meters from the office and a warehouse next to the factory. All of the admin, sales and marketing are done in the office building, the Hoppers are made in the factory and the warehouse is used to store the materials used in the production of Hoppers and also the finished products.

The office, factory and warehouse are constantly communicating during the day. For example, the office sends details of new orders to the factory, the office sends requests to the warehouse for orders to be parcelled up and sent from completed stock, the factory sends requests for materials to the warehouse and the warehouse sends details of completed orders to the office. Currently, it keeps details of both its suppliers and its customers on a paper-based record-keeping system. Messages are passed from one building to another by hand.

Miss Jones is the General Manager. She's had an idea! Could all of the communications be done on a network, with a database to keep records? She feels sure there are lots of opportunities for computers to make the business more efficient than it currently is, but isn't a computer expert. At the next managers' meeting she outlines her ideas. It is agreed to call in IT-R-US Ltd, an IT consultancy, to carry out a feasibility study. They send in their most experienced Systems Analyst - you! Remember - you have been given a scenario, so you must relate your answers to that scenario wherever possible - and most of the time it will be possible. Make sure you know a few tricks and tips to relate answers to scenarios.

1) If you don't relate an answer to the given scenario wherever you can, what is the consequence for the marks you can be given for that question in an exam? (1)
2) What does a Systems Analyst do? (2)
3) Miss Jones has arranged a meeting with you to discuss the problem. Explain the importance for both you and the company of defining the problem accurately. (2)
4) You spent one week at the company doing a Feasibility Study. Suggest five areas that would appear in your Feasibility Report. Try and relate your answer to the scenario. (5)
5) You have now just completed your Feasibility Report. What happens next? (2)
6) As a result of your Feasibility Report, it has been decided to proceed with the project. What is the output from the analysis stage of the project? What is it for? (4)
7) Write down three examples of methods you want to use at Hoppers Ltd to collect information. (3)
8) As part of your investigation, you drew up a Dataflow Diagram of how data flows around the system at the moment. State exactly what a DFD shows you. (4)
9) You produce a systems flowchart. What additional information compared to a DFD does this show you? (2)
10) Generally, suggest reasons for producing any technical documentation. (4)
11) Give examples of the information you would expect to find in a data dictionary. (4)

NOTE: Some of what you produced in the Requirements Investigation and Analysis stage can also be part of the Design stage. There is an overlap, but don't let it confuse you. The Requirements Investigation and Analysis stage is all about finding out the information needs and requirements of the new system, the starting point being to find out all about the current way of doing things. The Systems Analyst needs to become a bit of an expert in the way the business is run. The design stage is about specifying a design that will meet the Requirements Specification produced at the end of the Investigation and Analysis.

12) Clearly explain what is meant by the 'user interface'. (1)
13) You have finished the design of the new system. You have specified the hardware needed, the software needed, the user interface and output design. You have done system diagrams and defined the data. You have handed over your design to a programmer, who has now built the system. They are ready to test it. They are going to write a Test Plan, putting all the tests into a table. Suggest headings for the table. (5)
14) What kinds of data should be selected when choosing data for test plans? (4)
15) The designer has finished testing the new database using black box and white box testing and wants to move on to alpha and beta testing. Describe what is meant by each of these types of test strategies. (4)
16) The new system has been fully tested. State the four possible implementation methods you could select. (4)
17) You have been asked to provide on-screen user guide. State four facilities provided by an on-screen guide. (4)
18) You have also been asked to do the technical documentation. Who is it aimed at and what is its purpose? (2)
19) State four documents or diagrams you would expect to find in the technical guide. (Do not include an index or contents page, although these should be part of the guide). (8)
20) The project has finished, although, as you know, all software has a limited life span. Explain the meaning of the term 'software maintenance' and suggest three situations that might trigger some software maintenance. (4)

Chapter 33 - Common applications

33.1 Introduction

In this chapter we will identify the features found in a range of business, commercial and industrial applications. We will describe some typical systems and pick out the computer features, identifying why IT has been used.

33.2 A stock control system (including order processing)

Many shops now use stock control systems. The term 'stock control system' can be used to include various aspects of controlling the amount of stock on the shelves and in the stockroom and how reordering happens. Typical features include:

- Ensuring that products are on the shelf in shops in just the right quantity.
- Recognising when a customer has bought a product.
- Automatically signalling when more products need to be put on the shelf from the stockroom.
- Automatically reordering stock at the appropriate time from the main warehouse.
- Automatically producing management information reports that could be used both by local managers and at Head office. These might detail what has sold, how quickly and at what price, for example. Reports could be used to predict when to stock up on extra products, for example, at Christmas or to make decisions about special offers, discontinuing products and so on.
- Sending reordering information not only to the warehouse but also directly to the factory producing the products to enable them to optimise production.

33.2.1 Advantages and disadvantages

Stock control systems ensure that just the right amount of stock are on the shelves. If there is too much stock, it ties up a company's money, money that might be better spent on reducing the overdraft, on advertising the business or on paying for better facilities for customers, for example. Too much stock means that some perishable products might not sell and would have to be thrown away and this would reduce a company's profit. If there were not enough products on the shelf, they might run out. If this happens, they would lose business and again, profits would not be as good as they ought to be. Stock control systems save a lot of staff time. Savings may be possible by reducing the number of staff needed in the business thereby improving profits. A stock control system will not remove the necessity for checking what is on the shelves regularly - things get stolen and these won't be recorded. Stock control systems also mean that a business may have to close down while the system is changed from a manual one. They also involve a considerable investment in equipment and support. Stock control systems require training and some staff may find them difficult to use. They can also break down so a procedure needs to be in place so the business can continue to trade. This may involve further costs as well, perhaps in the purchase of backup equipment or in the purchase of a support agreement. Usually, the benefits of a stock control system outweigh disadvantages.

33.2.2 An example

Imagine a shop that is part of a chain of shops. It has no computer systems. At the moment, the owner of the shop orders goods. This takes up her valuable time. She knows what to order by using her experience of how quickly things sell and by looking on the shelves and seeing what is there! She also has to produce regular reports about how the business is going for Head office. This again is very time-consuming. Headquarters decide to introduce a stock control system. One simple solution would involve getting:

- A method of scanning in products at the checkout.
- A computer with a modem.
- A phone line if they haven't already got one.
- Some connections between the checkout and the computer and between the computer and the phone.
- Software.
- An ISP.
- Computer systems set up at the warehouse, the factory and Head office.
- Some training on how to use the system.
- Some support for when things don't work.
- Backup systems in case the computer breaks down.

33.2.3 A point-of-sale system

The place where the customers take the things they want to buy is called the checkout. Another name for the place where money is exchanged for goods is the Point Of Sale (**POS**). We have already said that part of a computerised stock control system would involve scanning goods at the point of sale. A system that is part of a computerised stock control system is known as an Electronic Point Of Sale (**EPOS**). To set up an EPOS, you would need:

- Coded information on each product, for example, bar codes.

- A method of quickly getting codes from a product into the till. If you were using bar codes, you could scan them in using a laser.
- Hardware and software that validates products as they are scanned and warns the checkout assistant if a product doesn't pass validation. Bar codes include a check digit for this purpose.
- A method of allowing a product code to be entered manually, when the bar code is unreadable.
- A method of sending the data from the product, via the EPOS, to the central stock control computer.
- A method of getting information about the product, for example, a description and the price of the product, from the central computer to the EPOS. This can be used to generate and print out an itemised receipt.

Once a customer has had their goods scanned, a receipt is generated and payment is due. This could be in cash or by cheque but it is more common to pay by credit card. Not all EPOS systems can take credit card payments, but most can. If an EPOS can then it is known as an **EFTPOS** system, or Electronic Funds Transfer at Point Of Sale. In this system, a customer hands over their credit card (or other payment card). The magnetic stripe on the card is scanned and the information held on it is quickly sent to the credit card company. They check that it hasn't been reported stolen or the credit card limit hasn't been reached. They then send back to the checkout assistant an authorisation to accept payment. A credit card slip is printed out on two pieces of carbonated paper using a dot matrix printer. (This is just about one of the last major uses of the dot matrix printer). The customer signs the top copy (and indirectly, the bottom copy is also signed). They are then given a copy of the slip and can leave with the goods. The money is transferred electronically from the customer's bank account to the shop's bank account.

33.3 Marketing

Marketing is the term given to finding out or anticipating what a customer wants and then setting about attracting those customers by designing the right product and by directing customers to the product through advertising.

We have already mentioned loyalty cards in passing and these are very important for marketing purposes. A supermarket, for example, can build up a profile of each customer's shopping habits. Every time a customer shops, the purchase details are recorded in a master file. Various details can be recorded in a transaction file, such as what the customer bought, how much they spent and when they shopped. The master file can be regularly updated using the transaction file and this builds up a profile of the customer. This information can be used to target promotions and advertising more accurately at individuals. This is important because advertising is one of the biggest expenses in most businesses. Computers can play a role in a wide range of other marketing functions. For example:

- Special promotions (e.g. two for the price of one) can easily be done using the software provided as part of the stock control and EFTPOS systems.
- Reports can be generated that show how well a product sells e.g. by area, by shop, over time or by promotion, These reports can help the marketing department select the best promotion method.
- DTP software can be used to produce promotional literature.
- Email can be used to market products directly at individuals. This could be done by sending out special promotion vouchers that the customer can print off or sending newsletters, for example.
- Questionnaires can be produced and the results analysed and presented using spreadsheets, for example.

33.4 A payroll system

Consider a factory with a lot of workers. Each worker has a record that holds information like their name, address, what their hourly rate is, how much they have been paid so far this year, how much tax they have paid so far this year, their pension contributions and their National Insurance contributions. These details are held on a master file on computer. A master file is simply a file that stores the main data - the data doesn't change that often but can be updated when necessary. In this case, the master file is the file of employees. It doesn't change that often, possibly only once a week when everyone's pay is calculated or when someone tells them about a change of address.

Each day, the workers 'clock in'. Their ID number, the time they clock in and the time they clock out is recorded on a 'transaction file'. A transaction file is simply a file that records recent data. In this case, the file holds the details of who has worked what hours over the last week. At the end of the week, everyone's pay is calculated using both the master file and the transaction file. The master file and the transaction file are retrieved. Then a special payroll program is run that takes the master file and updates it using the information held in the transaction file. Not only is the master file updated, however, but pay slips are also printed out. The transaction file is wiped clean, ready for the following week's data.

33.5 Automatic Teller Machines (ATMs)

The same kind of system is used with ATMs (cash point machines). Everyone's bank account details are held on a master file at their bank's Head office. Every time a particular ATM is used, the details of the transaction are recorded in a transaction file. In the middle of the night, the ATM communicates with the Head office using a modem. It sends the Head office details of all of the transactions at that cash point in the last 24 hours, as recorded in the transaction file. This information is used to update the master file. The transaction file is then erased ready to store the next 24 hours of transactions.

33.6 A process control system

A process control system is a computer system that automatically monitors and reacts to changes in a system. We have seen an example of this type of system in the chapter on real-time systems.

33.7 Computer Aided Design (CAD) and Computer Assisted Manufacturing (CAM)

CAD software is used to produce design drawings and diagrams on a computer. CAD software could be used to design a new chair, a football stadium, a microchip or a new wing mirror for a car. Traditionally, when something had to be designed, an engineer would sit down with some paper and produce a technical drawing. There are many advantages for the engineer and the company, however, in using CAD to produce the design.

- Templates and old designs can be used to speed up the design process of new products.
- Different designs can be relatively quickly tried out.
- Colour, special effects, 3D views and animation of designs are possible to help everyone 'view' the design before it is built. This may be especially impressive when shown to the media or a customer.
- It is possible to zoom in on particular areas of drawings.
- CAD designs can be automatically linked to software that costs out the design, saving a lot of time in manual calculations and avoiding human error.
- Designs can be stored on computer. Designs done on large pieces of paper have to be stored, archived and retrieved. This used to involve whole rooms and employing staff to look after storing drawings.
- It is easier and takes up less space to backup drawings if they are held on computer.
- Designs can be emailed quickly to other departments anywhere in the world. This is useful because it removes delays created by the post and aids teamwork between members in different parts of the world.

When designs are done on computer using CAD software, some consideration should be given to the best hardware to use. For example, a large VDU will help to see more of a particular diagram and will allow better use to be made of higher resolution screens, a plotter will allow very large drawings to be produced to a high degree of accuracy and a graphics tablet may be useful because it allows the designer to draw in a more natural way.

We have already said that one advantage of CAD designs is that they can be used to automatically cost out the design. It is possible, however, to link the design done on a computer using CAD software to the actual machines that produce the product! When a design is completed, details of the design can be converted into instructions for the machines that produce the product. The machines can then start producing the product without a human having to set them up. This saves time and reduces the number of staff needed in the manufacturing process. If any modifications are made in the design using CAD, these can simply be passed to the machines and the changes will be made automatically - the machines don't have to be stopped. This improves the number of products made and so improves efficiency. When production is linked to CAD as described above, it is known as Computer Assisted Manufacturing, or CAM. CAM is a term used not only to link the design process with the manufacturing process. It also encompasses the management of the supply of components needed to manufacture the products and the factory planning process, to ensure that production and output are maximised.

33.8 Generic applications

You should be aware of and ideally have some experience of the selection of common generic applications for a particular purpose. The phrase 'generic software' or 'generic applications' simply means that a particular piece of software can be used in many different situations - it isn't a piece of software that can only be used for a very specific business application. There are many examples that could be given in this section. We will highlight some of them. It should be pointed out that the functions performed by a type of software have become very blurred in recent years. We will see some examples of this as we describe each type of application.

33.8.1 Word processing software

The tools that come with word processing applications should ensure that a user (who actually uses them) produces high quality documents. When letters were written using typewriters, it was easy to make mistakes. When they were made, they could be 'typed over' with correction fluid but this did not look professional, especially if more than one occurred - the only choice was to re-write the whole document. (The author can remember having to re-type many pages for university projects!) In addition, it was possible that mistakes were not actually found in the first place because they were manually checked. The documents themselves would only contain text, although with time and some clever use of scissors, glue and a photocopier graphs and pictures could be included! Nowadays, word processing software can be used for a whole range of things. Examples of the types of applications include:

- Preparing letters and other documents requiring textual information.
- Producing mail merge documents and labels.
- Producing basic promotional literature such as certificates and flyers. This used to be the domain of desk top publishing software but most word processing software can easily combine pictures, text, different layouts and tables, for example.
- Producing a basic flat file database with records that can be sorted. More complex databases needing advanced features will require a database application. Nevertheless, tables of records with basic facilities can be produced in many word processors.
- Simple spreadsheets.
- Simple diagrams can be produced by many word processors because they come with drawing tools.

The expectation of perfectly word-processed documents came with the introduction of word processing software. There is little excuse for incorrectly spelt work because documents should be spell-checked and users can set the software to correct documents automatically. Grammar checks, whilst not perfect, ensure that users think about their English! The provision of thesauruses should ensure that users are not lost for words or use the same words too often. Documents can easily be made more interesting by adding borders, watermarks, pictures or tables. The days of boring-looking documents are long gone! The traditional role of secretaries has all but disappeared, too. They are much more likely to be highly skilled, better-paid personal assistants rather than simply 'typists'.

33.8.2 Spreadsheet software

Spreadsheet software applications are typically organised in a way similar to traditional book-keeping systems. This type of software is aimed at applications that involve the processing of numbers, for example, payroll systems and accounts systems.

- There are many different work sheets.
- Each work sheet is arranged into rows and columns of 'cells', or boxes, into which a piece of data is entered.
- Each cell and therefore each piece of data, has an address that can be referred to, for example, A4 or H12.
- Formulas can then be written into cells that refer to data in other cells. The results of the formulas are displayed.
- When a piece of data is changed, all the values displayed in cells where there are formulas are automatically updated.

Spreadsheets are not only used for processing data. Many successful databases have been set up in spreadsheet software! This is because spreadsheet software gives you the ability to manipulate rows of data, search for values and even do complex queries. Spreadsheet applications have had a major impact. Once a spreadsheet has been set up and has been tested, it can be re-used. Data can be entered and formulas will do the calculations automatically. They will be correct every time! Before spreadsheet software, calculations had to be done manually, which introduced the potential for human error. Some accountants' handwriting was very suspect and this was another potential source of error. Spreadsheet software produces very neat and readable work sheets. It is a small step in software to go from data to graphs and charts and reports. This was a large step before software was used.

Modelling is another major benefit of spreadsheet software. It is possible, for example, to build a model of the economy and run a simulation. Many 'what if' questions could easily be asked and tried out. For example, consider the question "What would happen if taxes were put up by a penny in the pound"? This could be tested in the model to see how the economy would react. This kind of modelling has enabled companies to produce very good business plans because they can 'try out' different market strategies in models before actually implementing them without risk.

33.8.3 Database software

This type of software is aimed at data processing, usually involving records. Applications could include a pupil database, a room-booking database for a hotel or a database to hold driving licence holders' details. Features typically include the following:

- The ability to store similar records in tables.
- The ability to set up user-friendly data input forms.
- The provision of validation routines so that both data consistency and integrity are maintained in the database.
- The ability to reorganise records quickly.
- The ability to search through records to find the ones that meet complex criteria.
- The ability to write formulas that can process data held in records.
- The ability to produce professional-looking reports.

Database applications now mean that it takes just seconds to completely re-sort thousands of records or search for records that meet certain criteria. For example, consider a pupil database. Perhaps one minute you need all the records in alphabetical order and the next minute you need them in tutor groups. Suppose you need the records of everyone who was born before a particular date. These are big jobs if the records are held in a paper-based system but will take only a few seconds on a computer-based system. It is very easy to mislay records in a paper-based system. For example, if somebody files a record in a wrong place, how can you find it again except by starting at the first record and working your way through all of them. This problem is practically eliminated with computerised databases.

33.8.4 Desktop publishing software (DTP)

Word processors often have many features of DTP programs. However, a DTP program should still have a wider range of tools to manipulate text, pictures and the layout of material on the page, for example. They are typically used for producing quality certificates, menus, promotional literature, business cards, posters, newsletters and newspapers. DTP software has put within everyone's reach the potential to produce very high quality documents and publications. Many people now use them to produce magazines for their hobbies and clubs. DTP packages typically come with many templates and wizards, so that even a novice can quickly put together a super-looking document. These can be printed out easily on printers that will organise what needs to be printed on which page. Printers that print automatically on both sides of the paper cost only a few hundred pounds. This would not have been possible a few years a go.

33.8.5 Presentation software

Many different types of multimedia presentation packages exist now. There are many advantages and disadvantages of doing a presentation to a group of people using a presentation package compared to say, using OHP slides.

- You can easily combine sound, animation, movie clips, text, special effects and scanned images in a multimedia presentation.
- Templates allow the production of quality multimedia presentations quickly.
- You can quickly try out different ideas for a presentation on computer but exploring ideas would take far longer if you were using a non-computer method of presentation.
- Presentations could be interactive. For example, a presentation could be run on a computer at the front of a library. The user could use a touch screen to navigate through options presented by the presentation. There would be no need for someone to actually be there doing the presenting.
- A presentation done using a presentation package could be sent all over the world by email and used. Again, if it was designed to be interactive, or to run constantly, you wouldn't need to have someone present it. It could be loaded up on a computer and run.
- It is usually straightforward to import text, graphs, or data, for example, from other pieces of software, especially if the presentation software is part of an integrated suite of programs. This can be very useful for business professionals who need to present very polished reports, a marketing plan or their vision for the future development of the business, for example.
- A little training is always required with any software packages.
- The software and the hardware to run the application can initially be expensive.
- What happens if you are doing a very important presentation and the computer breaks down? You need to have a backup plan and ideally some technical support at hand. Of course, the latter is not always possible but a sensible speaker needs to have a plan if the equipment doesn’t perform in the way it should!

33.8.6 Drawing software

This type of software is used to produce drawings and diagrams. These can be saved and then used in a web page or in a presentation or in a desktop publishing piece of work. They have many tools such as ones that allow you to draw shapes, add special effects and mix your own colours. Drawing software allows ideas to be quickly tried out, archived drawings to be quickly modified to produce new drawings and special effects to be added - again, all within the reach of people who wouldn't have been able to produce quality drawings only a few years ago. Drawings also take up a lot of space and engineering companies used to have whole rooms and staff dedicated to storing drawings. Now, they can all be stored on computer! They can also be easily retrieved and easily sent to people all over the world. Of course, this has a security implication that needs to be thought about. Physical drawings locked away in a secure room may arguably be more secure than drawings stored on a server that is prone to hacking, disgruntled employees, fire and so on.

33.9 When is generic software not appropriate?

1) If some new technology came into being, it may require features that are not available on standard packages. For example, when smart phones and tablets came into existence, new software had to be developed so that they could be given the features and functions we have all come to be familiar with. Generic software applications could not have done this.
2) Some machines are one-off designs, for example, the software needed to control space equipment remotely. Again, generic applications wouldn't be appropriate here.

3) Sometimes, a generic application can be used, but needs to be adapted through specialist programming. For example, if some data needs to be read into a database but it is not yet in the form that the database can use then some non-standard software must be written to put the data in a form that the database can use! A generic database application can be used but some additional software has been specially written to customise it.

33.9.1 Custom-built software verses off-the-shelf software

When a problem has been identified and a solution is being planned, both the Systems Analyst and the customer need to make an important decision regarding the type of software to use. Will a custom-built piece of software (sometimes called 'bespoke' software) be used or will a generic application be used? For example, if a company needed to create a customer database, do they commission a completely new application in Pascal or Java or Visual Basic or some other programming language, or do they commission a product made using an off-the-shelf generic application such as Access? There are advantages and disadvantages to both approaches.

A comparison of custom-built software with off-the-shelf software:

- You can buy and use off-the-shelf software straight away. Writing a new application can takes time.
- Writing a new piece of software may take hundreds of man-hours. This will add a great deal to the final cost of the product! Off-the-shelf software is generally cheaper.
- Off-the-shelf software has been around a while. Any problems with the software may well have been documented along with solutions and patches. The company who produced the software may have fixed many reported problems and released later versions. Bespoke software will have to go through the process of proving itself.
- There is very often a wide range of sources of help and support on the Internet for off-the-shelf software. Help may come from support groups, user forums and newsgroups, for example. This type of help will not be available for bespoke software.
- Before deciding to go ahead with off-the-shelf software, it should be possible to seek recommendations from other users, perhaps using the Internet. It should be possible to gauge how good the product is, how good the support is, for example.
- If a company writes a new application, then they should support it. This can be both a good point and a bad point. If they wrote it, then they should know all about it and should be able to provide quality support. However, they may well charge a lot of money for support, or the support they give may not be that good. Unfortunately, however, the company that bought the software will be stuck with them - they will not easily be able to find additional help!
- If a product is made using off-the-shelf software, then it should be possible to maintain the product using anyone who knows about the particular package. There should be a big pool of experts to select from. If, however, a bespoke product is produced, it may be harder for anyone except the original company to maintain. And what happens if they go out of business?
- Bespoke applications can be tailor-made for an organisation. It provides exactly what functions the company needs, whatever they are.
- Bespoke applications can be designed to integrate with whatever software and hardware currently exists.
- There may well be situations where an off-the-shelf application doesn't exist that can solve a particular problem. In this case, the only option would be to build a new application from scratch.

Q1. Do some research on the Internet. What information is held in a typical bar code for a supermarket product?
Q2. Do some research on the Internet. How do you calculate the check digit on a 13 digit ISBN number?
Q3. Do some research on the Internet. What measures have been taken in recent years to reduce credit card fraud?
Q4. Describe some benefits of a CAD system for producing technical drawings.
Q5. What is the difference between POS, EPOS and EFTPOS systems in a shop?
Q6. How can IT help in the marketing of a new product?
Q7. Define Computer Aided Manufacturing.
Q8. What is a 'software patch' used for?
Q9. Describe two benefits of bespoke software compared to off-the-shelf software.
Q10. What is meant by 'generic' software?

Chapter 34 - Information systems

34.1 Data verses information
Data comprises of a set of codes that have a structure. For example, you might have a set of integers (whole numbers) or a set of real numbers (numbers with a fractional part) or a set of strings (each string is made up of an ASCII character) and so on.

- Here is an example of some data: 12, 45, 34, 23, 44, 91, 5
- Here is another example of data: 12.66, 14.89, 3.65, 611.00, 723.04, 76.01
- Here is yet another set of data: Fish, Chips, Sausages, Beans, Peas

Each set of data has a structure. What the data lacks, however, is an **interpretation**. What does the data mean? It is not until you are told that the first set of numbers refers to the number of cars sold by Cooper's Garage in each of the last 7 months that you have information. It is not until you are told that the second set of numbers refers to a bank customer's balance over the last 6 weeks that you have information. It is not until you are told that the third items in the list are the most popular sellers in the local chip shop that you have information. Data is a set of coded, structured symbols. Information is when data is given a context.

34.2 Passive and interactive information systems
In the context of computing, an 'information system' is any computer system that has data in it that can be rearranged and accessed by users in the form of information. Information systems hold data on secondary storage devices, for example, on a CD or a hard disk. Sometimes, the data held on storage devices can be changed by users. Sometimes it cannot be.

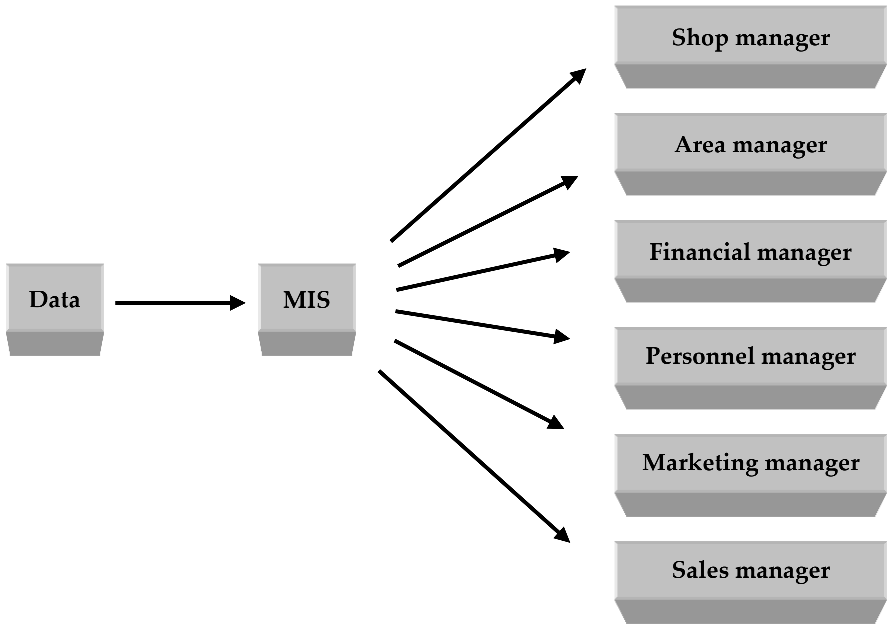
If the information held in storage can be accessed but not changed, it is known as a '**passive information system**'. An example of this kind of storage system would include an encyclopaedia held on a CD. You can access the information on the CD. You can even search through it. But you can't actually change the information on it.

Some information systems are known as '**interactive information systems**'. This is because you cannot only access the information, you can change it, too. An example of this is a school's database of pupils used by the secretary. The secretary can access any pupil's details. They can even search through all of the pupils' records to find ones that match some particular criteria and they can change the details held on the database, for example, when a pupil moves house. Another example might be a shop's database of products. It can be searched to see what products are available and it can be changed, perhaps as part of a stock control system. Each time an item is scanned at the checkout the number of items available is reduced by one. New products can also be added to the database and ones that are no longer sold removed.

34.3 Management Information Systems (MIS)
A Management Information System, or MIS, is a piece of software that collects data from a range of sources and turns it into information appropriate to a recipient's needs to enable them to make decisions. A company usually has some managers who are responsible for the day-to-day running of a business (Operations Managers). They need to make sure that there is enough stock, that the staff are well-trained and that the paperwork in the business is up-to-date, for example. There are also the Strategic Managers. Strategic Managers are responsible for planning the direction the business will take, forecasting what will happen in the future, controlling the costs of the business, marketing the business and the products that they sell, ensuring that resources are sufficient for the business to be successful, to name but a few. Strategic Managers may be at an area level or a national level or even an international level. Each type of manager, however, whether they are an Operations Manager or a Strategic Manager, has their own particular information needs that must be satisfied to enable them to carry out their job effectively. An MIS seeks to satisfy all of their individual information needs.

34.3.1 Different layers of management have different information needs
Consider a mobile phone chain of shops. Each shop will have a manager. Each shop manager needs to know how many of each type of phone they have in the stockroom. They need to know how many they sold last week, how much money they took in, who the best salesperson was and when to re-order (often done automatically) for example. In other words, each shop manager needs to have information relating to the operations within their own particular shop. An area manager, however, will not need to know the detail of what is happening in each shop. They will be concerned with summary financial information, to get an overall picture of sales in their area. They might want to know how much money each shop in their area took in last week in comparison to the target sales for that shop. The strategic managers at the head office will want summaries of each area. The financial manager will want financial summaries. The personnel manager will want summaries of the numbers of staff, the

hours each worked, the cost of overtime and the like. The marketing manager will want specific information about one type of product that they have been heavily promoting. They may also want to see some graphs that compare their prices to their competitors over the last week. The sales manager may want to see a breakdown of total sales by salesperson or by product.



Different managers have different information needs.

Each manager, then, has their own information needs. An MIS seeks to meet those needs by doing a number of things.

- It will provide reliable information. If information given out by any system proves unreliable, managers will quickly abandon it. Once a system has lost the confidence of the people it is supposed to serve, it is very difficult to regain their trust.
- An MIS will provide up-to-date information. Information has a 'shelf-life'. Its usefulness soon lapses with time. It can quickly be given the label of 'historical data' and be of limited use to managers trying to effectively run a business.
- A good MIS will give the manager the information they need. Too little information and they won't have all of the facts upon which to make business decisions. Too much information, however, and they may not be able to find the facts they need. An MIS that gives a manager the information they need will potentially improve the quality of the manager's decision-making!
- A good MIS will not only give a manager the right amount of reliable and up-to-date information. It will also present the data in a way that is best for that user. This may involve giving the manager a graph, a table of results or a written description, for example.
- An MIS should be able to provide knowledge that previously took time to collect using other methods. This means that a manager could have more time to allocate to other activities, such as customer care.
- An MIS should be able to better illustrate trends in areas of business. This should help with any future planning that needs to take place.
- An MIS should help a company to become more profitable and to give it an edge over their competitors.

An MIS is *not* a data processing system. A data processing system takes raw data and processes it into a regular and well-defined form. It is not rearranged into different forms for different managers so that they can make better decisions. Examples of data processing systems include:

- processing monthly bank statements for customers
- processing employee payslips
- producing a list of supermarket items that must be reordered
- producing a list of people who have not paid their annual subscription to a club
- producing a list of pupils in a class.

Examples of the output from an MIS include:

- a report showing the prices charged for flights to New York by competitors
- a report showing how many mobile phones were sold in each shop in a chain
- a report showing how profits varied each month over the last year
- a report detailing the overtime worked in each supermarket in a chain of supermarkets.

34.4 Knowledge-based systems (Expert Systems)

A 'knowledge-based system' is a synonym of an 'expert system'. It is "an application of artificial intelligence to a particular area of activity where traditional human expert knowledge and experience are made available through a computer package" (British Computer Society, 'A Glossary of Computing Terms'). In other words, it is a piece of software that has a go at replacing experts' knowledge and experience. In this section, we will describe the components of the software that make up an expert system.

34.4.1 The components of knowledge-based software

Knowledge-based software systems have identifiable parts to them. These are the knowledge base, the rule base, the inference engine and the user interface.

1) The knowledge base

The knowledge base is the name given to the part of the software that holds the facts. It contains the knowledge for a particular area of expertise, such as a medical diagnostic expert system or an engine fault diagnostic system.

2) The rule base

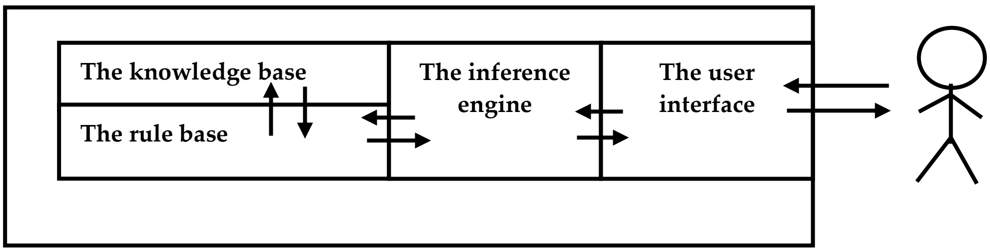
The rule base holds the rules which determine how the data in the knowledge base relate to each other.

3) The inference engine

This is the piece of software that works out how to apply the rules in the database of rules to the data in the knowledge base. It takes requests for information from the user interface and then searches through the knowledge base by applying the rules in the rule base. The inference engine retrieves appropriate knowledge from the knowledge base and passes it to the user interface software. It separates the user interface from the 'clever' part of the software. That means that different applications can be written by different people, each one having their own user interface - and they will all be able to use the same expert system.

4) The user interface

The user enters requests for information. They may do this by entering in answers to closed questions (questions which have only a few possible answers). The answers will result in the knowledge base being reduced further and further until only limited facts are left. These can then be returned to the user along with a probability factor. These four components can be represented with the following diagram:



The four components of knowledge-based software.

34.4.2 Talking about an expert system

We know that if we want to describe what an expert system is, we need to mention the knowledge base, the rule base, the inference engine and the user interface. We could also highlight the characteristics of an expert system in the following way:

- Expert systems are specialised. Each expert system attempts to be an expert in a focused area of expertise.
- When answers are returned, they are often given probabilities. For example, in a medical diagnostic expert system, a high temperature, headaches and a cough might return:
    - common cold - 61.78% chance
    - influenza - 38.21% chance
    - bubonic plague - 0.01% chance.

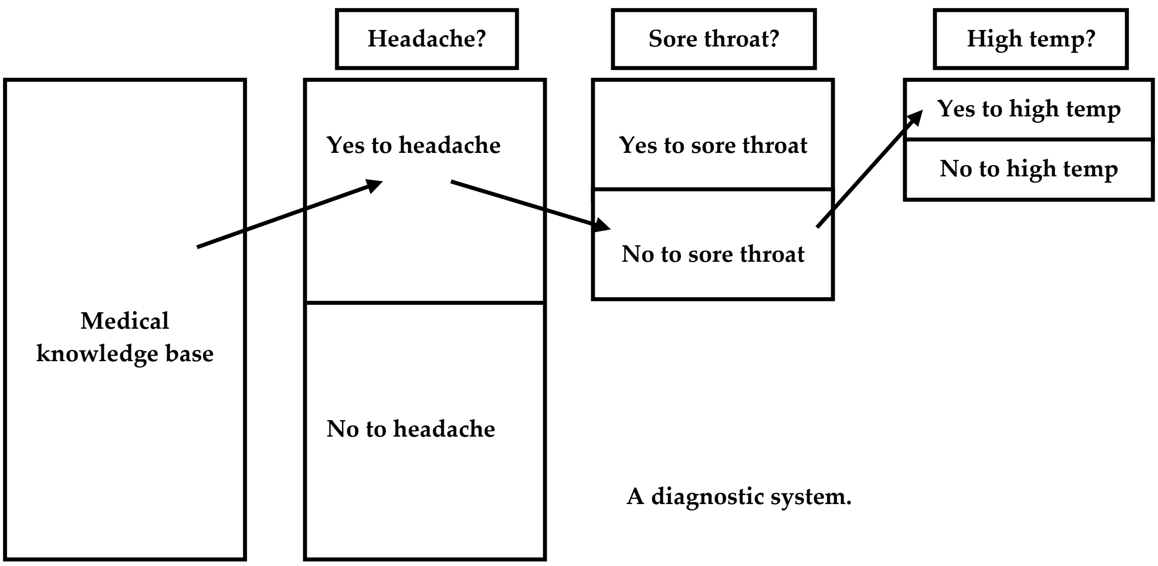
- A user is often asked a series of questions and must select an answer from perhaps two or three choices for each question. When the answers have been selected via the user interface, they are passed to the inference engine. This then triggers a search of the knowledge base using the rules in rule base. The resulting facts with a probability score are then passed back to the inference engine and on to the user interface.
- Sometimes an explanation or advice may be part of the answers returned.

34.5 Types of knowledge-based systems

There are three types of knowledge-based systems that you should know about. These are diagnostic systems, advice-giving systems and decision making systems. For each type, you should be able to describe a classic example.

34.5.1 Diagnostic systems

- These are systems that ask a question, the answer given by the user resulting in the knowledge base being reduced in size. The classic example is a medical diagnostic system.
- The first question might be 'Do you have constant headaches'? If the answer is 'Yes', then one portion of the knowledge base can be explored further. If the answer is 'No', then a different portion can be explored. Whatever the answer, the number of facts that will be looked at has been reduced.
- Suppose the answer to the above question was 'Yes'. The next question might be 'Do you have a sore throat'? Again, the answer can be either 'Yes' or 'No'. If it is 'Yes' then that reduces the part of the knowledge base for people with headaches even further to a part for people with headaches and sore throats. If it is 'No', then that will reduce the part of the knowledge base for people with headaches but without sore throats.
- Suppose the answer was 'No'. The next question might be 'Do you have a high temperature'? This process will continue until the number of facts left is very small. These can then be reported to the user, perhaps with a probability factor and some additional comments.
- We can represent the process with the following diagram. Notice how the knowledge base gets smaller and smaller with each question answered.



A diagnostic system.

Another example of this kind of system is a car engine diagnostic system. When a car has a problem with its engine, the problem can be diagnosed with the help of an expert system. For example, the first question might be 'Is there an oil leak'? If there is, that would reduce the knowledge base to an area that deals with problems involving oil leaks. If there isn't, that would reduce the knowledge base to an area that deals with problems not involving oil leaks. Another question could be asked that further reduces the knowledge base and so on, until the number of possible causes of the problem are small enough to report back to the user.

34.5.2 Advice-giving systems

You have already seen a classic example of this type of knowledge-based system, when you looked at stock-control systems. In this kind of system, a process is monitored by the software. When the software detects particular situations, it gives advice! In a stock control system in a shop, the stock is constantly monitored: the amount of a product in stock is decremented each time a customer buys a product and has it scanned and it is increased each time a delivery from the main warehouse is received. The stock control system 'knows' that when a product falls below a certain number, it should advise that a re-order take place. Stock control systems can also be called 'decision-making' knowledge-based systems if they are able to re-order automatically, without any human intervention.

Another example is the safety systems fitted on planes that help them avoid collisions. As the plane flies, it scans ahead, checking that the path is clear of other planes. If it does detect a plane coming towards it, it should advise the pilot to change course. Like stock control systems, these systems can also be called 'decision-making knowledge-based systems if they are able to automatically change course to a safe route, without human intervention.

34.5.3 Decision-making systems

In the section above, we saw that advice-giving systems can also be called 'decision-making systems' if they have enough information and the hardware and software to allow them to actually take decisions themselves, without human intervention. Many process systems are both, because they enable a user to select whether the system will be fully automated, or will simply report and advise. An automated share-selling expert system can be both. Instructions can be given to the software that tell it to automatically sell a share when it reaches a certain price or it can be told to give a message to the stockbroker, advising that the shares be sold. The stockbroker can then review the situation, discuss it with the owner of the shares and then take action.

34.6 Some advantages and disadvantages of expert systems

Software can never completely replace a human. For one thing, a human has hunches and intuition. Some might argue that removing these elements is a good thing! Humans can, however, evaluate completely new situations whereas an expert system is bound by rigid rules and their current knowledge base. Indeed, an expert system's knowledge base and rule base must be kept up-to-date. This may seem straightforward but if the expert system is in a third world country, for example, it means that some training will be needed so that somebody locally can do this.

To have an expert system means that the hardware and software must be bought. Not only may this be a problem in some situations, but careful thought needs to be given to maintaining the hardware and software - you don't want the system to stop being used the moment there is a problem, simply because nobody has been trained to maintain the system.

Expert systems may well be a wonderful substitute for situations where it is impossible to recruit highly trained people. They should also be much cheaper than the equivalent human. It may simply not be possible to pay for enough doctors to work in third world countries, or it may not be possible to recruit enough stockbrokers to work in a firm. There are some dangers, though. It may be that people come to rely too much on the power of the expert system! These systems are never perfect. They may contain software bugs. They may cause situations that have not previously been considered. Expert systems that automatically bought and sold shares around the world nearly caused a stock market meltdown in the 1990s because they fed off each other and caused each system to sell sell sell!

Q1. Use an example to clearly explain the difference between data and information.
Q2. Apart from a CD and a hard disk, name two other secondary storage devices.
Q3. Define a 'Management Information System'.
Q4. State the characteristics of a good MIS.
Q5. What is an 'inference engine'?
Q6. How would you go about making a new Expert System?
Q7. Discuss the pros and cons of online 'diagnose yourself' type Expert Systems?
Q8. Discuss the pros and cons of using medical Expert Systems for a very poor, third world country?
Q9. State the three types of knowledge-based system.
Q10. Use the internet for research. Apart from medical-related Expert Systems, state three other areas which have made use of Expert Systems.

Chapter 35 - The implications of using computers

35.1 A note on this chapter
Much of the knowledge required for this chapter will come from wide background reading. You can learn some basic facts and get a few ideas from these notes, but they are no substitute for you reading many articles about different applications and the way they have changed lives. This is not something that can be done at the last minute, a few weeks before exams! You need to get into the habit of reading three or four articles a week and making some notes about each of them. Start a folder just for articles that you've read so that when you come to your exams, you should have read many different articles.

35.2 Trends in computer use
The Internet is a good source for articles. Computers have had many effects on individuals in society. They have impacted on the way individuals work, socialise and run their lives. As a result of computers, an individual's values are now in constant flux. The moral and ethical framework that guides an individual is constantly changing as is the economic and legal framework within which lives are led.

35.3 Moral, legal and ethical
You can debate the subtleties of the meanings of these three words forever. We will use three very simple definitions.

- Moral – the law of God
- Legal – the law of the land
- Ethical – the law of groups

Morality is the notion of right and wrong in society. It is governed by what people feel is good or evil. This very often is tied historically to a country's religious heritage. The legal framework of a country comprises of those things written down that say what you can and can't do and the penalty you must pay if you do things that are not allowed. The ethical framework refers to groups of people, each group having their own principles and modes of behaviour. These are directly related to each particular group's activities. So, for example, doctors, teachers and MPS each have their own code of ethics.

35.4 The digital divide
The digital divide is the term used to describe some groups of people having access to computers, training on computers and computer-based information while other groups have limited access or none at all. For example:

- Third world countries, already economically disadvantaged, are finding themselves at a further disadvantage because their citizens generally do not have the same opportunities as those in developed countries. Their citizens generally may not have the money to buy expensive computers and there may not be the same opportunity of education that you and I take for granted. The priority may simply be to find enough food for that day!
- The digital divide can also refer to different groups of citizens within one country. Unemployed people, for example, may not be able to afford to buy computers, pay a subscription for Internet access or have the skills or confidence to get themselves on computer-related training courses.
- Another group of people who may find themselves disadvantaged are pensioners. Their opportunity to access modern technology may be limited by their income, their background and the perception that computers are for younger people.

There are many examples of organisations trying to reduce the effect of the digital divide. Schemes exist to give people better opportunities of access. For example, pensioners and other groups are having training plans designed especially for them and delivered at local schools and colleges, there is subsidised or free access is in place in many libraries and recycled computers are given to disadvantaged groups.

35.5 The law lags behind the advances of computers
The pace of change in computing technology is incredible. Opportunities exist for criminals to carry out crimes such as fraud in new ways using new technology. For example, emails can be sent pretending to be somebody from a bank and asking the receipient to login to their bank using a fake link set up by the criminal. This link goes to a website set up by the criminal that then allows them to capture the person's bank login and password details to steal money, a technique known as 'phishing'. Another fraudulent use of the Internet is to set up fake websites with fantastic offers, hoping to get people to pay for goods they

won't receive using their credit card details, which may then also be used by the criminals. Some activities are very 'crime-like' but aren't breaking any law. It may be difficult to prosecute somebody because the legislation isn't adequate and may take years to be strengthened. For example, for a number of years, it was very difficult to prosecute anyone who hacked into somebody else's computer system. The existing laws were difficult to apply to unauthorised access of data. The result was that a new piece of legislation had to be passed in the shape of the Computer Misuse Act 1990. Another example is the Data Protection Act 1998. It had to be produced because the existing Data Protection Act didn't fully protect an individual's right to privacy when data was sent over the Internet. When change happens, the legal system often catches up and offers protection a few years later! For many years, individuals and companies were using the Internet to send millions of unsolicited emails worldwide to people and companies. Sending SPAM annoys people because it slows down Internet traffic - it uses up 'bandwidth'. It causes problems for some Internet Service Providers because they cannot cope with the volume of emails going in and out of their equipment. It fills up people's inboxes with rubbish. Children can get sent links to pornography or other web sites. The problem has not gone away, but for now there is some legislation that gives users some rights to tackle SPAM. This is in the shape of the Directive on Privacy and Electronic Communication (2002/58/EC). This requires the senders of emails to ask receivers of emails to 'opt in' to receiving them. How effective this will be in preventing someone from sending bulk emails from outside of Europe remains to be seen.

35.6 The effects on a village

Many village facilities have closed because they are uneconomical. These have included post offices, banks and village shops. This has had a big effect on people in villages. These facilities often provided a social life, a community spirit, a place to meet and jobs for the local community.

Many people have complained about the effects of closing facilities. Bus services were often poor and some groups of people felt isolated, especially if they didn't run a car. Some of the complaints, however, have been reduced as a result of the Internet. This is because the Internet has started to change the way that people live their lives. Many people now have Internet bank accounts. They don't need a traditional bank anymore because they can apply online for a mortgage or loan, transfer money, view statements, deal with questions and problems, for example. Supermarkets are expanding their home delivery services nationwide. It is now possible to shop online and have an order delivered to your house. People are increasingly using email to send letters. They are increasingly using the Internet to socialise, using chat rooms with voice and web cam facilities.

35.7 The ease of software piracy

It is very easy to copy software, download music, download videos and access illegal material such as pornography from the Internet. Regardless of the latest safeguards in place, the nature of the Internet is such that the way around any copying problem can be found quickly. The real question for the individual is, 'Should I, just because I can'? Copying software from someone is theft. Downloading music and videos from the Internet is also theft unless, for example, it is freeware. Accessing pornography may well be possible but it may also be illegal. An individual has quite a few moral and ethical questions to ask themselves before they walk along a particular path. They need to ask themselves what sort of person they are and how they want others to see them.

35.8 Changes in patterns of leisure

Children are playing a lot of computer games today. Many of the games are violent. They are not playing nearly as much sport as they used to and are not as healthy and fit as children of previous generations. These patterns of leisure worry some people - children may not be socialising and forming social skills in the same way as previous generations. As ADSL use grows, it is now becoming feasible for individuals to download bandwidth-intensive applications such music and video. This may be the beginning of the end for video shops and a more widespread change in the way we buy and listen to music and video.

35.9 Changes in patterns of working

- People have more opportunity to work from home now.
- Many computer-based jobs, for example, writing software, can be done from anywhere in the world. This gives an opportunity for employers to hire cheaper people with the right skills. For example, India has a pool of highly-educated programmers that can be employed at cheaper rates than western programmers.
- Many mundane jobs have been replaced by computer-controlled machinery. For example, some welding jobs in car factories can now be done with robots. The quality of the work is better. The robots can be run for 24 hours a day with only a limited amount of time down for maintenance. They don't go on strike or need to be paid (although the initial cost of the robots is considerable).
- Robots need specialist skills. Someone has to design them, build them, maintain them, program and re-program them. This has meant new opportunities for workers willing to be trained in newer technologies.

- Email has replaced a lot of communication that was done on paper. Initially, people talked of a 'paperless society' but reality has set in. Many communications, however, are now done electronically. Both internal and external communications are instant, relatively cost-free compared to the post and have reduced the amount of paper consumed, which is good for both the company and the environment.
- With the advance of video conferencing, the reasons for people to be out of the office for days while they travel to meetings have been reduced. This has meant that they can be doing more constructive things.

35.10 Changes in patterns of education

1) People can now do courses online. These might include multiple choice tests that get marked immediately or video tutorials. This may well help people who are trying to hold down one job but train for another or people who are house-bound for any reason.
2) Computer people are expected to constantly update their skills. Developing the necessary skills to help oneself and sort out one's own problems is extremely important in education now and the process starts early on in your school career. These skills are really focused on during post-16 education because much of the support you will have been used to disappears at University. When you start work, you may be provided with opportunities for training but much of your free time may be taken up in you keeping yourself up-to-date!!

35.11 Privacy and confidentiality

Information held on computer can now very quickly be copied and distributed or sent via the Internet within seconds to anywhere in the world. We all expect our details to be kept private by organisations that hold our details. Organisations have both a moral duty but also a legal obligation to keep our details safe - the Data Protection Act 1998 seeks to ensure this. An organisation can take a number of practical steps to keep information private and confidential:

1) It can ensure that a named person is responsible for ensuring that the organisation's DPA policy is enforced efficiently. This would ensure that employees are very clear about their responsibilities.
2) The Data Protection Act should be closely followed. This means, for example, that data should be deleted when it isn't needed anymore and shouldn't be sent to countries that don’t have legislation comparable to the DPA 1998.
3) The organisation should ensure that access to the hardware that holds the data is restricted. This could be done by ensuring the hardware is in locked, secure rooms that can only be accessed by authorised users.
4) The organisation could ensure that data files are password-protected, to ensure that unauthorised people who gain access to the files can't open them.
5) Data could be encrypted using a software encryption tool such as PGP (Pretty Good Privacy). This means that even if the data is accessed or intercepted whilst being emailed, it can't actually be read.
6) The organisation can ensure that the backup policy in the organisation is being followed and that the backup copies of data are themselves held securely and in encrypted form.

35.12 Current legislation

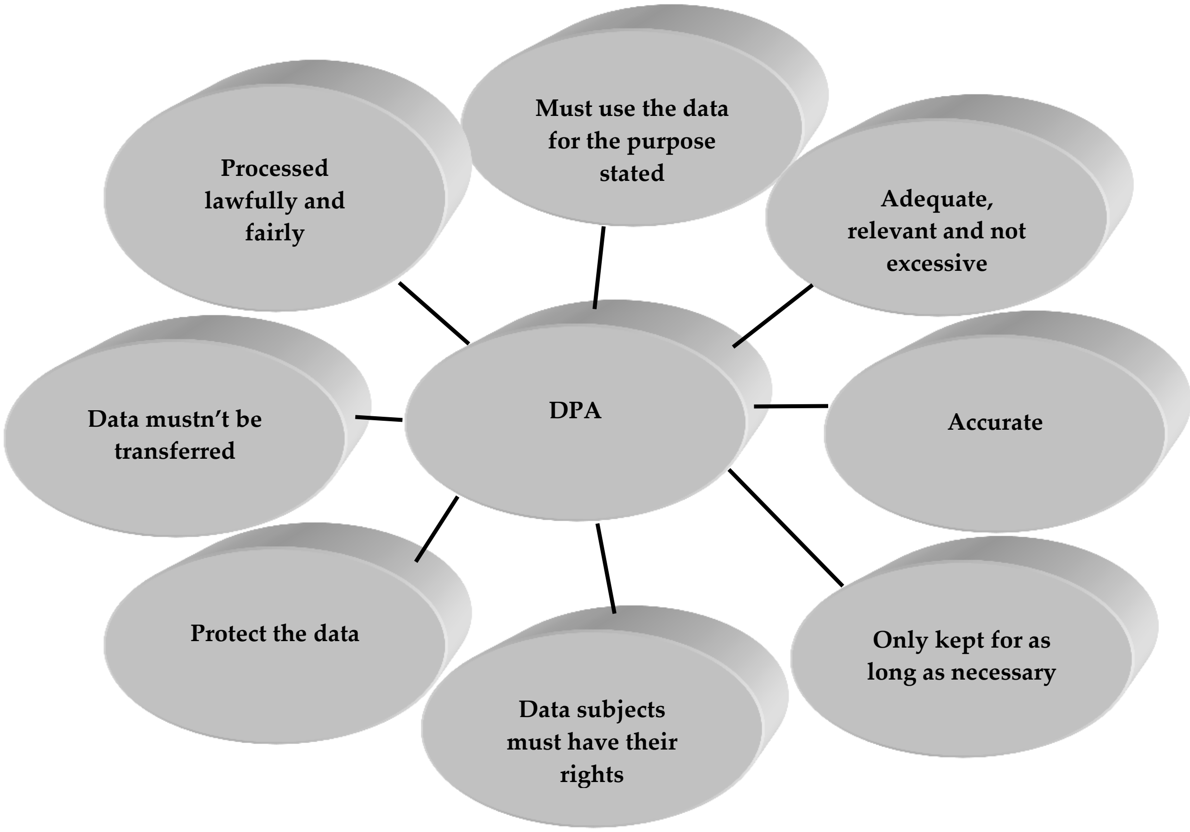
As computers have become more widespread, so the need for legislation has grown. There now exists legislation that seeks to protect our health and safety while working with computers, to protect our privacy, to ensure that those who seek to carry out criminal acts using computer technology are punished and to ensure that intellectual rights to material are protected. One major problem with any country's legislation, however, is that it is difficult to enforce those laws if the 'crime' is carried out in another country. The Internet is a worldwide phenomenon that crosses the boundary of every country. What is illegal in one country may be perfectly legal in another country, or may simply be impossible to enforce. There are lots of good sources that deal with this issue on the Internet. Search Google using keywords like **privacy**, **legislation**, **Data Protection Act**, **European privacy legislation**, **SPAM**, **junk mail**, **cookies** and so on.

35.13 Data Protection Act 1998

Organisations collect data and store information about individuals. It is important, however, for each organisation to recognise that the information collected about an individual is private and that the individual has a right to expect that it stays private. Each organisation should only collect the information that it actually needs and should be up-front about what it needs it for. The DPA 1998 was an extension of the DPA 1984. It bought the 1984 Act up-to-date with European legislation and also included extra safeguards, such as including data sent over the Internet.

When an organisation wants to keep data about individuals, it must register with the Data Protection Commissioner. They have to fill in a form that gives details of their organisation, says what data they want to collect, says what they want to do with it and says who will have access to the data. The organisation must then use the data in the way they said they would. If it doesn't register, or uses the data in a way that it hasn't declared, then it may be subject to legal sanctions.

The DPA 1998 lays down eight principles of good practise, supported legally, which organisations must follow.



8 principles of the DPA.

1) **1. "Personal data shall be processed fairly and lawfully".** This means that a company must be up-front about collecting personal data. It must seek permission from individuals to collect and process their personal details before they actually do it.

2) **"Personal data shall be obtained only for one or more specified and lawful purposes".** In other words, an organisation has to use the data they collect in the way that they said they would use the data when they registered with the Commissioner.

3) **"Personal data shall be adequate, relevant and not excessive".** We have already said that an organisation must declare to the Commissioner that it intends to collect data for one or more reasons. It must then collect only the information it actually needs and not collect any data that it doesn't really need.

4) **"Personal data shall be accurate and, where necessary, kept up-to-date".** An organisation must make attempts to ensure the information is accurate and up-to-date. For example, a school may, once a year, print off the personal details it holds about you, send them home and get someone to check, sign and return it. Any data can then be changed as necessary.

5) **"Personal data ... shall not be kept for longer than is necessary".** Companies must remove data if they do not need it any more. They should have a procedure in place to ensure that data kept on file is regularly reviewed.

6) **"Personal data shall be processed in accordance with the rights of data subjects".** An organisation must have in place a procedure to allow anyone who has data kept about them to see that data. This usually means having a form available so that any individual can request to see their data in writing. There is sometimes a small fee payable as well. The organisation must then provide the data within a fixed time.

7) **"Appropriate technical and organisational measures shall be taken against unauthorised or unlawful processing of personal data and against accidental loss".** An organisation must take practical steps to ensure the data is safe and secure. These can include restricting access to files using password protection and encryption,

restricting access to the hardware that can access files and having a procedure to backup files daily and storing the backups in a fire safe or securely off-site.

8) **"Personal data shall not be transferred to a country or territory outside the European Economic Area, unless that country or territory ensures an adequate level of protection".** In other words, data cannot be sent or accessed from another country outside of the EEA unless they have similar legislation to the DPA. If you have a web site that holds personal details that can be accessed by someone from another country, then this law applies to you!

35.13.1 Some disadvantages of the DPA

While most people would agree that the legislation is useful, there are some drawbacks. Some people would argue that while it sounds good in practice, it is very difficult to enforce. For example, if you are running a small club and store members' details on the computer, you are supposed to be registered - but how many are? The DPA legislation means extra administration and expense for an organisation. For example, somebody has to be responsible and take the time to ensure that data is kept accurate and up-to-date. Somebody has to administer the system that allows somebody to see their details. Somebody has to be responsible for making sure a company follows the DPA. Whenever somebody has to do something, it involves that person being away from the core activities of the organisation and involves an expense for the company. Some might argue that the last principle of the Act described above is impossible and impractical to enforce. How can you monitor who accesses data from an online database via a web site from another country? How can you enforce regulations? Conviction rates are low.

35.14 Directive 95/46/EC (2000) and directive 97/66/EC

We have already said that there is a problem when one country tries to impose its values and laws on people in other countries. One approach is to have legislation that crosses national boundaries. Directive 95/46/EC is European legislation that lays down rules designed to protect the rights and privacy of individuals with regard to data kept about them across Europe. Directive 97/66/EC is another piece of legislation concerned with provisions for data privacy and protection in the telecom industry.

35.15 Privacy and Electronic Communication Regulations (PECR) Act 2011

The E.U.'s "e-Privacy" Directive from 2002, was amended in 2009. All E.U. member states had to bring the Directive into their own law by 2011. The U.K.'s amended Privacy and Electronic Communication Regulations (PECR) Act 2011 became law on May 26, 2011. The law stated, amongst other things, that companies operating in the E.U. and the U.K. must get the consent from its website users for cookies. Cookies are small programs that get installed on your computer when you visit a particular website. They allow websites to offer a more personalised experience, such as remembering a user's preferences and they allow a website owner to track how often their pages are being visited. Cookies can, however, also be used to track a user's online behaviour and other 'interesting' personal information.

35.16 The Computer Misuse Act 1990

This law came in to being specifically to deal with hackers, people who seek to gain unauthorised access to computer systems. In the early days of computing, hackers were seen as 'a little bit naughty'. It soon became apparent that they can cause untold damage to national security, can cause havoc with a company's legitimate operations and can steal a lot of money! Existing legislation was difficult to apply to hackers. The result was the Computer Misuse Act 1990. The act does three things.

1) It makes it illegal to access data to which you have no right. For example, you are not allowed to try to guess or find out a friend's password and then gain access to their files! You can be fined and jailed for up to 6 months.
2) It makes a more serious offence of accessing data to which you have no right with the intention of carrying out other crimes. For example, if you hacked into a bank's system with the intention of stealing credit card numbers, then you would fall foul of this law. You can be fined and jailed for up to 5 years.
3) It makes it illegal to change any data. If you hack into an area to which you have no right and start deleting files or modifying data then you will be breaking this law. You can be fined and jailed for up to 5 years.

35.17 The Copyright and Patents Act 1988

People who write, paint, compose music, design web pages or invent something, for example, have 'intellectual rights' over what they have done. They own the **copyright**. This means that somebody who wants to use what they have done must get permission first from the copyright owner. The copyright holder can refuse to give permission, give permission freely, give permission but attach some conditions of use or could charge for permission. These rights are enshrined in law in The Copyright and Patents Act 1988. For example, If you find a web site you like, you cannot make copies of the web site. You cannot burn copies of the web site on to CD without permission nor can you use images you found there without permission. Many web sites, photographs and images now incorporate software that 'stamps' the images with the copyright owner's details. If you do a computing project as part of your course, you cannot include work in your project that somebody else has done without properly giving credit to the author. If you do use somebody else's work without giving it due credit then this is known as **plagiarism**. It is both unethical and a breach of copyright. There are many web sites offering projects for sale for both school work and university work. Educational institutions and exam boards have become very wise to these sites and now regularly run software through submitted coursework to look for passages that have been stolen.

35.18 The Management of Health and Safety at Work Regulations 1992 / Electricity at Work Regulations 1989

Computing technology and how it is used has thrown up many questions in relation to health and safety. There have been concerns regarding the radiation emitted by VDUs, the strain placed on eyes by looking for long periods at computer screens, the effects of repetitive movements on bones (Repetitive Strain Injury), the effects of sitting incorrectly at computers and using electrical equipment in busy areas. Organisations have a responsibility to ensure that the environment workers have to carry out their jobs in is a safe one with any risks minimised. Employees also have a duty to ensure that they work safely and in accordance with any training they have been given. Health and safety legislation makes clear the obligations of both employers and employees. These are discussed later in this chapter.

35.19 Other legislation

There are other laws that have an impact on users of computers. These include the Human Rights Act 1998, the European Convention on Human Rights, the Freedom of Information Act 2000, the Anti-terrorism and the Crime and Security Act 2001.

35.20 Accessing controversial information via the Internet

One country's laws and values are not necessarily another country's laws and values. If one country decides that hard-core pornography is perfectly legal to show and sell and their citizens put web sites on the Internet, how can another country like the UK stop people viewing this kind of material, even though it is against the law in the UK? The answer is that it can't. Nobody owns or runs the Internet so it is very difficult for anyone to have control over it. We live in a democracy and we expect freedom of expression and to a large degree freedom of information. Most citizens, however, accept that there are times when there is a 'national security' argument for having some information restricted. Before the Internet, each country could decide exactly what their nationals could have access to. Post Internet, however, the situation has completely changed. It is very easy to set up anonymous web sites that have all kinds of controversial material on including pornography, how to make a nuclear bomb and libellous gossip. This information crosses every boundary. It is difficult to convict anyone of anything anymore!

35.21 Health and Safety

The widespread use of computers in society has brought along with it a wide range of health and safety issues. There is legislation in place to ensure that both employers and employees do their jobs in a safe working environment. In this section, we will outline the problems and the practical steps that can be taken to reduce the problems.

35.21.1 Repetitive Strain Injury (RSI)

This is a condition that can affect anyone who has to use the same parts of their body to make the same kind of movement over and over again. It can affect musicians, secretaries, checkout assistants and computer users, for example. RSI builds up over time and can affect the joints of the hands or the back, for example, and the muscles, nerves and tendons around the joints. It is much easier to prevent than cure! Providing workers with regular rest breaks can reduce this problem.

35.21.2 Eye strain

Constantly looking at a bright Visual Display Unit can cause eye problems. Eyes need to move around to stay lubricated - staring at a screen can cause a condition called 'dry-eye'. Looking at bright screens can strain eyes and cause defective vision. The problems can be reduced by ensuring:

- a good screen contrast. Using a screen filter helps.
- that users know that they should look away from the screen every 10 minutes or so.
- users take regular breaks away from the screen.
- employees do not have defective vision. They should be given regular eye tests.

35.21.3 Injuries caused by bad posture

People who work for long periods at a computer can suffer from a range of problems such as backache, wrist strain or a sore neck. These are often caused by a bad-working posture at the computer or a poorly designed workstation. Employers should ensure that employees can sit properly at computer stations and have had training to ensure that they know how to adjust equipment. Employees have a duty to follow the advice they have been given.

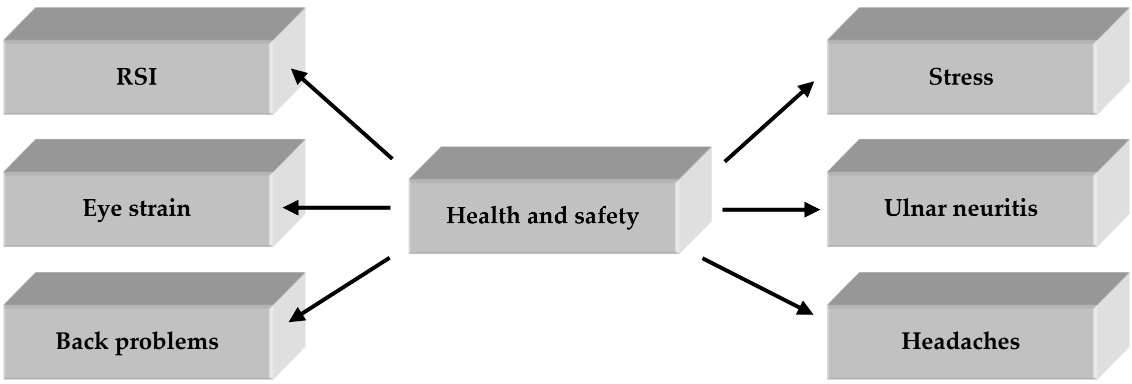
- They should be able to adjust seats so that the lower arms are horizontal and at the same height as the keyboard.
- They should be able to sit with their thighs horizontal with their legs under the table and have footrests.
- They should be able to adjust the height and tilt of screens. Ideally, the eyes should look forward at the screen, not downwards or upwards. The back should be straight and the head should be directly over the spine. If the head is leaning forward or backwards, it puts pressure on the bones of the spine.
- They should be able to adjust keyboards and they should be able to use a wrist rest for both the keyboard and the mouse. If wrists are not supported, they can quickly become strained.
- They should be able to retrieve hard copy without over-stretching or bending low down.

35.21.4 Stress
People get stressed out for a range of reasons:

- New software and hardware products, and new updates, are introduced that have to be learnt. For some older people, this may be a problem, especially if they are trying to keep up with younger employees who have grown up with IT.
- Software exists which constantly monitor performance. Data input operators could worry, for example, that they will lose their jobs if they don't go fast enough or make too many mistakes. A strike in July 2003 crippled Heathrow Airport because employees had concerns about the introduction of a new computerised swipe card system.
- Some employees may have few outlets to discuss their problems because some working environments involving computers reduce or remove the opportunity for social interaction.

35.21.5 The working environment
An employee's working environment should be considered. Employers should ask themselves some questions. For example:

- Are there any cables that are trailing around the computer that somebody could catch themselves on?
- Is the ventilation adequate? Computers create heat. This can cause headaches and drowsiness.
- Is the lighting adequate?



35.22 Environmental issues
Computers require energy to build. The components that make up computers use energy in their manufacture. The majority of energy sources contribute in various degrees to polluting the atmosphere and the environment. It has been estimated that to build just one computer requires approximately ten times its weight in chemicals and energy! To make one computer, according to scientists at the UN University in Tokyo, you need about 250 Kg of fossil fuel, over 20 Kg of chemicals, over 1000 different chemicals, including lead, arsenic, cadmium (all toxic) and over 1500 Kg of water.

Around the world, we continue to make computers in huge quantities. Approximately 150 million will be made in the next 12 months alone. If you then multiply that number by the amount of energy, chemicals and water required, it is not difficult to see how computers contribute to pollution. If you also consider that a typical PC would be lucky to be in use for just three years (compared to 10 - 15 years for a car and about 15 for a fridge), you can get an idea of the numbers of PCs involved.

European legislation has been introduced that demands 70% of a PC is recycled when it is finally scrapped. Whilst governments have encouraged recycling, some people have argued that this is the *wrong* approach. It takes a lot of energy, they argue, to recycle computers! It can also in itself be the cause of environmental damage. For example, PCs are being transported to developing countries such as China for recycling. The process to recover metals such as copper, lead and gold from computer hardware involves the use of acid baths – and this is not good for the environment. It is also not good for the people directly involved in the recovery process. This is because health and safety standards are well below those of the developed countries and the appropriate protective equipment is rarely provided. A better plan of attack would be to reduce the number of computers being built in the first place. This could be done in a number of ways.

- You could encourage people to buy second-hand PCs with warranties - not many people know about the existence of companies who buy and refurbish computers.
- You could give tax breaks to companies who buy second-hand PCs, rather than tax breaks to buy new PCs.
- Companies could refurbish PCs and send them to developing countries for use there.
- You could encourage people to upgrade existing parts of a PC rather than upgrading their entire PC.
- You could encourage PC manufacturers to invest in making products last longer.
- You could encourage a paperless society. A lot of hard copy is still produced. The more paper used, the more trees have to be cut down. Even if these are from renewable sources, turning trees into paper involves energy as well as involves making parts of forests an eyesore.

Q1. What is 'SPAM'?
Q2. What is meant by the digital divide? Give two clear examples that illustrate a digital divide.
Q3. Define 'copyright'.
Q4. Why is it difficult to legislate and control pornography in the UK using the Internet?
Q5. Describe two disadvantages of being a teleworker.
Q6. State any four principles of the Data Protection Act 1998.
Q7. Describe ways to reduce the potential for eye strain when using computers.
Q8. Describe examples that illustrate how computers can raise stress levels in some groups of people.
Q9. Do some research on the Internet. Why do some people object to the recycling of computers in the third world?
Q10. Suggest ways to extend the life of a PC.