The social life of networks*

An Inaugural Lecture by John Naughton[1]

This lecture series was created to honour John Beishon, the founding Professor of Systems at the Open University and my first Head of Department. He was, by turns -- and sometimes simultaneously -- a friend, a mentor, a scourge and an inspiration. From him I learned a great deal about innovation and subversion in organisations – especially in this one. And in his later career – long after he had left the Open University and been appointed to rescue a failing Polytechnic from an inferno of ideological intolerance – I saw him display the kind of personal courage that is quite alien to most academics. Intellectuals are good at many things, but in general moral fortitude isn't one of them.

But then John wasn't your average intellectual. He was ferociously bright and resourceful, but not what you'd call cultivated. Aristotle, Plato and Spinoza had passed him by. He first trained as a metallurgist, and to his dying day displayed an intense interest in defective or careless welding. He then switched to psychology, for reasons that I never understood. After a D.Phil in applied psychology at Oxford, he made the transition into that strange oxymoronic subject, 'management science', becoming a Reader at Sussex University in the late 1960s. He was therefore already on an interdisciplinary trajectory when he was appointed Professor of Systems here in 1970.

From the moment of his arrival in the Faculty of Technology, two of his most prominent characteristics became evident. The first was his profound belief that the wrong people were in charge – of the Faculty, of the University, of the country and indeed of the world in general. This was partly a product of his ideological heritage. He had been, in his youth, a Communist sympathiser if not actually a Party member. But it was also partly a product of his anarchic temperament. He was by nature a trouble-maker in the best sense, though of course those set in authority above him did not always see it that way.

---

[1] Professor of the Public Understanding of Technology at the Open University.

No sooner had he arrived in the faculty of Technology, for example, than he declared that the proposed pace of academic development envisaged by the Dean and other senior colleagues was far too leisurely. He blithely announced that the Systems Group would produce four courses in its first four years – a productivity target roughly four times greater than wiser counsels had believed possible. He then recruited a team of young academics – of whom I was one – and delivered on that rash promise. And of course in some quarters he was never forgiven for that. Nothing quite irritates academic colleagues as much as success.

The first course we created was called Systems Behaviour. It was designed to give students an appreciation of the power of systemic insight by looking at a number of complex real-world systems. One of the case studies chosen for examination was the telephone network, and therein lies my first tale.

Remember that the year is 1972. The British telephone network was a state monopoly run by a Stalinist outfit called Post Office Telecommunications, a division of the nationalised Post Office which had been created from the old General Post Office in 1969. Let us call this sinister organisation POT. (Later on -- in 1981 -- POT metamorphosed into British Telecom or BT, a state-owned corporation independent of the Post Office.) Like most state corporations, POT was supply- rather than demand-driven. It *had* customers, of course, but their needs were regarded as subservient to those of the system. They could not, for example, choose their own telephone handsets: only those approved by Post Office Telecommunications could be connected to the system. And you couldn't purchase phones – you could only rent them from POT. White ones, I seem to recall, cost more. If you wanted a new phone line, then you filled in a form and waited until the company deigned to install it. It could take weeks, sometimes months. The same applied if you wanted a new extension in your house, or a change to your PABX – your company switchboard: you had to wait upon the Post Office's pleasure.

This mindset of total control was hard-wired into the organisation's corporate DNA. And it had sinister as well as comical aspects. For example, there was an intimate relationship between Post Office Telecommunications and the security services. Every employee had to sign the Official Secrets Act upon taking up employment. Why? Because when MI5 or MI6 or Special Branch wanted to tap someone's phone, it had to be done by a POT employee physically installing the tap.

This then was the organisational context of the system that John Beishon decided OU students should study. It would have been relatively easy to have constructed an 'official' explanation of how the telephone network operated, possibly even with the assistance of its owner. But John had the profound insight that if you really want to understand a system then you should look for ways in which it can fail. And at that time, a group of technological anarchists called Phone Phreaks – that's phreaks with a ph – had begun to explore the weaknesses in the network. They had discovered, for example, that using an oscillator to emit a tone of a particular frequency into a handset could give you access to certain kinds of system management facilities – it could, for example, enable you to make international calls for free. These people were driving the telephone authorities wild, for good reasons and bad. And chief among them was a young technical journalist named Duncan Campbell.

It will not surprise you to learn, therefore, that when John Beishon went looking for a consultant to help with the creation of our course module about telephony, it was on Campbell that his gaze alighted. And you can imagine the response. POT was first incredulous, then furious, then incandescent with corporate rage. Threatening noises were made to the University. Questions were raised about John's suitability for an academic post, about his judgement, ideological background and beliefs. But he – and the University – stood firm, and the module -- created by Peter Zorkoczy, a colleague from the Faculty's Electronics Group -- proved popular and academically successful with students.

But that wasn't the end of the story. At this point the other side of John's character came into play. He was, as I said, a metallurgist by background, and intensely practical by nature. He was good with his hands, and loved dismantling and repairing things. The Systems Group at that time was housed in a temporary building, now long demolished, and was growing like crazy as more academics and support staff were recruited. It wasn't long, therefore, before the original arrangement of phones and extensions became dysfunctional. We needed a new topology for our departmental phone system.

Accordingly, a request was made to the telephone authorities for the necessary alterations. And back came the response – after the statutory interval – that its engineers would be available to do the work at their convenience. Needless to say, this infuriated John, so one weekend he came in on Saturday and rewired our phone system.

Again – knowing what you *now* know about Post Office Telecommunications – you can imagine the outcome. Departmental legend has it that when the company discovered the crime, it threatened to terminate telephone service not just to the Technology Faculty but *to the entire university.* I don't know how the matter was resolved in the end, but I have no doubt that it did little to enhance John's esteem in the eyes of the University's senior management.

These two stories – about the consultancy offered to Duncan Campbell and the unauthorised rewiring of the Systems Group's telephone extensions – are interesting because they reveal something of the personality of the man we honour today. But as it happens they also serve as a good jumping-off point for my main concern, which is the social dimension of communication networks.

-- o0o --

There was a time when telephone networks were probably the most complex machines that engineers had ever built. From our point of view, they had two important characteristics.

The first is that they were tightly controlled.

The second is that they were optimised for a single function – that is to say, voice calls.

Tight control was implemented in two main ways. Firstly, through ownership. Telephone networks were either the property of the state (as in Britain[2]), or of a state-licensed monopoly, as in the US. And ownership brought with it draconian powers. It enabled the owner, for example, to determine who and what could connect to the network, and on what terms; and what they could use the network for.

The most comical illustration of this that I know of comes from the US, there the giant AT&T corporation was the owner of the network. My friend Larry Lessig tells the story in his book, *The Future of Ideas.*[3] What happened was this:

In 1956, a small company in the US made a device called a "Hush-a-Phone." The Hush-a-Phone was a simple piece of plastic that attached to the mouthpiece of a

2 With one minor exception – the city of Hull had been allowed, for quirky historical reasons, to operate its own telephone network.

3 Lawrence Lessig, *The Future of Ideas: The Fate of the Commons in a Connected World,* Random House, 2001.

telephone. It blocked noise in a room so that someone on the other end of the line could better hear what was being said. The device had no connection to the technology of the telephone. It had no electronics in it of any kind. All it did was block noise, much the way you might do by cupping your hand over the phone.

When the Hush-a-Phone was released on the market, AT&T objected. The device was a "foreign attachment", it said. Regulations forbade any foreign attachments without AT&T's permission. AT&T had not given Hush-a-Phone any such permission. The Federal Communications Commission agreed with AT&T, and the Hush-a-Phone was history.

Now of course, as Lessig points out, Hush-a-Phone is an extreme case. The real purpose of the foreign attachments rule was, at least as AT&T saw it, to protect the system from dirty technology. A bad telephone attached to the telephone system could, AT&T warned, bring down the system for a whole region. Telephones were lifelines: at the time, AT&T provided telephone connections for 85 per cent of all US households, and they had to be protected from the experiments of an inquisitive nation. But you get the point: the effect of tight ownership of the network was to *restrict innovation to what the owner would tolerate*. AT&T was able to dictate the pace of technological change.

The second key aspect of telephone networks was that they were designed for a particular application – the making of voice calls. This determined their technology. It made them what we call *circuit-switched* systems.

If I wanted to call my mother in Ireland, I first dialled into my local exchange, which then set up a link to a regional exchange which then set up a link to an international switching centre, which then set up a link to its counterpart in Ireland, which then… - well, you get the idea. Eventually, Ma's phone would ring and when she picked it up it was as if a continuous length of copper cable stretched from my handset in Cambridge to hers in County Mayo. But that was an illusion created by circuit-switching.

Circuit switching was fine for voice communication. But it was hopeless for linking computers. It was inherently slow – think of all those switches which had to be thrown in sequence to make the connection. As time and technology moved on, and the switches changed from being clumsy electro-mechanical devices to all-electronic ones, the time needed to set up a call reduced. But still it was slow compared to the operating speeds of computers – even in 1965.

Secondly, circuit-switching was intrinsically uneconomic. Telephone plant – lines, exchanges, switches and so on – costs money to install and maintain. It involves expensive assets which ought to be worked as hard as possible. But while Ma and I were talking, nobody else could use the line. And the circuit had to be maintained for the duration of the call, even if there happened to be long periods of silence during it. This might be appropriate for human conversation – where silences are often part of the interaction, as they were with Ma and me. But for data communications it was idiotic because transactions between computers are intrinsically 'bursty' - they take place in intense, short spurts punctuated by much longer periods of inactivity.

Consider, for example, the communications requirements of time-shared computing – then becoming common in 1972. Incidentally, it's worth remembering that the OU was the first university in the world to offer time-shared access to undergraduates – and we were doing it in 1972. Other universities provided such access to graduate students, of course, but I believe we were the first to give it to primary degree students.

In a time-shared system, the user sat at a terminal connected, via a circuit-switched line, to a host computer. For much of the time she – and the circuit (which, remember, is tied up by the connection) - was inactive. She was thinking, perhaps, or hesitantly typing in best two-finger style. Then she hit the Return key to send the line of text and there was a frantic burst of data down the (telephone) line, followed again by absolute silence. The host machine absorbed what the user had typed, and responded with another burst of characters which appeared on her screen. Then the line went quiet again. So most of the time, that expensive telephone connection was doing precisely nothing. In fact, it was reckoned that in a typical interactive computer session less than one per cent of the available transmission capacity was being used.

The point is simple: a network that is optimised for one application, may be useless for other – newer – applications. Which is why the old-style telephone network was doomed.

But I'm running ahead of the story.

-- o0o --

In 1972 – just around the time that we were writing our course on Systems Behaviour, a new communications network had come into being in the United States.[4] It was called the ARPANET, and it was specifically designed to enable computers to communicate with one another. Its technology was radically different from that of the phone system: it was a *packet-switched* network in which small packets of data were moved between computers over a network of permanent connections. It was funded by Bob Taylor, who was then a senior official in the Department of Defense's Advanced Research Projects Agency, and with whom I later had the honour of co-authoring a paper on managing research teams[5]. The ARPANET had been constructed in the teeth of opposition and professional incredulity on the part of AT&T. Despite this opposition, by the Autumn of 1972 the new network was up and running smoothly. It clearly worked.

But the ARPANET was a military system, with the kind of unitary architecture that you get when the prime contractor is the Pentagon. The question for ARPA was then: *what next?* Already by that time, several other experimental packet-switched networks were up and running. There was one, for example, created by Donald Davies[6] and his colleagues at the UK's National Physical Laboratory in Teddington; there was one in France, and another one – operating wirelessly – called ALOHA at the University of Hawaii. So ARPA set up a new research project – the Internetworking Project. Its objective was to create a network that would seamlessly mesh all these different networks into a global system.

The Internetworking Project was led by a group of young post-docs and graduate students who had cut their teeth on the ARPANET. Chief among them were Vint Cerf (who is now a Vice President of Google) and Robert Kahn, an exceptionally gifted engineer.

---

[4] With one node in the UK – at University College, London.

[5] John Naughton and Robert W. Taylor, "Zen and the art of research management". In Andrew Herbert and Karen Sparck-Jones (Eds), *Computer Systems: Theory, Technology and Applications*, Springer, 2004. Taylor was an inspired manager of researchers. He later (in the 1970s) set up and ran the Computer Science Lab at Xerox PARC which produced most of the computing and networking technology we still use today. See *Dealers of Lightning* [publication data]

[6] Who later served as External Assessor on *You, your computer and the Net*, the OU's first major online course.

The design problem that confronted these young men (and they were all males, I'm afraid) was as difficult to solve as it is simple to formulate. It was this:

How do you design a communications network that is future-proof? Or, more prosaically, how do you design a network for applications that nobody has yet invented?

The Cerf/Kahn solution was stupendously elegant and simple. It was based on two axioms.

First, you abandon the idea of ownership or central control. There should be no gate-keeper, no AT&T or BT determining what the network could be used for or who should be able to access it. This is the *permissiveness principle.*

Second, you do not optimise the network for any particular application. Instead you make the network as simple and general as possible, and leave all the ingenuity to the applications that people dream up for it. You assume, in other words, that your users are smart. This led to the idea of a network that simply did one thing: it took in data packets at one end, and did its level best to deliver them to their destinations at the other end. The network wouldn't care what's in those packets or what they represented. If you could do it with packets, then the new 'Internetwork' would do it for you. This is what became known *as the "end-to-end" principle.*

In the ten years between 1973 and 1983, Cerf and Kahn and their colleagues articulated these principles and implemented them in protocols and software. On January 1, 1983, the new network – now called the Internet – was switched on.

And at that moment the world changed.

-- o0o --

Most of the technologies which have shaped our civilisation have had their effects through unintended consequences or applications. The mechanical clock first made its appearance in Benedictine monasteries in the 12th and 13th centuries. The impetus behind the invention was, in Neil Postman's words, "to provide a more or less precise regularity to the routines of the monasteries, which required, among other things, seven periods of devotion during the course of the day. The bells of the monastery were to be rung to signal the canonical hours; the mechanical clock was the device that could provide precision to these rituals of devotion."[7]

But what the monks did not foresee, as Lewis Mumford pointed out, is that the clock is a means not only of regulating periods of spiritual devotion, but also of regulating the actions of men. "The mechanical clock", Mumford wrote, "made possible the idea of regular production, regular working hours and a standardized product".[8] In short, without the clock (which was designed to facilitate the worship of God), capitalism (which after all is the worship of Mammon) would have been impossible.

The history of technology is full of stories like this. Thomas Edison was convinced that the phonograph would find its main use as a dictation machine for business; Marconi regarded radio as a point-to-point technology; the inventors of the digital computer thought they were building a calculating machine; and so on. In each case, the inventor of the technology thinks he knows what his invention is for, but is wrong. In the end it's society that discovers what the killer application is.

The Internet is a radical departure from this pattern, in that its inventors explicitly acknowledged from the outset that they had no idea what it would ultimately be used for. So what was a bug for most inventors was actually a feature for them. And what they invented was essentially a machine for springing surprises. Or, more prosaically, *a system for enabling disruptive innovation.* For that was the logical consequence of their two design axioms: an ownerless, permissive network that did only one thing – deliver data packets from one end to the other.

What the design philosophy meant was that if you had a good idea for a product or a service, and if that product or service could be delivered by using data packets in the way specified by the network's protocols, then the Internet would do it for you, with no questions asked. There would be no gatekeepers telling you to keep out; and no moral policeman saying "Oi! You can't use the network for that!"

The result was an explosion of creativity which shook (and continues to shake) the established order – in business, in government, in security, in education – to its core. It

[7] Neil Postman, *Technopoly: the surrender of culture to technology*, Vintage, 1993, page 14.
[8] Lewis Mumford, *Technics and Civilization.*

would take much more than a lecture to chronicle the extent of this explosion – to traverse what Manuel Castells calls "the Internet galaxy"[9] – so I'm going to focus today on just three examples which illustrate the profound consequences of the end-to-end architecture that Vint Cerf and Bob Kahn devised.

The first of these surprises was the World Wide Web.

Nobody knows how big the Web is. When Google stopped boasting about the number of web pages that it indexed, the figure stood at just over eight billion. At a conference here the other day, the head of research at Yahoo talked about 20 billion web pages. There are serious people out there who reckon that this is just the tip of an iceberg – that the total size of the web may be 400 times the number of pages indexed by search engines. If that's true, then we're looking at a Web that already contains more than *three trillion* pages. It's an unimaginable transformation of our communications environment. Imagine now what life would be like without it. And it has come into being in just over a decade and a half.

The most significant thing about the Web, though, is not its scale but the fact that it was effectively the creation of a single individual – the British physicist Tim Berners-Lee. In the late 1980s Tim (whom I had the honour to present for an honorary degree in this theatre in 2001) was working at CERN, the European particle physics lab in Geneva. As a huge research facility with a large shifting population of visiting scientists from all over the world, CERN suffered from terrible collective memory problems: documents were always getting mislaid; research results were held on experimenters' computers all over the world; it seemed impossible to conceive of a filing system that would hold all the data generated by the particle accelerator.

So Berners-Lee sat down and worked out a scheme for a global hypertext system that would enable information and data resources to be stored and retrieved from all over the Internet. He called it the World Wide Web. In a single *annus mirabilis*, working with a very small group of colleagues, he designed the protocols and wrote the server and browser software needed to implement the idea. And then, in January 1991, he released it on the Net. The rest, as they say is history.

---

[9] Manuel Castells, *The Internet Galaxy,* Oxford University Press, 2001.

But note this. Berners-Lee consulted *nobody* about any of this except his managers at CERN who gave him permission to spent *time* working on the project. When the software was finished, he did not have to seek the permission of anyone to launch the Web. He simply put the software and the protocols on the Internet, and the network did the rest. In doing so, this modest, self-effacing physicist changed the world. And what enabled him to do so was the end-to-end principle embedded in the network's DNA.

Tim conceived the Web in 1989. Ten years later, in the United States, a disaffected, music-obsessed teenager named Shawn Fanning was wondering how to find and share music on the Net. People forget that there were always music files on the Net, but in 1989 a German company, Fraunhofer Geshellschaft, obtained a patent for a method of compressing audio files which reduced their size without noticeably degrading sound quality. The official name for this technique was "MPEG Audio Layer III", but it rapidly became known as MP3 and it had the effect of making it easier to transfer audio files, simply because it made them much smaller. So the arrival of MP3 encoding led to an explosion of music files stored on hard drives on both personal computers and Internet servers.

The problem was: how could a music lover find these files? And once you'd found them, how could you obtain a copy? This is what was bugging Fanning, so he sat down and wrote some software that solved both problems. His solution came in a two-part package: a small, downloadable 'client' program that would run on people's PCs; and a server program that could act as a kind of constantly updated dating bureau which put PCs seeking a particular track directly in touch with PCs whose owners had that track and were prepared to share it. Fanning called his system Napster, and in 1999 released it onto the Net.

I don't need to tell you what happened next. Napster went from zero to 80 million users in the eighteen months it took the record industry to get it shut down.[10] But by the time it was finally closed, the genie was out of the bottle – teenagers all over the world had got the file-sharing habit. And they haven't lost it: today, more music files are being illicitly 'shared' over the Net than were ever exchanged in the heyday of the original Napster.

---

[10] There is now a new company called Napster (www.napster.com) which is a legitimate vendor of online music.

Now of course everybody knows this story. But strangely enough, I don't think people have fully absorbed its lessons. It provides, for example, another vivid demonstration of the power of the end-to-end principle. Shawn Fanning had an idea that could be realised using data packets. So he wrote the software and the Net did the rest.

The file-sharing phenomenon is a vivid illustration of the capacity of the Net to spring surprises. In this case, the surprised party was the record industry, which ought to have recognised the Internet as a heaven-sent technology for distributing its product, but didn't. And by the time it woke up, the game was over. Despite all the legal huffing and puffing of the industry and its expensive lawyers, the die is cast. From now on, all recorded music will be online, period. The only question is: what percentage of it will be paid-for?

More than that, what Napster showed was that music need never go 'out of print', as it were. Up to 1999, only a tiny fraction of all the music ever recorded was obtainable for purchase. When the stock of recordings of a particular album was sold out, then the music was effectively unobtainable, unless the publisher decided to reissue it. But Napster showed that every recording ever made could be available 24x7x365 – which is why the service was correctly dubbed "the celestial jukebox".

For me, though, the most interesting aspect of the Napster story is what it tells us about innovation and the Internet's role in it. It turns out that what Shawn Fanning invented was something far bigger than a device for ripping off record companies. It was in essence an innovative way of harnessing what Clay Shirky calls "the dark matter of the Internet"[11] -- those millions of hitherto-useless PCs connected to the Net. The technology was later dignified with the name "peer-to-peer" (P2P) networking and it has evolved to become one of the most important communications technologies that we have. It is, for example, the technology underpinning the wildly-successful Skype Internet telephony service which is currently growing at the rate of 150,000 new subscribers *a day*. Peer-to-peer networking is the technology by which the Hollywood studios will use to distribute their movies online. And yet, if you were to listen to the clamour of the record companies and their media megaphones, you would think that

---

[11] Clay Shirky, "PCs are the Dark Matter of the Internet", available online at: http://www.shirky.com/writings/dark_matter.html (last visited 15 June, 2006).

P2P was the spawn of the devil, rather than an increasingly important part of our networked future.

I want to turn finally to the third great surprise that the Internet has sprung on us: the spread of open source software, and in particular the Linux operating system.

The term "open source' is in fact a business-friendly euphemism for "free software". The idea that software should be free originated in the 1980s with Richard Stallman, a celebrated MIT researcher. What he meant by it was *not* that software should cost nothing, but that users should have the freedom to modify the code to suit their own circumstances. "Free as in free speech, not as in free beer" is how Stallman explains it. The French would say software *libre*, not software *gratis*. But however the principle is expressed, at its heart is the freedom to tinker.

Stallman's great idea was to devise a licensing system that would enable programmers not only to write and distribute free software, but also to ensure that the software thus created would remain free. He came up with a special licence – called the GPL (which stands for the GNU[12] General Public Licence) – which uses copyright law to ensure that anyone who takes advantage of the freedom to tinker in order to create new software has to pass on the same rights to any subsequent users of what they create. In a neat inversion, Stallman christened this ingenious application of copyright "copyleft".[13]

The GPL was one of the seminal innovations of the 20th century. It has enabled programmers to create, share, improve and maintain much of the technical infrastructure that underpins the modern world. The Internet runs on free software, for example; most web pages are served by free software; most email travels courtesy of free software. And, most importantly of all, Linux is free software.

The origins of Linux are as striking as the origins of the web. Once again, it goes back to a single individual – in this case a Finnish graduate student named Linus Torvalds who bought a PC in 1991 and didn't want to run a Microsoft operating system on it. So he took and modified a toy operating system called Minix[14], which had been

---

[12] GNU stands for "Gnu's Not Unix" – which is a programmer's joke. Stallman had set out to create a free clone of the Unix system originally created by AT&T Researchers. He called his project the GNU Project. Programmers love recursion!

[13] Sam Williams, *Free as in Freedom: Richard Stallman's crusade for free software*, O'Reilly, 2002.

[14] See http://www.minix3.org/ (last visited 11 June, 2006)

created for teaching purposes by Andy Tanenbaum at the Free University of Amsterdam and released under a free software licence. Torvalds then released *his* toy operating system onto the Net and invited people to critique and improve it. Using the free software tools developed by Stallman and his colleagues at MIT, other programmers did just that[15], and Linux evolved out of the collaborative efforts of first dozens and later hundreds of programmers into the sophisticated operating system that it is today.[16]

If you're not a programmer then the significance of this may have escaped you. A modern operating system – like Linux or Microsoft's forthcoming new version of Windows, *Vista* – is an unimaginably complex object. Vista has 50 million lines of code and more than 50 layers of interdependency[17]. In terms of numbers of components and the density of the interactions between them, Linux and Vista are far more complex than a jumbo jet.

Yet Linux has been created by a community of developers who rarely if ever meet face-to-face, and who work for free. And they have produced something that is remarkably robust and stable. For example, I gave up using Windows-based computers in 1999 because of the fragility of the software: I found that every machine I owned gradually degraded, and wound up having to reboot my computer several times a day. In contrast, the servers which host the various websites for which I am responsible all run on Linux. I cannot remember when we last had to re-boot any of them: some have been running continuously for over 200 days. So it's not surprising to find that Google runs on Linux. So does Amazon. So does most of the stuff that NASA owns. If you have to bet your life – or your business – on the stability of an operating system, then Linux is what you choose.

And yet this extraordinary artefact has been produced by a group of people linked only by commitment, technical virtuosity – and a communications network which enables them to collaborate. In that sense, Linux is the best example we have to date of what Manuel Castells[18] calls the networked enterprise.

---

[15] Which is why, strictly speaking, Linux ought to be called GNU Linux.

[16] See Glyn Moody, *Rebel Code: Linux and the Open Source Revolution*, Perseus, 2001.

[17] http://blogs.msdn.com/philipsu/archive/2006/06/14/631438.aspx (last visited 18 June, 2006). According to this source (a former Windows developer who works at Microsoft) its predecessor, Windows XP had 40 million lines of code.

[18] Castells, *op. cit,*

And this in turn has a further significance. Remember what I said about the complexity of a modern operating system. For the last few years, we have watched Microsoft – the richest, most talented, most determined company in the computer business and (according to the Financial Times[19]) the third most valuable company in the world – grappling with the gargantuan task of creating Vista, the latest incarnation of Windows. The chequered history of Vista is a story of a great corporation buckling under the strain. When it finally ships early next year Vista will be years behind schedule, and in its early releases at least, will be only a pale shadow of what was once envisaged. The corporate trauma wrought within Microsoft by the struggle to bring Vista to market prompts a sobering thought: will there ever be another major upgrade of Windows? Is the task of creating such a code monster now beyond the reach of even the most able and wealthiest corporation? Have we reached the point where we have to concede that huge software systems can no longer be built in the traditional, monolithic manner – that we need to do these things another way? And might the process that produced Linux be that way?

Curiosity about the effectiveness of the open source production model is what led the Berkeley political scientist, Steve Weber, to embark on the first sustained study of how the Linux model works.[20] His conclusion is that the most significant thing about open source is not the software it produces, but the method it has invented for making unimaginably complex things. He sees an analogy in the Toyota lean production system of automobile production as revealed in a celebrated 1980s MIT study[21]. The Japanese company invented a way of making cars that was radically different from how they were made by Western manufacturers.[22] In the end, these manufacturers faced a stark choice: make cars the Toyota way or go out of business. Now, all cars are made using the Toyota approach. The MIT study, writes Weber, "made two simple and profound points: the Toyota 'system' was not a car, and it was not uniquely Japanese".

---

[19] "FT Global 500", *FT Magazine*, 10/11 June, 2006, page 22.

[20] Steve Weber, *The Success of Open Source*, Harvard, 2004.

[21] As described in James P. Womack, Daniel T. Jones and Daniel Roos, *The Machine That Changed the World: The Story of Lean Production*, HarperBusiness, 1990.

[22] "Lean production ... uses teams of multiskilled workers at all levels of the organization, and uses highly flexible, increasingly automated machines to produce large volumes of products in enormous variety. The term "lean" comes from its using half the human effort in the factory, half the manufacturing space, half the investment in tools, and half the engineering hours to develop a new product in half the time." It also led to much higher product reliability and reduced warranty costs. See http://www.dau.mil/educdept/mm_dept_resources/navbar/lean/02tch-mtctw.asp (last visited 11 June 2006) for an overview.

By the same token, he continues, “Open source is not a piece of software, and it is not unique to a group of hackers”.[23]

-- o0o --

John Beishon recruited me to work here at the Open University in 1972. By coincidence, that was also the year when I began to write for the *Observer*, the newspaper for which I have written regularly ever since. So I have been an academic and a journalist all my working life. Conor Cruise O’ Brien, my famous fellow-countryman, was similarly afflicted. In fact he once said, *apropos* his dual career, that he had “a foot in both graves”.

In recent years, I have been the *Observer*’s Internet specialist. Within the office in London, my weekly column has been assigned an interesting name. On Friday mornings the Editor of the Business Section of the paper can sometimes be heard shouting, “Where the hell is Naughton’s rant?” – except that he doesn’t use the word “hell”.

I suppose I should be annoyed by this, but secretly I’m rather pleased. At least it carries the implication that I feel strongly about my subject.

And I do. In fact, I devoted a year of my life to writing a book[24] about the origins and significance of the Internet because I felt strongly about the way influential people in our society were belittling a development that I regarded as being as liberating as the printing press had been four centuries earlier. I still remember the day I started on the book. It was after a lunch with the editor of a prominent British newspaper (I won’t say which one, but it wasn’t the *Guardian* or the *Observer*) who had said, with that sneering tone that the English ruling class traditionally reserves for engineering, “About this Internet thingy, deah boy – isn’t it just the Citizen Band radio *de nos jours*?”

It was an intensely irritating remark at the time, of course, but it had a salutary effect because it forced me to ask what was really significant about the Net. And that led me to some of the insights that I have tried to share with you this afternoon. The more I thought about it – the more I contemplated the implications of the architecture that

---

[23] Weber, *op. cit,* page 224.

[24] John Naughton, *A Brief History of the Future: the origins of the Internet*, Phoenix, 2000.

Vint Cerf and Bob Kahn conceived -- the more I realised how radical a technology this was.

And what made it radical was essentially very simple. Cerf and Kahn had designed --- not a network of computers, or even a network of computer networks -- but a system for connecting *minds* which enabled them to innovate, to communicate and to collaborate *without dictating the terms under which they could do so*. In other words, it's the *people* who use it that make the network so powerful, not the technology itself. It's obvious, really – but then, as many of you know, I am slow on the uptake. The funny thing is that when the penny finally dropped, what came into my mind was nothing to do with technology, but some lines from TS Eliot's poem, *Little Gidding*:

We shall not cease from exploration
And the end of all our exploring
Will be to arrive where we started
And know the place for the first time.

If you have been, thank you for listening.