2 June 2013

Smart Cities

Session I: Lecture 2:
Turing's Legacy

Michael Batty

m.batty@ucl.ac.uk
@jmichaelbatty

http://www.spatialcomplexcity.info/
http://www.casa.ucl.ac.uk/

References so far ……

Andrew Blum (2012) **Tubes: A Journey to the Center of the Internet**, Ecco, New York.

George Dyson (2012) **Turing's Cathedral: The Origins of the Digital Universe**, Vintage, New York.

Dava Sobel (1997) **Longitude: The True Story of a Lone Genius Who Solved the Greatest Scientific Problem of His Time**, Walker and Company, reprint, New York

Tom Standage (2007) **The Victorian Internet: The Remarkable Story of the Telegraph and the Nineteenth Century's On-line Pioneers**, Walker and Company, reprint, New York.

How did it all begin – we will discuss the background to the development of the digital computer and then move to talk about how computing has developed in the last 60 years. But first we will say something about the motive force for computation and it technology and this involves us saying something about electricity, then logic. This lecture which is our second we will organize in the following topics.

- The Motive Force of Computation: Electricity
- The Representation of Numbers: The Binary System
- The Idea of an Algorithm: Turing
- Miniaturisation of Electric Circuits: Moore's Law
- Communications and Computation: Metcalfe's Law
- Hardware to Software to Dataware to Orgware, and then to to the modern day … next time

The Motive Force of Computation: Electricity

Electricity – basic feature of all material. It is so widely part of nature that we simply take it for granted without knowing much about. It has a long history: the Greeks knew about it – electric fish – sparks – material has a charge – and when you bring materials into contact interesting things happen ….. when charges are in contact

Basic ideas of electricity began to be explored in early 1800s – We have seen the work of Michael Faraday at the Royal Institution but there were many developments in the 19th century – electricity came to be regarded as a force – a very powerful force, much more powerful than gravity and in some senses, electricity is the force that powers the modern world.

Force and waves – James Clerk Maxwell tied it all together mathematically– in fact we noted the place where Maxwell did some of his work at Kings College London in the 1860s. I think he also did quite a bit of it in Scotland and he was a little later than Michael Faraday. He went on to found the Cavendish Lab in Cambridge after his spell at Kings

But we must say that two others things were developed during the 19th century – first the telegraph by many people but particularly Morse of Morse code fame– a kind of early binary transmission system, and then the telephone. But both these were, like most of the internet based on wires not wireless – there is a very nice book by Tom Standage called **The Victorian Internet** worth reading about telegraph

So electric circuitry produced the power to drive this type of communication. It is basic.

The Representation of Numbers: The Binary System

Ok the basic nuts and bolts for moving information were established by the early 20th century – but what was to be moved – can we move anything – how do we convert data into a form that could be universal.

Here the binary number systems comes in – basically from classical times people had speculated that we could reduce numbers to simple pulses. In fact Francis Bacon in the 16th century laid out a kind of binary and implied that all numbers could be so transformed into 'yes' and 'no', 'on' and 'off' – but it was not until the 1920s that the binary number system was formalised and linked to electric circuits. This drew on George Boole's algebra which he developed in the mid-19th century.

Essentially if you can reduce all numbers to 0 and 1 and combinations therein and thereof, you can use switching to transmit them. The basic logic of this transmission was first demonstrated by Claude Shannon at MIT in his Masters thesis in 1937 and became the basis for numerical computation during the war years when the first large scale electronic – digital computers were built.

Before then in the 19th century, the first analog computers were built where switching was done mechanically as in Charles Babbage's difference engine and many similar workable systems were developed at places like MIT in the 1930s and 1940s. The German code breaking machine Enigma was also based on such analog devices in the UK and our next computer pioneer Alan Turing was associated with this – there is a movie starring Kate Winslet about this if you are interested

I can’t give you a course in binary here but the essence of the idea is interesting and here is a table of our the first 10 decimal numbers are converted.

http://en.wikipedia.org/wiki/Binary_number

0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10 – (A)	1010

You can find out all about this on Wikipedia – how to use arithmetic on them. Essentially if you can reduce all numbers to 0 and 1 and combinations therein and thereof, you can use switching to transmit them.

The Idea of an Algorithm: Turing

The next step in the evolution of ICT which proceeded in parallel to developments in electricity/communications and number representation was even more profound.

This is the idea of the universal machine – using binary type system but more in terms of the development of a procedure for computation. Alan Turing showed in 1936 I think how an abstract machine – a Turing machine – could be built that could compute forever. Thereby he showed that computation was universal using these ideas in that it could in principle compute the computations necessary for a digital computer – i.e. itself. This idea of an algorithm is hard to define really, so I am going to quote from the Wiki entry ….

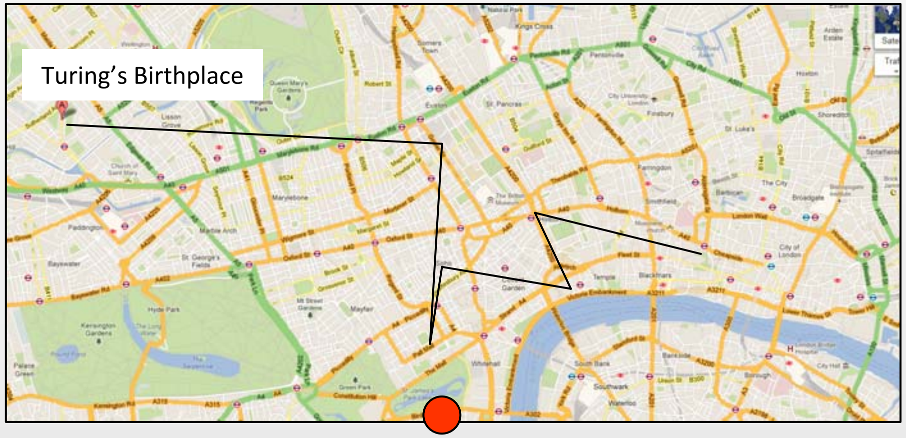
“He proved that some such machine would be capable of performing any conceivable mathematical computation if it were representable as an algorithm.

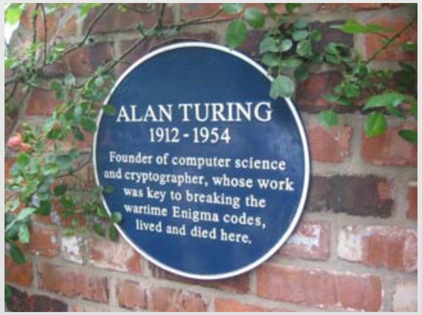
He went on to prove that there was no solution to (David Hilbert’s) *Entscheidungsproblem* by first showing that the halting problem for Turing machines is undecidable: in general, it is not possible to decide algorithmically whether a given Turing machine will ever halt.

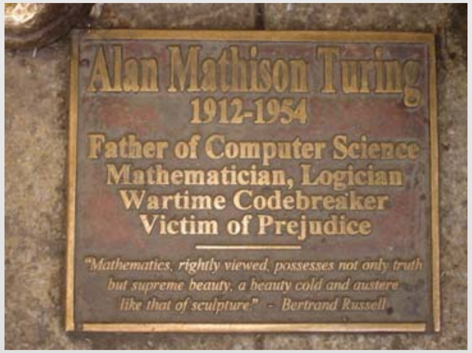
Although Turing's proof was published shortly after Alonzo Church's equivalent proof using his lambda calculus, Turing had been unaware of Church's work.[25] Turing's approach is considerably more accessible and intuitive than Church's. It was also novel in its notion of a 'Universal Machine' (now known as a Universal Turing machine), with the idea that such a machine could perform the tasks of any other machine, or in other words, is provably capable of computing anything that is computable. Von Neumann acknowledged that the central concept of the modern computer was due to this paper.[26] Turing machines are to this day a central object of study in theory of computation.”

Turing died in early life in tragic circumstances in Manchester where he was then deputy director of the computer lab in 1954 ….

Here too could feature on our Walk Through the Smart City because he was born not so far away to the north and east of central London







Interesting to note in passing that both Turing and Shannon are known generally not for their major work – in fact Turing more for Enigma and Shannon more for information theory.

There are lots of online sources where you can explore the history of computing and code and networks such as the BBC web site

2001 Movie

Miniaturisation of Electric Circuits: Moore's Law

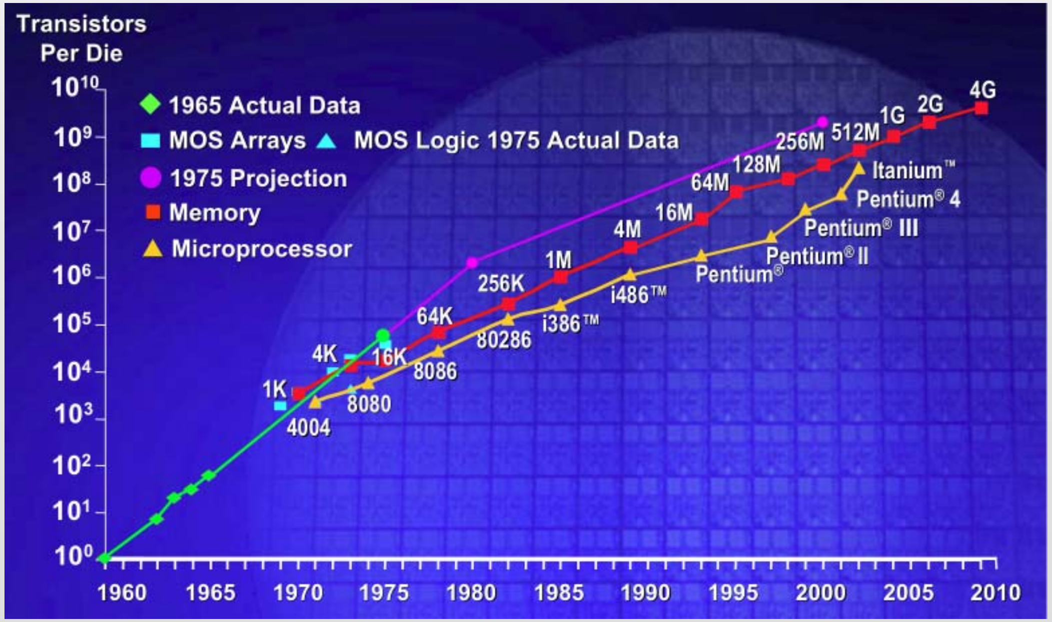
The fourth element in all this which is arguably the most important in that it made computation all pervasive was the invention of circuitry that relied on new materials – silicon

At Bells Labs in 1947, the transistor was invented by William Shockley, John Bardeen and Walter Brattain. This sets us on the road to miniaturisation. It was more about materials than numbers, about silicon as it happens

And this led to the story of Silicon Valley. Shockley went back home to the Valley, founded his own company that led to Fairchild Semiconductor that led to Intel – the rest is history as they say

In 1959, the integrated circuit was developed and then in 1971, the breakthrough – the microprocessor – all the circuitry needed for a computer on a single chip.

Gordon Moore one of the founders of Intel coined his famous law that ***the amount of circuitry/memory on the same sized chip was doubling every 18 months*** and has been ever since.



Communications and Computation: Metcalfe's Law

At the same time, there were rapid developments in networking and in the material of networking. In parallel to miniaturisation, fibre optics was invented.

Computers had begun to be networked from the very beginning but there ARPANET was founded to linked main frames across distance in 1969 and in the 1970s at Xerox Parc local area networking was invented in terms of the Ethernet.

There is much to say about all this but here it is worth noting that as computers get connected together, then they share resources and these increases as the rate of the number of connections of the network. This led to Metcalfe's Law – the ***value of a telecommunications network is proportional to the square of the number of connected users of the system (n^2).***

There is also Gilder’s Law and Sarnoff’s Law and so on. Let me state quickly five of these laws but there are more that relate to miniaturisation, memory, cost, value and so on of ICT.

Moore's Law: formulated by Gordon Moore of Intel in the early 70's - the processing power of a microchip doubles every 18 months; corollary, computers become faster and the price of a given level of computing power halves every 18 months.

Gilder's Law: proposed by George Gilder, prolific author and prophet of the new technology age in 1988 - the total bandwidth of communication systems triples every twelve months. New developments seem to confirm that bandwidth availability will continue to expand at a rate that supports Gilder's Law.

Metcalfe's Law: attributed to Robert Metcalfe, originator of Ethernet and founder of 3COM in the 1970s: the value of a network is proportional to the square of the number of nodes;

Sarnoff’s Law: states that the value of a broadcast network is directly proportional to the number of viewers, and believe it or not

Zuckerberg’s Law: the information that people share doubles each year

Hardware to Software to Dataware to Orgware to ….

Ok there are many many things we can say about modern computation and to finish this history so far, let me give you a rapid run down of how computing has changed through this process of miniaturisation and networking and

The 1950s and 1960s: ***The mainframe era*** – essentially users delivered by hand their programs to some central large computer which were operated behind closed doors. Programming and operation were completely separate. There was not such thing as software; the term is very hard to source – thinking versus machine operation has something to do with it but it gradually emerged I think in the 1970s and came in with a vengeance once the PC took off.

The 1970s: ***The minicomputer era*** – essentially users has their own small mainframes which were often networked with teletypes or even visual display units in star like fashion around the machine. There was some sense too that users needed to know more about the operating system than before as they had more control.

The 1980s: ***The PC revolution***: with the invention of the microprocessor, small computers became the norm but the end of this decade and the laptop or portable PC came onto the scene. Networks of machines appeared and by the 1990s one was connecting PCs to networks to act as go betweens between data and users and big computation on i and mainframes

The 1990s: ***extensive networking***, continued miniaturisation, the web and the emergence of hand held computers

The 2000s: ***hand held devices – phones, tablets and related devices*** – software and data on such devices and also continual computation

The 2010s+: ***Apps, many different devices***: some specialisation is beginning with respect to computers for different roles

At the same time as computers have become smaller and faster and bigger in terms of memory, special purpose machines continue to be devised – supercomputers, parallel computation on arrays of devices, grid computing

The client-server architecture is now writ large. But in this value chain the value of hardware has fallen – this is Moore's Law in action. What has happened is that the value of Software has risen but it now appears that software is becoming so general that it is different combinations of software that are being devised. Hard to know if software is as important as it was.

Now if we look at other elements of computing, data and organisation, which we might refer to as dataware and orgware, then these are becoming more important in the value chain – how much more so is tricky to figure but organisation is absolutely key to good operation and integrated data too is key – orgware and dataware may well be the keys to the smart city, rather than hardware and software

One last issue before we more to the next session is that graphics and interaction are all important too – and that right from the beginning, there was graphics of a kind. But not until the PC was graphics utilised in terms of the interface and only when memory was sufficient – Xerox Parc, then Apple led the way but so did the workstation companies – individual minicomputers. Graphics is key as we will see.

You can get the lecture powerpoints
from the web site
http://www.spatialcomplexity.info/

There are more resources at
http://blogs.casa.ucl.ac.uk/
http://www.casa.ucl.ac.uk/

No references as yet apart from the basic review paper
These will be gradually placed on the blog.

Thanks, Any Questions?