Topic 4

Dependence and Change

Contents

Learning Objectives

- **Deliberation on the relationship between technology and society**
- **Awareness of the impact of technology on society**
- **Awareness of the influence of society on the development of technology**
- **Understanding of the degree of dependence of modern society on technology**

4.1 Introduction

We have already mentioned the degree to which modern society has become dependent on technology. The impact of technologies such as the motor car, the telephone, television, etc. are well documented. The impacts of computer technology are still being discovered but much has already been written about them. We shall attempt to summarise the current situation in this topic and look to the future in the next topic.

It is important to note, however, that there are two cause and effect relationships at work. It is not only the case that society is influenced by technology. The direction and speed of technological development is also heavily influenced by society. Since this latter relationship is often hidden and working behind the scenes we shall devote more space to it here than to the more obvious influences in the other direction.

Computer technology, like any other technology, brings risks as well as benefits. Indeed, the ubiquity of computer systems means that there are probably more such risks, with greater potential dangers, inherent in computer technology than any other that mankind has had to deal with (Neumann, 1995). We shall consider a particular class of computer system known as a safety-critical system in which failure would be disastrous and we shall look at some systems where that disaster actually occurred.

The best known example of our dependence on computer technology was the Year 2000, or Y2K problem, also known as the Millennium Bug. We shall examine this as a case study in computer dependence but also as a study in the trust which the world now has to place in the computer industry and computer professionals. Are we living up to that trust? Many would say the Y2K problem showed that we are not.

4.2 Technology and society

Technological developments affect society at large both directly and indirectly. There are few aspects of our work, recreation, domestic life, welfare or law enforcement services that are not affected by computers (Baase 2003, Spinello and Tavani 2001). Try thinking of some daily activity that has not been affected by computer technology. It won’t be easy. Brushing your teeth, perhaps? Your toothbrush and your toothpaste were both almost certainly developed with computer assistance. The water arriving at your tap has probably been delivered with the assistance of computer control at some point in its journey. I could go on but I’m sure you get my drift and I do not wish to dwell on the broader impacts of technology on society with which I am sure you are already very well acquainted.

It is the influences in the other direction which are not so obvious. Societal factors can, and do, affect the speed and direction of technological developments (Bijker et al., 1989). They always have but it is indicative of how little recognition we give them, that we have great difficulty in recollecting any examples. Try thinking of some. Again, it won’t be easy. Commercial, political, cultural and economic pressures and choices have had a considerable impact on the technology we see around us. It is worth looking at a couple of examples from the past even though, by definition, they will mostly pre-date computer technology (MacKenzie and Wajcman, 1985).

4.2.1 The refrigerator

At the turn of the twentieth century, the idea of a refrigeration machine was conceived. The key question at the time was how to power it. Gas or electricity? You know the outcome but do you know why? Let’s look at the issues.

Gas refrigerators have hardly any mechanical parts whilst electric refrigerators need a compressor and a motor. Therefore gas machines are much less likely to break down. The early electric models made a lot of noise and even nowadays they aren’t silent, as gas ones are.

Gas supplies were more prevalent than electricity supplies. Gas had been around longer and the infrastructure was therefore more extensive. More homes could take advantage of a refrigerator if it was powered by gas.

The gas refrigerator was supported by two comparatively small companies - Servel and SORCO - whilst the electric refrigerator was supported by three giants - General Electric, General Motors and Westinghouse.

The power behind modern day refrigerators is electricity and not gas because of commercial power, not because it was the best decision technologically.

4.2.2 The machine tool

By the middle of the twentieth century it had become possible to develop an automatic version of the machine tools used by skilled craftsmen for high precision metal working. These machines would do the job of the machinists more consistently and for longer periods at a time. The issue here was how they were to be instructed what to do – by recording, and then playing back as often as you liked, the actions of a skilled metal worker or by using a numerically controlled (NC) machine which had to be programmed.

The record/playback approach was comparatively cheap, whilst the NC machines were very expensive.

The record/playback machines used a skilled metal worker which would seem to have been an advantage over the programmers, with little metal working knowledge, necessary for the NC machines.

However, these machines became the subject of hot political debate in certain quarters, notably the United States Air Force (USAF). Machine tools were used for the high precision metal working necessary in aircraft production. There were, almost certainly unwarranted, concerns at the time about the reliability of workers who belonged to trades unions and the option of eliminating the need for these workers appealed to a number of politicians.

The outcome was that the supporters of the record/playback approach were bought out. Not only that but, because the NC machines were so expensive that the suppliers who were to use them couldn’t actually afford them, the USAF paid for the installation of the machines in the sub-contractors’ factories.

It is an ill wind that blows nobody any good, and it must be said that although the decision to develop NC machines was a political one, with little technological or economic merit, NC became an important driving force in the early development of computer aided manufacturing.

4.2.3 The computer industry

You may be thinking that the refrigerator and the machine tool are a long way from the modern computer industry and we have little to learn from such examples. There are, however, plenty of examples to be found in the world of computer technology. The trouble is that the technologies that weren’t taken up and developed aren’t here now, by definition, and so we can’t see what we are missing or what might have been.

One of the early battles in the modern era of computing, and there were plenty of others before the modern era, was fought between Intel and Motorola. Both companies set out to produce 16-bit processor chips. Motorola's 68000 chip was a full 16-bit chip with both address and data buses being 16-bit. Intel's 8086 chip, however, whilst it had a 16-bit address bus, had only an 8-bit data bus. In the transition from 8-bit to 16-bit, Intel had only gone half-way, two fetches were needed by the Intel chip to retrieve the full 16-bits of data and it was thus slower than Motorola's offering. When IBM launched their personal computer, which spawned the home computer revolution, they chose to use the Intel chip rather than the superior Motorola one. Motorola continued to make superior chips, generally keeping one step ahead of Intel but selling to a much smaller market than Intel had cornered thanks to IBM and the IBM compatible PCs, most of which have an Intel inside to this day.

The Apple Macintosh, which was in many respects superior to the IBM PC, used Motorola chips but such was IBM's dominance of the market that Apple computers remain the computer of choice for only a small proportion of the world's computer users.

Another company to benefit from IBM's early dominance was Microsoft. You won't need to be told how dominant Microsoft has become but you might pause to ponder on where we would be today if Apple and Motorola had enjoyed the same success as IBM and Intel. Would Microsoft have backed both horses perhaps, or might they have chosen the wrong one and remained a small player in the computer world?

In recent years Microsoft have had to fight a number of legal battles to defend their proprietary software from the ever-increasing volume of open source software which is threatening some of their key products - their MS Office suite of programs, their MIE web browser and even their MS Windows operating systems. Are Microsoft just too big a company to lose out in the long term? We shall have to wait and see but, whether we approve of Microsoft's current dominance or not, we should never gainsay the enormous contribution it has allowed them to make to all of our lives.

4.3 Safety-critical systems

Our dependence on computer technology is at its most sensitive when we put our lives in its hands. Hazard analysis and risk assessment are essential steps in the development of a system where safety is critical (Levenson 1995).

4.3.1 Hazard analysis

Hazard analysis is the first step in ensuring that a system will cope with all the eventualities that it might encounter. The analysis starts with hazard identification; determining the hazards that might exist when the system is used. These hazards should then be classified according to their severity and likelihood of occurrence. Each hazard should then be decomposed in order to identify the precise circumstances under which it will arise.

4.3.2 Hazard severity and likelihood categories

The US Department of Defense uses four categories to identify the severity of a hazard. They are, in descending order of severity, Catastrophic, Critical, Marginal and Negligible. Levenson (1995) recommends six categories of hazard likelihood. In descending order of likelihood they are Frequent, Probable, Occasional, Remote, Improbable and Physically Impossible.

4.3.3 Risk assessment

The final risk assessment has to balance a number of things. Costs and delivery times are the main factors which affect how much development time and effort can be devoted to safety measures. There is no such thing as absolute safety but worst case scenarios should be used to combine the hazard severity with its likelihood to ascertain whether the final system can be made safe enough and whether this can be accomplished within budget. It is quite proper for the risk assessment to conclude that the development should not take place if the hazards cannot be brought down to an acceptable level. What is acceptable, of course, requires professional judgement.

4.3.4 Failure examples

We shall now briefly describe two notorious software failures. We leave you to judge whether negligence was involved.

Ariane 5

The explosion of the European Space Agency's Ariane 5 rocket on its maiden flight in June 1996 was caused by loss of guidance and attitude information shortly after lift-off. The official inquiry into the disaster found that this was due to specification and design errors in the software of the inertial reference system. The inquiry discovered that tests conducted on the system were inadequate. If they had been performed more thoroughly the disaster would probably have been avoided. Mercifully this was an unmanned mission and nobody was killed or injured. This incident demonstrates that computer software has become so complex that bugs can remain undetected, even in the incredibly thorough and safety-conscious world of space technology.

Therac-25

Between June 1985 and January 1987 there were six accidents involving massive overdoses of radiation from a Therac-25 radiation therapy machine. These accidents, which resulted in deaths and serious injuries to patients, were caused by software errors. The errors had also been present in an earlier model, the Therac-20, but because that was an older machine, there had been additional hardware safety mechanisms which had prevented the accidents. The more modern Therac-25, it had been felt, could rely on software safety mechanisms alone.

4.4 Y2K – A case study

The first thing to note about the Y2K problem was that it was more than one distinct problem. In addition to the issues surrounding the use of a two digit format for the year (the practice of dropping the 19 from the front of the year), there was a further problem in that the year 2000 was a leap year but not everybody knew it. The year 1900, for instance, hadn't been a leap year and the part of the leap year rule which applied to the year 2000 hadn't been applicable since the year 1600. So, in addition to the two digit year problem, there was also the possibility of some computer systems working on a 365 day year rather than a 366 day year and incorrectly leaving out the 29th of February.

However, we shall leave the leap year issue aside and concentrate on the problems caused by the two digit year format. The first problem is that if only two digits are used then time periods in excess of 99 years are not storable and might lead to systems crashing as a result of overflow problems. The second problem is that if you are dealing with dates that span the millennium boundary then the century becomes ambiguous and dates can be sorted incorrectly. The year 2000 should come after the year 1999 but if the first two digits are dropped from both then the year 00 will be sorted to fall before the year 99. The third problem, which made everybody hold their breath at midnight on the 31st of December 1999, was what would happen when the year 99 had 1 added to it? Would it try to become 100 and crash the computer? Would it become -1 with equally devastating results?

With so many computers out there, in people's homes, embedded in their domestic appliances, in safety-critical systems such as aircraft control systems, life support systems and nuclear power stations it is not surprising that many people, both within the computer industry and without, feared the worst. No system could be considered immune. System clocks and date reckoning could turn up in the least likely of places for no other reason than that they were a built-in function of the processor being used.

This was certainly a no-win situation for the computer industry. Firstly, and with some justification, they were blamed for the problem arising in the first place. In their defence though, we should point out that the industry had been aware of the problem for some time and had been warning organisations that their legacy (old) systems needed to be replaced or upgraded. The programmers of the 1960s and 1970s really hadn't expected their programs to still be in use at the turn of the century.

However, no matter where the blame lay, it was clear that if the computer industry encouraged governments and private industry to spend enormous sums fixing the problem then there wouldn't be problem any more and they would be accused of scare-mongering to increase their own profits. On the other hand, if they played the problem down then there would undoubtedly be catastrophic failures of some systems and they would be blamed for that instead.

There is no question that the Y2K problem was real enough. During simulations of the roll-over from 1999 to 2000 –

- *All the computer screens went blank at the International Federation of Airline Controllers*
- *4 million gallons of raw sewage were dumped onto a Los Angeles street*
- *A robot assembly line crashed at a General Motors factory and the security system prevented staff from leaving*

Following the millennium there were –

- *Nuclear power plant failures in Japan, Spain and the USA*

- *Healthcare system failures in Brazil, Norway, Sweden and the UK*
- *Power distribution failures in Honduras and South Korea*

All of these incidents were genuinely attributable to the Y2K problem. In the run up to the year 2000 though, it wasn't at all clear just how many such cases there might be.

World-wide, the cost of the Y2K fixes that were actually implemented has been estimated at £300,000 million. The USA alone spent £60,000 million, including £30 million on a special Y2K command centre. £20,000 million was spent in the UK but in Italy only £500 million was spent, including £1.5 million on billboards to warn people of the risk. Italy fared no worse than the USA or the UK as it turned out.

Perhaps it was what was learned from the Y2K problem that will, in the end, justify all the fuss it caused. Computer professionals learned that there was a lot of legacy software out there and the life expectancy of software was considerably longer than they had realised. They learned that most IT departments didn't have a clear idea what software they actually had and were using. There was a need for proper software inventory management, methodologies and tools. Computer users learned how reliant they had become on IT and that that reliance was both broader and deeper than they had realised. Company executives learned that IT had become too important to be left solely to the boffins. Managers, from the boardroom down, learned that they need to be considerably more critical of, and better informed about, IT solutions and practices.

What should the final verdict on the computer industry be? Take your pick –

1. The fact that the Y2K problem arose demonstrates that the computer profession is immature and its practices are unsound
2. The computer profession's reaction to the Y2K problem was a triple success story –
   i). Success in raising awareness of the problem
   ii). Success in persuading enterprises to invest in fixing it
   iii). Success in fixing the problem
3. The computer profession refused to accept any liability for the problem, provoked widespread hysteria and then exploited the fear it had generated for financial gain

Here's a final thought. The Unix operating system uses a 32-bit signed integer to hold the time and date. This variable has been counting the seconds since midnight GMT on the 1st of January 1970. It will roll over to a negative number at precisely 03:14:07 GMT on the 19th of January 2038. Is that too far off to worry about?

4.5 Assigned task

You should prepare a 15 minute presentation on one of the following two topics. The presentation will be made at either your fourth or fifth tutorial meeting. Your presentation might benefit from a consideration of some of the issues covered in the next topic.

1. Computer Games

Eugene Provenzo has compared computer games unfavourably with more traditional toys as follows –

"In the case of [computer games] the child has almost no potential to reshape the game and its instrumental logic. There is literally one path down which the player can proceed. The machine and its program impose an instrumental logic on the play situation and the activities of the child. In the light of evidence … concerning the violence, aggression, and stereotyping found in these games, this fact is particularly disturbing."

Eugene Provenzo, "Video Kids"
Harvard University Press, 1991

Computer games have advanced considerably since this was written but should we be any less concerned than Provenzo? Are there reasons to be even more concerned now?

2. Open Source

The protagonists in the open source versus proprietary software debate have made the following statements –

"Digital information technology contributes to the world by making it easier to copy and modify information. Computers promise to make this easier for all of us. Not everyone wants it to be easier. The system of copyright gives software programs "owners", most of whom aim to withhold software's potential benefit from the rest of the public. They would like to be the only ones who can copy and modify the software that we use … What does society need? It needs information that is truly available to its citizens – for example, programs that people can read, fix, adapt, and improve, not just operate. But what software owners typically deliver is a black box that we can't study or change."

Richard Stallman, Founder of the Free Software Foundation
"Why Software Should Not Have Owners", 1994

"It [the GNU General Public Licence] also fundamentally undermines the independent commercial software sector because it effectively makes it impossible to distribute software on a basis where recipients pay for the product rather than just the cost of distribution … Two decades of experience have shown that an economic model that protects intellectual property and a business model that recoups research and development costs can create impressive economic benefits and distribute them very broadly."

Craig Mundie, Senior Vice President, Microsoft Corporation
"The Commercial Software Model", May 2001

To what extent do you agree with each of them? Are there elements of truth in both statements?

4.6 End of topic test

<!-- IU
Please insert a multiple choice test here. The correct answers to each question are underlined.
-->

Q1. The Y2K problem was also known as the –
a). Bug of the century
b). Millennium bug
c). Roll-over bug
d). Two thousand year itch

Q2. Societal factors can affect both the speed and what of technological developments –
a). Cost
b). Direction
c). Popularity
d). Size

Q3. Which of the following was NOT cited as a societal pressure on technological developments –
a). Commercial
b). Cultural
c). Fashion
d). Political

Q4. Which of the following pairs was NOT suggested as an advantage of gas refrigerators over electric refrigerators –
a). Availability of gas
b). Cooler
c). No mechanical parts
d). Quieter

Q5. Microsoft's and Intel's success was attributed to which company's market dominance –
a). Fujitsu
b). IBM
c). Motorola
d). Siemens

Q6. Which of the following is NOT one of the US Department of Defense hazard categories–
a). Catastrophic
b). Critical
c). Moderate
d). Negligible

Q7. The Ariane 5 explosion occurred in which year –
a). 1957
b). 1966

c). 1996
d). 2001

Q8. The Therac-25 accidents were attributed to what kind of error–
a). Hardware
b). Medical
c). Software
d). User

Q9. Which of the following is NOT a leap year –
a). 1600
b). 1900
c). 2000
d). 2004

Q10. What was the estimated cost world-wide of the Y2K problem –
a). £3 million
b). £30 million
c). £300 million
d). £300,000 million

References

Baase, S., 2003, *A Gift of Fire: Social, Legal and Ethical issues for Computers and the Internet*. Prentice Hall.

Bijker, W.E., Hughes, T.P. and Pinch, T., 1989, *The Social Construction of Technological Systems*. MIT Press.

Levenson, N.G, 1995, *Safeware: System Safety and Computers*. Addison Wesley.

MacKenzie, D. and Wajcman, J., 1985, *The Social Shaping of Technology*. Open University Press.

Neumann, P.G., 1995, *Computer Related Risks*. Addison Wesley.

Quinn, M.J., 2005, *Ethics for the Information Age*. Addison Wesley.

Spinello, R.A. and Tavani, H.T., 2001, *Readings in CyberEthics*. Jones and Bartlett.