UNLOCKING
CONSCIOUSNESS

BRIAN MIND FORUM

Appendix 021

History of Software

Arguably the first digital computer was Colossus, which was used at Bletchley Park to break enemy cryptography during the Second World War. Colossus could only do the one task it was designed to perform. In 1948 researchers in Princeton, Manchester and Cambridge began to make the first 'stored program computers' work. The key was that information to be processed was stored in the same way on the same memory 'media' as the list of instructions or program. The valves and circuits was called the *hardware,* and to differentiate the list of instructions, the program was called the *software.*

A few years later Crick and Watson identified Deoxyribo Nucleic Acid (DNA), which appeared to be a massive 'program'; that contained a list of instructions that could build a living being. Very little research was carried out in cognitive neuroscience until in the 1980s and nineties microcomputers found their way onto the desks of scientists. Science was very compartmentalised and so many games programs were being written by teenagers with no academic credentials, 'programmers' tended to be ignored. However, the evidence became overwhelming that the mass of neural networks in the '*brain'* was very like the '*hardware'* in computers, and the electrochemical signal patterns that flowed over these networks was virtually identical to computer programs. The computing profession was very pleased with itself that it had identified this '*software'*, or *'mind'.* Descartes's dualism was right after all.

Has maths been invented or discovered?

Down the centuries, mathematicians have agonised over whether they have invented maths or discovered it. The computing profession was quite clear it had invented software. However, over recent years, as many disciplines are converging, we have begun to realise that software is a crucial component of almost all systems that we have discovered and are learning about. The body has two hardware transmission systems; the neural networks of the central nervous system, which incorporates the brain, and the arteries and veins of the cardiovascular system. The software of the former is a mass of electrochemical signal patterns generating a series of electromagnetic fields and waves. The software of the latter is the blood stream that transports nutrients and waste around the body but also the mass of hormones and neurotransmitters of the endocrine system – in their way just as detailed, complex and sophisticated as the signal patterns.

The jury is out on whether the immune system has its own independent network, or uses the central nervous system to transmit its 'software' signals and the cardiovascular to transport its 'hardware' cells.

The chromosomes of the genome incorporate both hardware components that in due course build all the protein structures and the software instructions that select what occurs when, and in which sequence, and in due course coverts standard components into specialist organs. Wherever we look this duopoly is evident.

There is a strong argument to begin to think in terms of a new Science of Software.

The essence of the power of software is that the same hardware can be programmed to do a multitude of different software tasks. This means that one piece of physical equipment (hardware) can effectively carry an infinite number of tasks. This also means that the capacity of the brain is unlimited.

First examples.

One early example is a 17th century striking clock. One mechanism is in place to use a pendulum to regulate the pressure of a spring or series of weights to cause a minute hand on the clock face to rotate once an hour, and the hour hand once, or twice a day. Each time the minute hand passes the vertical it releases a second mechanism which uses the pressure of a second spring or weight to strike a bell and then reset itself. Add in a snail wheel – whose circumference starts with a radius of one out to a radius of 12. Now when the minute hand releases the second mechanism it moves against radius one and rings the bell once, then resets itself moving the snail wheel up one twelfth of a revolution.

The next time this mechanism is released it moves against radius two and rings twice as it resets at three. After twelve it resets itself moving the snail wheel to one. The exact same mechanism does twelve different tasks every day.

Jacquard Looms

In the 18th century there grew up a thriving weaving industry in Lyon. A weaver by name Jacquard hit upon the ingenious invention of controlling the selection and movement of the bobbins carrying the various different coloured threads on his looms, from a series of holes in a pack of cards. The cards replicated the activities of a weaver, but the cards worked much faster, did not make errors, and moved at a constant speed weaving a better quality cloth. The biggest trick was that the cards could control the pattern. Different sets of cards made different patterned cloth. Thanks to these cards, or 'software' any number of designs of cloth could be made from one standard loom, or 'hardware'.

The age of automation had arrived.

It is worth noting the economic effect. One weaver could control ten looms. Setting up was much less skilled than weaving, so the economic value of weavers collapsed. A few bright designers could 'program' their designs into cards, which then could produce an unlimited amount of cloth so their economic value soared. Jacquard could produce his cloth at a much lower cost and sell a better product at a lower price than his competitors, so his business boomed. There are no records of riots in Lyon.

Steam Organs.

The same principle was adapted for the steam organs, which can still be seen at funfairs – even in the 21st century! A belt of cards directs the pumped air to the various organ pipes in the sequence required and for the correct length of time to play virtually any tune.

The industrial revolution built on this principal and there were many riots as machines mass produced products putting many skilled people out of work, but at the same time creating new types of employment with different skills. Providing the whole economy was growing fast the Luddite threat was minimised. Of all the economists, especially those practising today have not taken on board the implications if the economy is not growing fast and steadily. Karl Marx may have got everything else wrong, but he was right when he forecast that "the workers become poorer the more wealth they produce". Over the last thirty or forty years many descendants of Jacquard's weavers have seen little or no rise in their incomes. The Jacquard descendants are increasingly extremely wealthy. There are not enough replacement programmer type jobs to balance this loss of purchasing power, and the populations of the western world, at least, have mostly acquired the majority of their home comforts. The dispossessed largely cannot see the vehicle of their destruction, and look for other scapegoats. Back to Software.

Babbage Difference Engines

Charles Babbage designed his difference engines, which were like very sophisticated clocks. Processing was carried out by cog wheels. Change the position of some of the control disks – prototype software, and these fixed engines - prototype hardware, carried out different calculations. Ada, Countess of Lovelace and daughter of Lord Byron, understood the potential of software but the Babbage difference engines were never built. Only in the late twentieth century was the expertise available, and a replica Mark 1 Difference engine some twenty yards long and ten feet tall, was built as a working exhibit of what might have been. It sits in the Science Museum in South Kensington and is a source of puzzlement to many people, not just school students that the iPhone in their pockets is many million times more powerful, but that, without that early thinking, perhaps their pockets might be empty.

In the late nineteenth century Hollerith started to build calculators that were driven by Jacquard like cards. Hollerith's punched cards just encoded numbers. Using pins to recognise holes and turn (clock like) wheels, Hollerith's machines could add, subtract and sort large volumes of information quickly and accurately. By the Second World War, photoelectric sensors were replacing pins and larger cards with space to encode the alphabet could be used both as input and output systems and also memory systems. They could carry out an increasingly wide range of different applications on one machine.

The first generation of '*stored program*' computers.

The stored program computer built on these concepts, but replaced cards and reels of paper tape with directly connected keyboards, and in due course speech recognition software; and a plethora of memory storage media.

Early computers required the programmer to control the input and output devices for each application, but as these became standard on each computer, subroutines were built to ease the programmer's task. For the first twenty years or so programming tended to be the junior partner to the hardware designers.

The Micro

The arrival of a computer small enough to sit on an office or home desk, with a keyboard and small printer, plus a screen to see what was happening, proved an instant success. In parallel the development of 'floppy disks' that could read and write data, could also hold programs. This was a major breakthrough. Almost all 'conventional' 'main frame' computer memory systems held data files in sequence usually on long reels to magnetic tape. To process one item on one file required processing the entire database.

Random Access memory

Floppy disks' and all their successors had a number of rewrite heads capable of spanning a number of magnetic 'storage' rings. Four or five companies competed to provide a basic 'operating system' program, which allocated space on the floppy disk systems, controlled the keyboard, organised the information on the screen and drove the printers. Program designers could just concentrate for the first time on the application, while the 'operating system' did the donkey work.

Within two years the computing industry was stood on its head. The standard interface both hardware and software suppliers became the 'Disk Operating System' or DOS. All hardware equipment had to operate to Dos Standards. All software programs had to be DOS Compliant. One of the great monopolies was born. This standardised the whole industry and hardware, but particularly software design leapt forward. Microsoft DOS grabbed this monopoly and became the richest company on the planet.

The Internet

All computing hardware systems have random access memory storage systems. Almost all systems operate 'on-line in real time'.

It would hardly be possible to send and receive emails if the sender and receiver had to look up every aspect of a message like the address and any attachments from one long sequential file of all the data available. E-mails could only be received and transmitted as part of a 'batch' every now and again. On-line shopping, banking, reservations, and so much more of today's familiar computing would not have happened, would not be possible..

Economics

By the 1990's the roles had been reversed and hardware, once the dominant partner was soon almost side-lined. True, today, people buy their favourite hardware but it is the software that calls the shots. The economics has changed. Replicating software is effectively free. Manufacturing hardware still costs money but all computers are now built automatically and have to be built in huge numbers to achieve the necessary economies of scale. Companies like Google cannot make money out of their basic service: a search engine. Brilliantly Google worked out how to make money out of advertising. Google knows a remarkable amount about all the people that use their search engine, and that information is valuable to advertisers. We pay for our searches, but painlessly because advertisers pay to receive our data. We use our data as currency.

Look at the Babbage's Difference Engine in the Science museum. That iPhone of yours will be in the You will hear someone say in wonderment!