WHY WE MISSED THE YEAR 2000 COMPUTER PROBLEM

By Stuart A. Umpleby

Research Program in Social and Organizational Learning
The George Washington University
Washington, DC 20052 USA
Email: umpleby@gwu.edu
URL: http://www.gwu.edu/~umpleby

Revised December 16, 1999

Presented at the 11th International Conference on Systems Research, Informatics and
Cybernetics, Baden-Baden, Germany, August 2-7, 1999

"No one knows what to do against the really new." W. Ross Ashby

The year 2000 computer problem has the potential to be a major disruption of the current worldwide economic growth and development (U.S. Senate, 1999). Many people should have seen it coming – computer scientists, futurists, system scientists, cyberneticians, science fiction writers, reviewers of software products, etc. The irony is that almost everyone familiar with computers knew about the use of two digit year dates, but virtually no one thought about the implications. People were aware about the dangers in the early days of computing, but not in recent years. How did we overlook such a significant phenomenon? There are several reasons. The focus of attention was elsewhere. Assumptions were made which were not valid. Many people lacked essential knowledge. There was no precedent or suitable analogy to guide understanding. There were institutional weaknesses. We missed opportunities. And human psychology was not helpful.

THE FOCUS OF ATTENTION WAS ELSEWHERE

Information technology has come along so fast and has been so exciting that most people focused their attention on creating or keeping up with new technology. No one seems to have been looking carefully at the robustness or the resilience of the systems which had already been built and installed. We tended to believe that information technology changes rapidly, and, indeed, some of it does. But some computer programs created in the 1950s and 1960s are still being used. Over the years they were expanded and modified, but the core processes continued to use two digit dates (Yourdon and Yourdon, 1998).

Prestige went to those creating new technology, not to those maintaining the old technology. The people maintaining the old systems were not the most innovative people. Innovators did not want to spend their time on a problem they saw as being technically trivial.

When we did rediscover the problem, it was assigned to the information technology (IT) staff – the people who install and maintain computer equipment. But these are not the people who are responsible for embedded systems. So people worked on the year 2000 computer problem for several years before the problem of embedded systems was widely recognized. Buildings, assembly lines and basic utilities are the responsibility of the production or maintenance departments, not the IT departments.

ASSUMPTIONS WERE MADE THAT WERE NOT VALID

People made many assumptions that were not valid. In the early days of computing each new computer required new software. Since new computers were appearing every few years, no one believed that software might be used for many years. However, the IBM 360 computer was designed so that it would run programs that had been written for earlier computers. The feature of "backward compatibility" greatly reduced the cost of upgrading to the newer, faster machines. Subsequent machines were also designed in this way, but the assumption, that software would have a limited life span because new machines would require new software, was not reexamined.

We had the idea that software does not spontaneously malfunction. We thought that if we could make a program work correctly once, it would continue to work correctly indefinitely. Computers are logic machines. Physical devices may spontaneously fail due to worn out parts, metal fatigue, etc., but software does not spontaneously fail. However, in the case of the year 2000 computer problem (y2k), it does. Operations which formerly yielded positive numbers can suddenly produce negative numbers, and records which previously had been in sequential order may suddenly no longer be in order.

The chief executive officers (CEOs) assumed that the chief information officers (CIOs) were able to keep essential equipment running. When CIOs told CEOs that there was a problem with the programs, the CEOs frequently did not understand the magnitude of the problem. Changing two digits to four digits sounded simple, even trivial. Why would a lot of time and money be required? However, CIOs themselves often underestimated the problem as is indicated by rising budgets to fix the problem.

The public assumed the technical people could keep systems running. They had been doing so for many years. After all, technology fails quite often, but problems are usually solved within minutes, hours, or days. The year 2000 problem is different because there is the possibility of many different kinds of failures at the same time. This means that a failure which might normally be repaired in a few hours or days may, in January 2000, not be repaired for weeks or months simply because more urgent problems are being attended to. More seriously, technological disasters are usually not caused by a single failure. Disasters usually happen when two or three systems fail at the same time, or when human beings respond incorrectly.

The problem was defined as a technical problem, or perhaps a management problem, but there were psychological and organizational issues as well, particularly for those organizations that started late.

LACK OF KNOWLEDGE BY MANY PEOPLE

Many people lacked essential knowledge. For example, few people know what a SCADA (Supervisory Control and Data Acquisition) system is or does, but they are essential parts of utility networks. Most members of the public never understood the full implications of y2k. People thought there may be some errors in telephone bills or bank statements, but they did not realize that they might be without water, electricity, or gas for a period of weeks or even longer. They did not realize that not just the billing and payroll systems of utilities were at risk but also the equipment for delivering services.

There was no procedures manual for y2k. Most organizations did not even have a list of the hardware and software they were using. Frequently documentation was missing. Sometimes even the source code was missing. (This means that if a program is printed out, all one sees are zeros and ones. A human being cannot read it.) Tools for creating inventories of hardware and software, which had been available for years, had not been widely used. Also, the new software tools used to fix the y2k problem were unfamiliar to IT managers.

The customary lateness of software engineering projects was known by only a few people outside the field of software engineering (Boehm, 1981, Brooks, 1995). Hence, managers, journalists, and the general public did not have a healthy skepticism about the promises and reassurances that were made about how much equipment has been fixed. Normally when a software project is not completed on time, the organization simply continues to use the old system until the new system is ready. But in the case of y2k this is not possible. On January 1, 2000, or whenever the failure date is, the system stops or begins to generate errors. In this respect y2k is different from other software engineering projects.

Journalists did not understand the technical issues (Samuelson, 1998). They did not know what questions to ask, and they did not feel they had time to learn. They did not know how to keep probing. In the early days journalists felt that if their publication had run an article on y2k, they had covered it. Many activists thought that when the subject made it to the front page, then people would understand. But they did not. Most people who read the articles decided that y2k was a technical problem which the technical people would fix. They thought it did not concern them.

People do not understand the interdependencies in modern societies. They do not realize how many suppliers a manufacturing company has or how interdependent utility systems are. How many companies need to fail in order to cause the failure of nearly all companies? Fortunately, many companies have been putting pressure on their suppliers to fix their equipment for y2k. If these efforts have succeeded in reaching the most critical companies, perhaps failures by businesses which have not been working on y2k will not cause much disruption.

Karl Mueller (1998), a sociologist in Austria, notes that societies are now inhabited by what he calls "Turing creatures" after the mathematician and computer scientist Alan Turing. A Turing creature is any device which processes information and "makes a decision," from a thermostat in a room to a device on an assembly line. Mueller notes that there are now about 60 billion of these "creatures" functioning within human societies, about ten times as many as there are human beings. Only one to two percent of these devices are date sensitive, but if all of the ones that have not been repaired or replaced go "on strike" at the same time, the damage to the economy could be very large. Because they have not gone "on strike" before, we are not aware of their presence.

NO PRECEDENT OR SUITABLE ANALOGY

People have not been able to think clearly about y2k because we have no precedent or suitable analogy to guide our thinking. Natural disasters – hurricanes, tornadoes, earthquakes, floods, even wars – are always local. The rest of the country or the world remains unaffected and is able to send help and supplies. Furthermore, natural disasters occur at random. There are usually no more than a few that have to be dealt with at one time. Y2k is different in that most y2k failures will happen at almost the same time to everyone in the world.

Most technological disasters are also local in scale. Occasionally bridges or buildings collapse, ships sink, or planes crash. But these events affect a small number of people. Possible examples of technical errors which affected a large number of people were the thalidomide babies and the use of lead pipes in ancient Rome.

Usually failures do not have ripple effects, because they are repaired quickly. However, the longer a disruption lasts, the more serious it is. Unrefrigerated food spoils. Unheated pipes freeze. Inventories of parts are depleted. Businesses which do not operate for a week or more are at high risk of bankruptcy.

Y2k caused some computers to fail before 1/1/2000, but these failures did not receive widespread attention before about 1997 (Levy, 1997). By then it was too late to fix all the affected equipment.

INSTITUTIONAL WEAKNESSES

Y2k did not fit the institutions of modern society. No organizations or divisions of organizations were designed to deal with it, at least not after it ceased to be a purely technical issue. There was no tradition of regularly auditing information technology used by an organization and no legislative requirement to do so.

Software is not guaranteed. License agreements basically say, "buyer beware." This is different from most other products. Until y2k the difference did not seem to be important.

The reviewers of software for technical publications did not check whether a new software product was year 2000 compliant. Usually the concerns were the number of features, ease of use, and speed of operation.

Managers usually had backgrounds in finance, law, or marketing. They did not understand how the technology worked. CEOs did not fully understand the vulnerabilities of firms either to technology or to networks of suppliers and customers.

Universities did not provide a warning about y2k in time to make a difference. With only a few notable exceptions academics have not been involved in y2k work. Universities have not dealt with y2k through the curriculum but rather by endeavoring to fix their own internal equipment. Curricula in engineering and management neither discussed y2k explicitly nor suggested institutional structures which would help society cope with a problem such as y2k. The field of management science has focused on optimization. Squeezing a small cost out of a large process can save a company millions of dollars a year. Creating resilient organizations has not been a central concern.

There was no experienced group to whom y2k could be delegated. Y2k has been delegated to technical departments, legal departments, share-holder relations departments, lobbyists, and public relations departments. But for all these groups, y2k was a new kind of issue. Only top level officials can decide how the organization should respond to a threat of this magnitude, but they also lacked experience and faced many urgent, more familiar tasks.

MISSED OPPORTUNITIES

During the George Bush administration there was a debate between the Pentagon and the National Institute of Standards and Technology over whether to adopt a two digit or a four digit date standard. The Pentagon wanted a two digit standard. NIST recommended a four digit standard. The Pentagon won the battle (Anson, 1998).

Even technology leaders such as Bill Gates did not understand the problem. Originally Gates said that y2k was "not a desktop problem." As late as 1998 Gates said he did not understand why an elevator would use a date. Apparently he did not understand building management systems. As late as the fall of 1999 Microsoft and other companies were issuing upgrades and more compliant versions of their software.

Political leaders in Washington apparently did not want to focus attention on y2k before the 1998 election. And during 1998 the press and the public seemed far more interested in sex scandals than in the possible failure of the nation's and the world's infrastructure.

HUMAN PSYCHOLOGY – FEAR AND DENIAL

Many leaders did not know what to do (Umpleby, 1998). The idea that if they acted, they might stir up public panic served as an excuse to do nothing or at least to limit discussion outside the organization. All institutions acted first to fix their own software and hardware. Working with suppliers and customers came later in the process. Concern about the communities in which workers lived came even later, if at all. This sequence of concerns slowed the diffusion of attention to the problem. Also, before individuals could act rationally, they had to work through their emotions (Kubler-Ross, 1971). This caused delay.

Members of the public and the press, when confronted with an uncertain situation, decided to trust authority figures, rather than investigate the phenomenon themselves. When confronted with conflicting opinions, people seemed to choose the explanation that made them feel comfortable. In addition, people and the press like to see confirming evidence. They react to actual events better than to hypothetical events or scenarios. Some y2k community activists hoped for some very visible failures in 1998 or 1999 which would serve to focus public attention on y2k. There were failures, but no really dramatic or life-threatening failures. Also, in 1999 the President's Council on Y2K Conversion conducted a surprisingly effective campaign to calm fears and reassure the public.

It is possible for people to overlook something important, because their formulation of the task either is not broad enough or simply looks in one direction rather than another. Because y2k was a new kind of problem, there was an on-going struggle to formulate it adequately. The problem of perceiving y2k is complex. Often the problem was not fully understood. Some people thought there would be very mild effects, others envisioned major effects. Sometimes the problem was understood, but people did not want to deal with it. Sometimes people saw the implications and became scared, so they pretended it was not there. Unfortunately, not confronting y2k has delayed action and so made the consequences more serious.

WHO MIGHT HAVE SEEN IT COMING

Many people might have seen y2k coming in time to give an adequate warning – computer scientists, management scientists, software vendors, IT journalists, science fiction writers and futurists or technological forecasters. Perhaps science fiction writers just could not imagine that the simple problem of representing four digit dates by two digits could seriously disrupt modern society. It is surprising that the many forecasts of life in the year 2000 did not reveal the y2k problem. On the one hand everyone connected with computing knew about the year 2000 computer problem, but no one was thinking about it.

In his book *Profiles of the Future* Arthur Clarke (1962) addresses the question of why people are not good at forecasting. He suggests two reasons – failures of nerve and failures of imagination. A failure of nerve is a failure to extrapolate a trend to its logical consequences. A failure of imagination is a failure to invent something that is technologically possible, but not yet present in society. Y2K has been a massive, worldwide failure of nerve – a failure to think a design decision through to its logical conclusions.

REFERENCES

Anson, Robert Sam. "12.31.99: The Y2K Nightmare," *Vanity Fair*, January 1999, pp. 80-144.

Boehm, Barry W. *Software Engineering Economics*. Englewood Cliffs, NJ: Prentice-Hall, 1981.

Brooks, Frederick P. *The Mythical Man-Month: Essays on Software Engineering.* Reading, MA: Addison-Wesley, 1995.

Clarke, Arthur C. *Profiles of the Future*. New York: Harper & Row, 1962.

Levy, Stephen. "The Day the World Crashes," *Newsweek*. June 7, 1997.

Kubler-Ross, Elizabeth. *On Death and Dying.* New York: Macmillan, 1971.

Mueller, Karl. "The Epigenetic Research Program: A Transdisciplinary Approach for the Dynamics of Knowledge, Society and Beyond," Institute for Advanced Studies, Vienna, Austria, Sociological Series # 24, March 1998.

Samuelson, Robert J. "Computer Doomsday?" *The Washington Post*, May 6, 1998.

Umpleby, Stuart A. "A National Action Plan for Y2K Recovery," in Alain Wouters, Philippe Vandenbroeck and Douglass Carmichael (eds.). *The Millennium Bug, The Year 2000 Computer Crisis*. Leuven, Belgium: Acco, 1998.

U.S. Senate, *Investigating the Impact of the Year 2000 Problem*. February 24, 1999. http://www.senate.gov/~y2k

Yourdon, Edward and Jennifer Yourdon. *Time Bomb 2000: What the Year 2000 Computer Crisis Means to You!* Upper Saddle River, NJ: Prentice Hall PTR, 1998.

top of this page