July, 1998

The Millennium Problem and the Marketplace of Ideas: Insights into Freedom, Responsibility, and Technological Development

Scott DeVito
Dennis R. Cooley, North Dakota State University - Main Campus

Available at: https://works.bepress.com/scott_devito/12/
THE MILLENNIUM PROBLEM AND THE MARKETPLACE OF IDEAS: INSIGHTS INTO FREEDOM, RESPONSIBILITY, AND TECHNOLOGICAL DEVELOPMENT

Dennis Cooley and Scott DeVito

I. Introduction

Millennialists believe that the year 2000 represents a pivotal date in the history of the world. They contend that, due to some natural or supernatural force, various disasters and calamities will occur at the turn of this century and that civilization as we know it will cease to exist. Unfortunately, they might be right, but for the wrong reason. As we approach and enter the twenty-first century much of the computer equipment and software in use today will, due to a simple flaw, either produce serious errors or simply fail to work at all. (Call this the Year 2000 problem or simply Y2K.) The cost of repairing the flaw is estimated to be between $300 billion and $600 billion in the U.S. alone. Capers Jones, chairman of Software Productivity Research, believes that the total worldwide cost of the Y2K problem will be on the order of $1.6 trillion. Furthermore, it is likely that the flaw will not be corrected in many key systems by the year 2000, resulting in the failure of many economically important systems and in a potentially multi-billion dollar increase in the overall cost incurred due to the problem.

A close analysis of the Y2K problem provides valuable insight into the ethics of technology. We contend that the primary cause of the Y2K problem is a lack of, what John Stuart Mill calls, originality (and the related loss of freedom, diversity, and understanding) in the information technology (IT) industry. If the various IT companies had supported originality, then they would have empowered their employees to respond to the foreseen difficulties associated with the change of millennium long before these difficulties blossomed into the current worldwide problem. We contend that originality, which requires and nourishes both intellectual
vigor and diversity, which require in turn freedom and variety of situa-
tion, not only develops a better marketplace, but a better world full of
people who have reached their true human potential. It is thus incum-
 bent upon the business world to adopt some form of originality.

In order to understand the moral ramifications of the Millennium prob-
lem, we first explain what the Y2K problem is and why it is a disaster
for the marketplace. Secondly, we briefly sketch out Mill’s argument
for originality. Next, we use Mill’s analysis, along with a Kantian prin-
ciple, to investigate who is to blame for the Millennium problem and
who should pay for repairing the damages. Finally, we show that Mill’s
theory of originality has direct bearing on moral technology for two
reasons. First, it calls for technological diversity and intellectual vigor
in the computer industry. Secondly, Mill’s limitation on diversity in ac-
tion is applicable to a moral limitation on the introduction of products
into the market.

II. THE Y2K DISASTER

The Y2K problem (simply put) is that many programs and the hard-
ware they run on do not represent dates using all the information
necessary to pick out a unique date. This problem comes in two types:
one is mainly a software problem, the other is mainly a hardware prob-
lem (although each problem can occur in either software or hardware).
Let’s call these problems Y2KS and Y2KH respectively.

Y2KS arises because many software programs represent the date
October 5, 1997 as 100597. They simply throw away any information
that picks out what the century is. This procedure is problematic be-
cause October 5, 1997 is represented in the same manner as October 5,
2097—as 100597. So programs that use this approach to date storage
cannot tell the difference between dates across the centuries. Not only
does this date storage format lead to confusion about the date repre-
ented it also leads programs to output incorrect calculations. For
example, imagine that we are calculating someone’s age (perhaps to see
if they are eligible for some government benefit). The person was born
in 1966. This is represented as 66, while this year (1997) is represented
as 97. The person’s age is then calculated to be (present year minus
person’s age is calculated as 00 - 66 = -66. This is impossible, an error,
and depending on how the program was designed, will result in various
responses from the program, including outputting anomalous results or
simply crashing the entire program.

Y2KH poses other problems because it discards different information
about the date than Y2KS systems do. Computer chips use a small
number of bits to store the date. For example, global positioning satellites (GPS) use a ten bit field to represent dates. These are the satellites that allow navigation equipment (on boats, planes, cars, etc.) to determine their location. Since each bit can represent a 0 or a 1, ten bits can represent a number between 0 and 1023. In order to represent dates with a ten bit field, one assumes that the number represented by the date field tells us how many (e.g.) weeks have passed since a particular reference date. The reference date that the GPS system uses is January 5, 1980. So, if the date field has a zero in it, then this must be the week of January 5/6 1980. If the date field has the number 51, then this is the week of January 5, 1981, while the date October 5, 1997 is represented by the number 855. The problem is that because the date field is only 10 bits long, the maximum number it can represent is 1023. So, the date system is good for only 1024 weeks (or about 20 years). After August 15, 1999 we will need a new way of picking out days. The way the Navy intends to fix the problem is to modify all of their hardware and software that relies on the signals from the GPS system to start using a new reference date. So, starting August 22, 1999, a 0 in the date field will refer to the start date August 22, 1999 not January 5, 1980.

This resetting should cause no problem (other than the financial cost) if your hardware or software has been modified to take into account this resetting of the base date. Unfortunately, many systems currently available and in use do not recognize the new base date. [Covault, 1997]

Both versions of the Y2K problem arose because of some (what were believed to be) reasonable choices that were made when computers first began to be used on a widespread scale. In the 1950s and 1960s most computer input was in the form of 80 character punch cards. As it was advisable to fit as much data on the cards as possible (using as few keystrokes as possible), there was a great deal of temptation to abbreviate any field you could. [Brigham, 1997] The year portions of dates were an obvious field to shorten, since you could design your programs to recognize that the year ‘59’ was really the year ‘1959’. This desire to save space also applied to disk storage space. From the 1950s right up until the 1980s computer disk storage space was very expensive. In order to save disk space and money, programmers shortened every field to the smallest size that could accommodate the information that the field contained. Since most programmers and system designers assumed that their programs would not be running in the 21st century, they truncated the year portion of the date field to just two digits. After all, (they thought that) they could reasonably assume that any date that their programs encountered would be a date in the 1900s. This approach to storing dates eventually became part of the culture and dates were simply stored in this fashion.
There are three types of systems that are affected by the Y2K problem: application software, embedded systems, and systems software. In addition to these systems, the very data that the systems use can create Y2K problems. In order to understand the philosophical and market implications of the Y2K problem we explain below what application software and embedded systems are and how they are affected by the Y2K. We focus only on these two types because these are the systems that are (1) most important for the smooth running of the country, (2) most difficult to repair, and (3) most expensive to repair.

A. Application Software

Application software consists of programs that run particular applications. For example, word processing software, payroll software, banking software, air-traffic control software, etc. This is the software that most people encounter. There are literally hundreds of thousands of application programs that contain hundreds of billions of lines of code. Most of these programs refer, in some way, to dates. Some programs use dates as the key of the record, others use dates as conditions for performing various functions (e.g., printing a report, transferring money, turning on the lights, etc.) and other programs use dates in calculations. Dates are an integral part of most of these programs and if the representation of dates is confused then the programs will not function in the way that they ought to function.

In order to understand the scale of this problem we need to look at a few examples. Chase Manhattan bank intends to spend $250 million over the next three years in order to repair its 200 million lines of code. Coopers and Lybrand have estimated that a company with 8000 uncorrected programs (or 12 million lines of code) would be required to fix one out of every 50 lines of code and test almost all of the 12 million lines of code. The government is also severely affected. The IRS expects to move 300 programmers and analysts from their current tasks to help make the IRS’s 100 million plus lines of code Y2K compliant.  

How close are the various sectors of the application code market to completion? U.S. companies are in the best shape. According to JP Morgan Securities Inc., awareness of the problem is high (at all levels) and funding has been found for many Y2K repair projects. Although being in the best shape does not mean that you are in great shape. Peter de Jager claims that as of May 1996 only 35% of all businesses had begun work on correcting the Y2K problem. Capers Jones‘ survey of his company’s 600 clients (of which 250 are Fortune 500 companies and 50 are government agencies) shows that large software vendors, insurance companies, and banks are in the best shape. Ten percent of these companies
THE MILLENNIUM PROBLEM

claim to be ahead of schedule and 60-65% are on schedule. (Twenty-five percent are behind schedule.) But of health care companies only 46% are on schedule and 51% are behind schedule. [Ibid., ff. 2]

The U.S. government is at least one year behind U.S. corporations and has seriously underestimated the cost of repair (at $2.7 billion). The General Accounting Office has estimated $20-30 billion in total Federal costs. (This may be a conservative estimate.) In recent testimony Sally Katzen, administrator Office of Information and Regulatory affairs Office of Management and Budget, testified that six (of 24) agencies have fallen behind their targeted plan for correcting year 2000 problems. This is all the more deadly because 15 of the 24 government agencies have left no time in their schedule to take into account unexpected delays. They expect to complete implementation of their repairs in November and December of 1999. (This is problematic because the information services community has an on-time-delivery track record of less than 20%.) According to Capers Jones, while certain federal departments are on schedule others are falling behind; e.g., defense is behind schedule, while labor, energy, and transportation are really falling behind. [Ibid.]

Matters are worse for state and local governments. The General Accounting office claims that only 25% of state and local governments will be ready by the year 2000. As of May 1997 only 30% of the states were in the testing stage of Y2K repairs, 46% do not have a count of the number of programs that need to be repaired, 25% do not have a completion date, and 45% have not completed a cost estimate. Furthermore, the states are failing to allocate funds from the budget towards addressing the problem and the few states that have allocated funds seem to be allocating inadequate amounts. In the Capers Jones' survey, 45% of state governments say they are on schedule and 52% are behind schedule. In city governments, only 30% are on schedule, while 68% are behind schedule. Finally, the situation is even worse outside of the U.S. Recognition of the problem and implementation of solutions by both foreign businesses and governments is lower than that of US local and State governments.11

What makes this all the more frightening, is that companies today are deeply dependent on other companies. For example, the world-wide-widget company relies on the Spanish-sprocket company for just in time delivery of sprockets. If the Spanish-sprocket company is not Y2K compliant, then when the year 2000 roles around their delivery system may collapse. The result is that even though world-wide-widget is Y2K compliant, it will have to shut down three of its plants for lack of sprockets. In essence, the inter-connectedness of companies today leaves them open to a domino effect from companies that they depend on who may or
may not be Y2K compliant. Since the above mentioned statistics imply that many companies are likely to not be year 2000 compliant and companies are deeply interdependent, it is likely that widespread economic damage will occur at the turn of the century from this domino effect.12

B. Embedded Systems

Application programs are not the only part of the computer industry vulnerable to the Y2K bug; many basic computer hardware systems, called "embedded systems," are also expected to crash at the turn of the century. Embedded systems are pieces of machinery that contain computer chips. This includes your desktop personal computer,13 the mainframe computer that your Bank uses, satellites in space, telephone switching boxes, etc.14 They are called "embedded" systems because they are devices, used to control, monitor, or assist the operation of equipment (or an entire industrial plant), that are integral to the operation of that equipment.15

There are two basic problems with embedded systems. First, in many cases, they are designed to work invisibly and are therefore very hard to spot. So, the management of an industrial plant or power station could think that they have replaced or modified all of its embedded systems, but may actually have missed a few vital systems. The second problem is that in many cases the chips cannot be repaired or replaced. This is due to the short technical life-span of chips (2-3 years). Once a chip has been replaced by a "superior" (e.g., faster) chip, it is nearly impossible to find blank versions of the old chip or someone who can program the chip. In addition, it is frequently impossible to remove the old chip from its circuit board without damaging the circuit board.

III. Moral Technology

This provides only a small part of the types of Y2K problems that are known. In the interests of time, we have omitted a great deal of technical details. However, the information we have provided clearly illustrates the Y2K problem and its wide-ranging implications. The problem is that the technology that our society relies on may fail to function at the turn of the century. This malfunction can and will have serious implications for the economy and security of the United States and the world.

There are a number of philosophical "morals" about technology that can be drawn from this impending problem and its causes. John Stuart Mill's position on originality, though usually associated only with the role of the individual in society, is useful in evaluating the moral implications of the Millennium Problem, as well. Mill defines originality as the combination of intellectual vigor and manifold diversity, which gives
the individual the ability to critically analyze a large variety of ideas in order to discover truths about the world. The lack of originality was a consequence of the computer industry’s attempt to limit freedom and situations. Instead of encouraging the discovery, analysis, and implementation of new and useful ideas, the IT industry forced people to conform to industry standards. We contend that by fostering conformity, the computer industry illicitly harmed individual development, and as a result, society as a whole.

A. Mill’s Originality

Though Mill’s work on originality was intended for the political realm, it can be applied to the business world as well. We believe that if the IT industry, and business as a whole, had adopted a position consistent with Mill’s stance on originality, then the Y2K disaster might well have been avoided—at minimum, the damage from the problem would have been considerably less than the $1.6 trillion predicted. To show why Mill’s political theory is applicable to business, we will briefly sketch out Mill’s position on freedom, intellectual vigor, and manifold diversity, which comprise originality.

In On Liberty, Mill argues that we ought not live by mere conjecture and accepted opinion. People should instead be exposed to a variety of viewpoints, many of which may be in opposition to those commonly held. Only by exposing ourselves to a diversity of opinions can we both discover the truth and have it play a vigorous role in our lives. Furthermore, total freedom from censorship helps lead to the best type of people, viz. those who direct their efforts toward originality. Originality requires people to be capable of creating or evaluating diverse ideas in order to discover the truth, and then live according to these discoveries.

Freedom of thought and discussion in any society are necessary because they allow people to fulfill their obligations to ascertain the truth of statements. Mill gives two reasons for people’s duties to discover the truth. First, each person should try to acquire the wisdom for which human nature suits him. Only by considering other’s opinions and ideas may a person determine which, if any, contains the whole truth or to what extent each is correct so that he may combine the true pieces together into a whole. Once truth is found, wisdom may follow and, of course, make the person a better agent than he otherwise would be. [85-142]

Secondly, not only does the truth make a person wiser, it is useful to him and the world he inhabits. Truth helps him to do what he ought. Mill says that “no belief which is contrary to truth can be really useful.” [91] This, of course, means that false belief not only prevents the person from acquiring the wisdom of which he is capable, it also leads him
to act wrongly. If someone has justified true opinion, then he knows what to do. He can analyze the situation and act appropriately. Furthermore, the more right actions there are, the better the world is as a whole. People, in order to realize their intellectual potential as human beings, must be given free reign to think and discuss opinions as they choose. The alternative is far too costly for it robs the individual of what, in part, makes him human, and causes the world to be worse off than it would have been with a free, clear thinker. Utilitarianism, thus, favors freedom of thought and discussion so that the overall result is much better than it would have been under censorship.

For Mill people must not only have the total freedom of forming true opinions, they also must have freedom, though limited, in acting upon those beliefs. People who do not think and act for themselves are merely "automatons in human form" [127], whose minds are "bowed to the yoke." [129] Eventually, "their human capacities are withered and starved," [129] so much so that they become mere husks of human beings. Thus, in order to reach their full potential, people must have intellectual vigor and manifold diversity, which come from the union of freedom and variety of situation. In other words, they need originality.

Mill maintains that originality is essential to the development and flourishing of each individual. Mill argues that:

the object 'towards which every human being must ceaselessly direct his efforts, and on which especially those who design to influence their fellowmen must ever keep their eyes, is the individuality of power and development'; that for this there are two requisites 'freedom, and variety of situations'; and that from the union of these arise 'individual vigour and manifold diversity', which combine themselves in 'originality.' [125]

Originality comes from a mind accustomed to weighing and evaluating a diversity of ideas, whether such notions are originated by him or someone else. But the mere evaluation is not sufficient for originality. There must be action based on the conclusion reached by the person. He must act and live in the way that is consistent with his individual vigor and manifold diversity.

In fact the lack of originality is extremely disvaluable to each individual and all of society. [101] People, in order to realize their intellectual potential as human beings, must be given free reign to think and discuss opinions as they choose, make informed decisions, and then act on them. The alternative is far too costly for it robs the individual of what, in part, makes him human, and causes the world to be worse off than it would have been with an "original" thinker. Utilitarianism, thus, favors originality so that the overall result is much better than it would have been under censorship.
THE MILLENNIUM PROBLEM

However, Mill does not promulgate a system of absolute freedom to do as one pleases whenever the desire strikes one to do so. Freedom in action, though not in opinion, may be limited if the agent makes "himself a nuisance to other people." [124] Nevertheless, if the action does not entail harm to anyone other than the actor, and perhaps minor harm to others, then it is permissible to do it; the agent only has himself to blame if he does not act in his own best interest. On these grounds, the only moral limitations that can be set on business's freedom and technological diversity are those that prevent severe harm to others.

The application of Mill's position on originality is obvious in the political field, however, the connection to the mundane realm of technology is less obvious, but just as important. Just as people can be victimized by political thought police, so may people be harmed by corporate thought police.

The greatest harm done is to those who are not heretics, and whose whole mental development is cramped and their reason cowed, by the fear of heresy. Who can compute what the world loses in the multitude of promising intellects combined with timid characters...? [101]

The business world also contains intellects whose character is too timid and who are cowed by the demands of their supervisors. As a result, the corporate world is peopled by men and women whose mental development is cramped.

Originality is essential for business because it helps us to discover new truths; point out and explain why commonly held old truths are merely false beliefs; discover and implement new practices; and set examples for others to follow. [132] Without originality we prevent the development of new useful ideas that can lead to safe inventions, which in turn, causes the world to be worse off than it otherwise would have been.

If the IT industry and the rest of the business world had followed Mill's advice—if they had not encouraged conformity, custom, and economies of scale—then we might have been able to avoid the Millennium problem in two different ways. First, the shortcomings of present software and hardware would have been foreseen and dealt with in a timely manner, instead of creating the present disastrous situation. More importantly, by encouraging originality, only the best technology would have entered the marketplace in the first place, rather than the severely defective products which currently threaten the financial health of many innocent people. No good thinker would ever introduce a product that is not worth the effort.

Finally, the most tragic result of the business world's drive for conformity is not the $1.6 trillion Y2K problem. The loss of people who would have been able to evaluate ideas, determine the truth, and act on
it is a huge cost to the world as a whole. Who can imagine what advancements in science, technology, and other fields were prevented by the coercion from business’s short term goals mindset? This is the true cost of disregarding Mill’s advice on fostering originality and one that we can ill afford.

IV. Who Is Morally Responsible for the Y2K Disaster?

Attributing responsibility for the Y2K disaster to the appropriate parties, and the moral blame that comes attached to it, is extremely difficult because the problem involves so many people and the various actions they performed. Though an arduous task, with enough patience and care we can untangle the different causal strands and trace the relevant ones back to those who bear moral liability for introducing dangerous technology into an unsuspecting marketplace.17

Before we start the process of attributing moral blame for the Millennium Problem, we need to address a response that the IT industry might make to criticism of its behavior in this matter. This response is not merely intended to mitigate the computer industry’s responsibility for the problem, but to eliminate it completely. The IT industry seeks to prove, using a utilitarian argument, that it could not morally have used a 4 digit years rather than a 2 digit year; therefore, they are not to blame for the Y2K problem because a person cannot be blameworthy if he has done the right thing.

The primary justification the computer industry uses for it actions is the extremely high cost of memory in the 1960s through mid 1970s. “Adding two century digits to the date field for a 100 million record file would have added at least 100 megabytes of storage requirement to a disk that cost upwards of $20,000 for 10-20 megabytes. It simply made economic sense to omit the two century dates.” [Hoyt, 1] This line of defense is effective if the cost of memory remained constant and there was no alternative, efficacious method for storing dates. No one can justifiably deny that the loss to the computer industry and society as a whole would have been enormous if computer technology was smothered at this early stage by the onerous burden of buying extremely expensive, unnecessary memory. Many of the advancements in technology would have been stifled because the high cost of storage would have prevented companies from investing in computer equipment or upgrades. With the lack of financial resources, the IT industry would have been unable to do all the research and development that led to some of the remarkable breakthroughs in technology, such as the Internet, home computers, automated tellers, etc. On these grounds, the
IT industry argues that, overall, the world would have been worse off than it currently is if they had used 4 digit years instead of 2 digit years.

The computer industry's primary justification is however not as strong as they would have us believe. First, the cost of the various types of memory, though admitted expensive early in the computer age, decreased dramatically in the mid 1970s onward. "The price index for memory chips [for instance] declined at a 37-percent average annual rate from 1975 to 1985 and at a 20-percent average annual rate from 1985 to 1996." [Grimm, 1] Prices for dynamic random access memory (DRAM) chips fell from $52.5 per 16 kilobit DRAM in 1976 to $4.81 in 1980 and $1.05 by 1983. (Ibid., Table 1) Prices per DRAM kilobit dropped from $1.81 in 1975 for 4 kilobits to $0.0103 in 1990 for 4 megabits, and finally $0.0030 in 1995 for 16 megabits. (Ibid., Table 2) The prices of other types of memory also fell precipitously during this time period. The lower prices of memory removed the onerous cost from the computer industry that implementing 4 digit dates would have caused them in previous years, thereby rendering any appeal to utilitarian considerations illegitimate. When it became more cost effective to increase the size of storage in order to handle larger date fields than to continue using two digit fields which ultimately lead to the Millennium Problem, then the industry was obligated to do so to maximize utility. According to utilitarianism, the computer industry could have started implementing a new standard for storing dates, but negligently did not do so.

The computer industry's primary justification is not effective even prior to the mid 1970s. Though it may seem to present an overwhelming consideration in favor of two digit dates, the high cost of memory is not sufficient to justify, on utilitarian grounds, this type of date storage. In 1986, B. G. Ohms described the Lilian date format. The Lilian format allows dates—up to the year 10,000—to be stored in three bytes of memory. A variant on this format allows dates between January 1, 1901 and December 31, 2079 (or any 179 year period) to be stored in only two bytes of memory. [Ohms, 249] Using the Lilian date format would require that a conversion utility be developed that would allow us to transform Lilian dates into (e.g.) Gregorian dates and vice versa. Ohms provides the algorithms for such a program, for the three byte Lilian date, that require less than 100 lines of machine language instructions (IBM System 360/370) and 4K bytes of storage. An extra benefit was that Ohm's program worked with many of the current programming languages; so, additional conversion programs and memory were not needed. Thus, a more efficient system of date storage was possible that required a relatively small amount of additional memory. The extra benefits of having four digit dates would have clearly made up for the minor
additional cost of memory to store and use the conversion program. The Lilian system, or one similar to it, would have avoided the Millennium Problem, while still allowing companies to spend resources to help produce the technology we currently enjoy. Hence, we could have had a much better world than the one that we now live in with its severely flawed technology.

Using the Lilian system as a standard would have been a classic win-win situation, if it had been created and implemented sooner. Here we see the danger of blind conformity to the standards, practices and ideas of the dominant group (a problem that we find at the core of the Y2K problem). By conforming to the Gregorian calendar and not trying to discover an alternative way of storing data efficiently, the seeds of the Millennium Problem were needlessly sown early in the history of the computer industry. A better solution was waiting to be found and implemented, if the industry had encouraged it. Unfortunately, as we all know, this did not happen. Utility was not maximized due to a lack of diversity.

Having dispatched the IT industry’s potential defense of their actions, we can turn to the primary task of this section—attributing moral responsibility to the various members of the IT industry. In the remainder of this section, we first concentrate on the individual contributor’s responsibilities to have avoided the Y2K problem, and then focus on the IT industry’s responsibility for the Y2K problem. By focusing the section in this way, we can draw the general conclusion that the computer industry as a whole bears a great deal of the blame for fostering conformity instead of originality and diversity in the market. Certain consumers, including the federal government, are guilty for the same reasons. Many businesses became liable for the current state of affairs by coercing IT companies to plan only for the short term, as they themselves did, rather than creating long term goals. We will also take to task the individuals who neglected their duties to broaden their minds with new ideas and opinions, so that they could become much better people and employees.

While determining individual moral culpability for the Y2K problem, we must always keep in mind that there will be degrees of responsibility for the participants involved dependent on their particular positions and actions in the computer industry. We have to ascertain how much actual free control the particular individual had over creating the problem, whether or not he had an obligation to foresee the impact his product would have on the market, and what his duties were in preventing foreseeably dangerous technology from being sold or implemented. Based on these three criteria, an agent who had a great deal of control, was obligated to determine what the long term results from the introduction of this technology where, and who had a duty not to allow dangerous technology to enter the
market will have more blame assigned to him than someone, for example, who was merely a cog in the machine.

For the purposes of better attributing moral responsibility, a correlation among the three criteria should be noted. If the answer to one criterion is known, then the answer to the other two can be quickly determined. The more control an agent has, for instance, the greater the obligations to foresee the result of the technology and keep it off the market. On the other hand, the weaker the control, the weaker the obligations are on the part of the individual.18

There are many different and compelling reasons to attribute blame to the various people responsible for the Millennium problem, but the sheer shortsightedness of all parties concerned is the primary justification for holding each of them liable. At every level of the computer industry, from lowly programmers to the lofty heights of computer firms’ CEOs and their customers, we see a surprising lack of foresight in assessing real world situations. Even though they should have seen the problem and solved it,19 IT industry members violated their obligations to prevent avoidable harm to others. Virtually all agents failed to realize that their technology was hopelessly unable to deal with the coming millennium change. Each person did little to nothing to prevent an avoidable economic disaster that has and will hurt many innocent victims, in addition to themselves.20

What is even more disturbing than the lack of adequate foresight is the arrogant attitude that many members of the IT industry exhibit toward the problem. Peter De Jager, has found a typical reaction to a discussion of the Y2K problem to be “Snickers and comments such as, ‘I won’t be in this position or company in the year 2000. It’s not my problem.’”[De Jager, 108] The fact that they are responsible for this disaster, and can avoid it only if they take the extreme measure of leaving Earth before the turn of the century, plays no role in the internal evaluation of their actions. They callously just don’t care.21

The best method for attributing individual blame is to start at the lowest level of the computer industry and work our way up to the top. By proceeding in this manner, it will be clearer how degrees of responsibility attach themselves to the various actions and people involved. On that note, let us begin with those who actually created the software, the programmers. (These arguments, for the most part, work easily as well against the hardware engineers.)

All programmers are morally liable for the Y2K problem, in part, because they did not bother to take into account that by making date variables only capable of storing two digits they were forcing their technology to become obsolete within a maximum time span of thirty years. Since it was easier to enter data with two digit dates, and cheaper to
limit the amount of memory required, they did what was in the short term best interests of their employer's bottom lines and the worst for the long term financial health of the business world. When programmers learned of the problem, they should have instituted a change in their programming practices.

The mere fact, however, that computer programmers were unethically lax in their duties does not entail that each are equally responsible for the Y2K problem, for, even within the class of programmers, there are subgroups. To attribute blame correctly, we must recognize that computer programmers can be of two sorts: those who work internally for a non-computer corporation or those who earn their living by creating programs for their companies to sell to others. The type of programmer an employee is, in part, determines how much responsibility she has for the Y2K problem. If the programmer works in-house, then the amount of blame she must shoulder is significantly greater than those of the other sort because she had a greater duty of foresight than her external counterpart.

There is a duty for workers in any technological field to be aware of new developments in their area of expertise. Employment is basically a contract between the employer and the employee in which in exchange for doing his job, the employee will receive some form of compensation. Since his employer, in effect, contracts with him to do specific type of job with relevant attendant duties, the agent must live up to the terms of the agreement. Part of every employment agreement is that the employee will do the best job he can for his company, within reasonable limitations. On these grounds, for instance, it is reasonable to expect programmers to be aware of the American National Standards Institute (ANSI) date standard which recommended 4 digit dates but allowed 2 digit dates as optional. When the standard was modified in 1988 to highly recommend 4 digit dates, each programmer should have been aware of the new standard.

Mill's marketplace of ideas requires an agent in the technological field to study the new technological breakthroughs, especially if they will have an impact on his employer and him. Programmers, for example, must acquaint themselves with trends and discoveries in the programming industry. They must read trade journals and expand their skills so that they may serve their employers in the best way possible. The best way, of course, is to inform the employer of new opportunities, warn of imposing threats, and implement new technology, provided the employer agrees to the latter. As the employee helps his company achieve its goals, he also acquires knowledge, thereby improving himself.
Even if the technology appears to be infallible, each person in the field still must think about whether or not it truly is. As Mill contends no person is infallible, and should always question his assumptions to determine if they still hold. [1972, 89] To illustrate his point, Mill says that even the almost certain Newtonian physics should be examined to see if it really captured the truth about the universe. [Ibid.]

In business and technological fields, to do less than maintain this standard of constantly evaluating one's most important beliefs would be malpractice. It would be along the lines of a doctor continuing to drill holes in the heads of patients to let out the evil spirits trapped inside, after it was determined that mental illness was due to a chemical imbalance in the brain. The archaic physician violates his contractual oath of doing no harm by using discredited methods.

Though not entirely escaping responsibility, external programmers who created software to be sold to others are less blameworthy than internal programmers. The former's duties lie less in keeping up with the new technology, and more in an obligation to inform the unwary of wrongdoing by their employers. The reason for exempting them from the former requirement is because their employers have the responsibility to keep them well trained in order for their product to be the best. First, it is in the company's best interests to make their workers as efficient as possible to enhance the company's competitiveness. Obviously, those employees with the greatest intellectual vigor outperform those without. Secondly, the computer industry has a responsibility—as we will see in a later section—to help computer illiterate customers get the best product they can afford. To do less would be to illicitly take advantage of the ignorant.

However, external programmers still must try to guarantee that their product will serve the customer's needs and not harm them unjustly. Though to do so is admittedly risky to the employee's position, they are morally bound to warn customers of dangerous technology that is being sold to them. If her employers are trying to take advantage of computer-naïve customers, then the worker has the same moral obligation based on the respect for persons and avoidance of unnecessary harm to blow the whistle that an employee of, for example, a rapacious funeral home has to inform the deceased's relatives of inflated prices.

The fact that the computer-ignorant customer comes to the computer industry and places its trust in the latter's hands, creates a duty based on Due Care for the industry to not exploit the customer. The buyer legitimately expects the seller to represent his interests, at least to the extent of not harming the buyer more than it helps it. As reflected, in part, by laws governing the free exchange system, there is an implicit
contract in the market that the seller will not attempt to knowingly allow the buyer to come to harm as a result of using the seller's product. If this contract were not present, then the free exchange system would be impossible, even in the modified form that capitalist countries use. By taking advantage of the consumer's ignorance, the seller has made it impossible for the transaction between the two to actually be a free exchange of goods, in which the buyer and seller necessarily have adequate knowledge of the relevant information of the product and conditions of the sale contract.

Certain programmers bear moral responsibility on another count as well. The incredible sloth of some programmers is reprehensible. Instead of creating code that was easy to understand and modify, quite a few programmers wrote code that is virtually impossible to decipher by anyone other than themselves. For example, rather than using the variable "YEAR," they would use "XX" instead. To make matters worse they failed to make comments on what the variable represented!

Gary H. Anthes describes a situation in 1972 in which he had to deal with a Naval program that would go into endless loops rather than send out secret messages containing coordinates for missile targets in the former Soviet Union. He admits to just patching the program, without bothering to document the repair, so that the system would work for the short term. Instead of being concerned about the possibility of further problems, Anthes says neither he nor his supervisor were "going to worry about an event that might occur long after we both had left the Navy." [1996, 78]

This negligence has created a situation in which, instead of merely making relatively minor alterations, programmers have to work through code line by line to determine not only date changes, but what some of the variables do as well. If it is not immediately apparent what a variable's function is, the new programmer must test it to determine what it actually represents; thereby adding greater costs to the company trying to become Y2K compliant.

The people and firms that sold the software and hardware technology to others shoulder even more blame than the programmers. After all, the programmers could make the somewhat sympathetic argument that they were merely cogs in the big machine. The ones in charge of the machinery, since they had far greater control than the programmer, however, do not have that luxury.

Computer companies failed in their duties toward their customers by not finding a better way to store dates than the two digit method or installing devices—either soft or hard—that would allow easy upgrades if memory ever became cheaper. One is reminded of the Ford Pinto case in which short term utility was mistaken for that of long term, much to
the detriment of Ford.26 The same principle applies here. Though they may have saved their consumers some immediate financial outlays and made a greater profit for themselves, the computer companies now have customers unhappy, at best, with them and a society that is questioning their large profits, especially since the taxpayer will have to foot part of the bill to correct the Millennium Problem.

The computer industry’s utilitarian calculus was based on the mistaken supposition that its customers would update their software and hardware before the manner in which dates were stored became problematic. The justification for the assumption seems to have been either one of two reasons—or a combination of both. First, the industry thought that since it would replace old technology with new as soon as it became available, their customers would act likewise. After all, the reasoning is, no one would want old programs and hardware when new ones had new and improved features.

Secondly, the industry assumed that the new technology would be vital to business competition. A company with better technology would be more competitive than one without. Hence, there would be a race to replace old software and hardware with new products in order to capture larger market share or, at bare minimum, maintain whatever share the company already had.

On both counts, the computer industry was mistaken. Corporations did not upgrade their old technology, which the industry thought obsolete. There was no real reason to do so; the market did not require many businesses to install the latest gadget to maintain or gain market share. Since the costs of replacing old software and hardware were high with seemingly few benefits, businesses forewent replacement expenses. The capital that would have been dedicated to upgrading was spent on items that appeared more promising to improve the company’s competitiveness and financial health; thus, helping guarantee that the millennium change would be a problem.

Though the computer industry does not appear to have intentionally caused the Y2K problem by assuming upgrade, they were still negligent in their duties. As all prudent agents know, the mere fact that someone is expected to do something does not entail that he will do it. The computer industry should have put themselves in the position of the consumer. If they had, they would have easily realized that if a business can get away without spending money on something it does not perceive it needs, then it will not do so. To gain this insight, all the computer industry had to do was look at how it operates. The IT industry does not purchase seemingly needless things; so, there is no reason to think any other business will have different attitudes. Thinking otherwise is illicitly blinding oneself to the obvious, and therefore, unethically negligent.
There is also the problem of the betrayal of the trust the country and businesses put in the computer industry. Everyone was led to believe that an increased reliance on computers would make the world a better (more efficient) place in which to live. Computers were supposed to help us. Instead, it seems they will severely harm each and every one of us in ways that we could not have imagined.

What makes matters even worse is that the computer industry seemed to allow this precarious situation to happen in the pursuit of their self-aggrandizement. [De Jager, 105] After all, we may be able forgive a simple mistake or oversight when made while acting in another’s interest, but to be harmed in the avaricious quest for Mammon is unacceptable. We can never condone, much less accept, people being illicitly hurt for mere financial gains by another.27

Since IT managers have the charge of making the company profitable in the long term and serving their customers as best they can, they should have instituted a practice of programming or installing technology that was capable of handling a four digit year. By focusing only on short term profit and conformity, they shortchanged not only their customers, but their stockholders and employees as well.

Finally, a certain amount of the moral responsibility, and therefore blame, must adhere to those who purchased the equipment or software. It is always the consumers’ duty to ensure that he knows the powers and limitations of those things he purchases. Again, each business manager must, within reasonable grounds, ensure that the business he serves is best served by the things he does or the items he buys. The purchaser must, therefore, acquaint himself with sufficient detail of each alternative product, so that he can make the best long term decision for his corporation. The only way of doing this is by analyzing a manifold of diverse ideas and opinions so that the truth may be found. If the consumer did not guarantee that his electronic purchases would be Y2K ready, when he could do so, he has failed in his duty to serve his company well.

We do not wish to imply that the consumer has to be an expert in these matters. After all, that is why he went to the software/hardware seller in the first place. However, he has to take up the burden that any reasonable person has of learning as much as he can about things he may purchase, within the boundaries of his job description, so that he can serve the long range goals of his employer.

To help guarantee that his company receives the equipment that best serves its needs, he must also ask the right questions, such as, “Are there any serious defects with this technology that I should know about? If not, would you put that in writing?” The seller would then be forced
to tell the potential buyer of problems or be held legally liable for damages, thereby giving the potential buyer a guarantee of a sort.

In order to fulfill his duties, an employee looking to buy a computer network for his corporation, for instance, must examine all the products on the market to determine which one best fits the needs of his business. If he fails in this obligation, then he betrays his company by needlessly putting it at a competitive disadvantage. He has violated the implicit work agreement to perform the tasks for which he is paid.

The mere fact that the consumer has been misled by the computer supplier does not entail that he has no duty to rectify the problem as soon as feasible. Sometimes the circumstances of a situation, even though we may not be responsible for them, require us to do things that we may not want to do. Sometimes, we have to resolve problems not of our own devising because we have duties to ourselves and others that are not abrogated by others creating situations unfavorable to us.

The managers of innocent businesses inflicted by the Millennium Problem have to make hard decisions about how their companies will deal with the issue. Though the companies had no hand in creating the situation, they are obligated to take effective measures now to try to stave off the worst consequences of the millennium change. Managers have to do what would best serve their companies, employees, and stockholders; they do not have the luxury of wringing their hands and weeping over the wrong done to them.

If a manager, once informed of the problem that his firm faces, does not immediately take measures to solve it, then he is guilty of business malpractice. Each manager is charged with doing what is best for the company; that is her job. She must make sure that her company can handle the change of the millennium by calling in experts to update the software and hardware.

An owner of a software company was recently interviewed on C/NET about how business executives were dealing with Y2K. He said that, four years ago, he had personally told a business executive that his company's software needed to be updated. The price at that time was $17,000. The executive refused on the grounds of costs, even though he knew that as the millennium neared more firms would be trying to update their computer services. When the executive contacted the owner this year to have the necessary updates done, the cost had risen to $170,000.

In this case the executive is guilty of malpractice. Though the initial cost was high, it was obviously necessary for the financial health of his firm. By waiting in the vain hope that the problem would resolve itself at no cost to him, the executive forced his company to pay ten times what it should have. There is no adequate excuse for this expensive inaction.
The fact that so many corporate managers, as the one above, were too focused on short, rather than long term goals for their companies contributed in large part to the Y2K problem. Instead of fostering diversity and flexibility in their technology, CEOs and others pressured the computer industry to produce software and hardware as quickly and cheaply as possible. When memory storage was expensive, companies did not want to pay for anything they considered to be nonessential. For example, the federal government required that all dates be stored using two digits data fields because four digit data fields would have required 100 megabytes more memory at a cost of $20,000 per 15-20 megabytes or $100,000-$133,333 total. [Walsh, 43]

This lack of foresight on the part of business managers driven by the relentless calls for greater profits by stockholders set up a dilemma for the IT industry. Should they try to force their customers to buy the Y2K compliant technology they obviously did not want, or should they merely provide their customers the goods they truly wanted? The decision of course was to give in to the customers' demands.

Though not wholly justified, the choice is understandable to us even today. A business that does not provide the products the potential buyer wants will not sell anything, especially if there is another producer who will do as the customer desires. Of course, without sales, a company must go out of business. Given this fact about the competitive market, in order to survive, it was in the computer industry's interests to do as they were told. Thus, the coercion that the IT industry suffered somewhat mitigates its responsibility for the Y2K debacle, while at the same time, shifting part of the blame onto the shoulders of the short sighted business and government agencies that also had a hand in creating it. Once again, when originality and freedom are illicitly limited, many suffer in the race for self-aggrandizement.

V. WHO SHOULD BE HELD LIABLE?

Who should be required to help defray the costs of cleaning up the Y2K problem? Since the problem is so far reaching and dangerous to the well being of all Americans, we believe that everyone, especially those who made a great deal of money from the suppression of originality and diversity, will have to pay for the solution. The situation is too dire to wait for those responsible for the problem to effectively deal with it. We also contend that the federal government should be the primary leader in the efforts to solve the problem as quickly as possible; partially because it helped cause the problem but mostly because it is the only entity with the means and ability to do it.
THE MILLENNIUM PROBLEM

The initial intuitive answer to the question of who should pay for wrongdoing is the people responsible for acting impermissibly. Retributive justice demands that the offender be punished in exactly the right amount to eliminate the burden he bears for the wrongdoing. However, in situations such as these, that solution may be morally unwise as well as impractical to implement. If those responsible parties who have made money from their illicit deeds were to be held fully liable for their actions, then the economy as a whole would be adversely affected. Recall that the cost for repairing the problem has been estimated at $1.6 trillion worldwide. The shock of so many people losing all of their assets would start a wave of panic which could only have a negative impact on financial institutions and other businesses.

A severely damaged economy causes unemployment to rise sharply, which in turn causes even greater pain and suffering. As the economic shock waves continue to move outward and set off secondary waves, more and more people are adversely affected. For instance, banks, their shareholders, employees, and depositors suffer when the newly unemployed cannot make payments on their debts. The harm done to banks and others causes additional harm, e.g., interest rates and bank fees increase, fewer loans are made, etc. The crises only continues to build and extend its reach further into the economy. In the end, many innocent people would be harmed unjustly merely for the sake of revenge.

We are not proposing that those responsible for the Y2K problem be let go without having to pay anything for their misdeeds. Far from it. Rather, when deciding how much they should pay, we must always be sensitive to the impact on innocent people and the economy. It is right to punish the guilty, but it is not morally worthwhile to do so if it leads to an even greater injustice being perpetrated by us.

Before determining who should pay for their roles in the Y2K problem and why, it is important to note that there are three morally different types of situations in which the problem arises. The first is for those corporations that failed to have their in-house software and hardware engineers make millennium friendly technology. Basically, they created defective computer technology that they only use themselves.

The second situation consists of those companies that purchased technology from a negligent vendor. In a large number of these cases, the buyer usually did not know that he was purchasing defective technology, nor did he have the necessary expertise to fully understand any problems he might encounter. The main entity responsible for the problem in this situation is the corporation that created and sold the defective goods.

Finally, there are those instances in which the vendor and customer each bear a portion of the responsibility: the buyer because it demanded
defective products and the vendor because it gave the customer what it wanted. Each, in effect, were co-conspirators.

The differences in the situations justifies treating the companies differently. A company that harmed itself should not be able to recover damages from the same sources that the duped company can. Capital to deal with the problem must, in part, come from the source of the problem. In the first and third situations, the origin of the problem is wholly the companies themselves; in the second, it is the company itself and the corporation(s) that manufactured and sold the defective technology. In each case, the majority of the compensation should, if fiscally and morally practical, come from the companies that produced and distributed the dangerous software and technology.

The amount of recompense should be determined in part by the assets and earning power of the culpable party. A rich person, such as Bill Gates with his $36 billion, must be required to pay much more than a retired programmer on a pension. The CEOs and managers have profited from their misdeeds more than the lower echelons; thus, they have a greater responsibility to compensate those that they have harmed. On the same line of reasoning, large computer corporations will also have to pay more than small ones; they can usually better afford it because they benefited more than the smaller companies.

The financially well-off should also pay more if they had greater control in making decisions that led to the Millennium Problem. Bill Gates and others profited hugely by requiring programming for the short term. Thus, while they boosted their companies' bottom lines they created an economic time bomb of virtually unprecedented proportions. Though admittedly coerced in part by their clients, because they still chose to create the situation that led to this predicament, they must be required to help get us out of it.

Determination of blame apportionment, and thus compensatory obligations, will not be that difficult to do. If a company did not produce any dangerous technology, then of course it will not be required to pay anything. It did nothing wrong.

On the other hand, those companies that produced defective hardware and software are liable for damages. Even though the different companies have used the technology to their advantage—that is they implemented it—they should still receive full compensation for any work done on it to eliminate the Y2K bug.

The moral justification for requiring compensation from wrongdoers in these circumstances is analogous to the case of a person buying a new car, and then finding that the brakes are badly worn after minor use. The dealer and car manufacturer are obligated to replace the brakes
THE MILLENNIUM PROBLEM

free of charge because they were defective in the first place, even though the man has received some benefit from them. To require him to pay for new brakes or what he used would be asking him to pay twice for one thing. He agreed to non-defective brakes in his contract, if he does not receive them, the company has the duty to compensate him in an appropriate way.

The same sort of obligation applies to the Y2K problem. Though they might have received benefits from using the defective technology, the entity that bought it did not buy it as defective. The buyer purchased what he thought was non-defective technology; that was what he agreed to. Thus, in order to fulfill the contract, the seller must repair his product, though he need not do anything more than merely set it right. Anything else would be a supererogatory act on the part of the seller. Unlike cars, clothing, food, and other things that wear out, technological products such as programs do not have a limited service life. As long as the hardware can support them, programs should continue to run forever.

Lest we forget, those companies that used undue market force on the IT industry must also shoulder some of the compensatory and retributive damages. To the extent that the consumers injured themselves or others, the computer industry’s burden is lessened. After all, each agent must pay for whatever damages it negligently caused. To require the software and hardware companies to pay for injuries for which they are not responsible would violate both compensatory and retributive justice.

The problem, of course, is to determine who must pay what amounts for a particular injury. Business, the IT industry, and government were not acting as concerted agents intentionally creating a tort. The Y2K problem came about due to the almost incredible negligence of each. Because they did not think through the ramifications of what they were doing by requiring or providing non-Y2K compliant technology, they created a situation in which a disaster could occur. The difficulty stems from how the negligence of each party contributed to the tort. This is something that perhaps only the courts, and probably not even they, can decide.80

Since the Y2K problem is so large and far-reaching, and the computer industry and business world appear unable to adequately resolve it, the government must take a leading role in the solution. First, the government is the only one who can compel those responsible for the Millennium Problem to pay their full share of compensation and punitive damages. Since the responsible parties are scattered across the country, and unwilling to foot their share of the bill, there has to be some agency to coerce them into doing their duty. The federal government is the only entity with the authority and power to do so.81
Of further consideration is that the federal government's infrastructure for dealing with disasters is already in place. There is, therefore, no need to find someone else to take over the task of collecting money. The government has a great deal of experience that others do not have in this area; thus, it can act more quickly and efficiently than a new agency that is just now determining how to proceed.

Second, since the amount of capital required to fix the Y2K problem is so massive and time so limited, the government is the only source that can produce adequate funds quickly enough to deal with the difficulty. The facts of the matter are that the Millennium Problem must be taken care of within the next two years and it will cost well over $600 billion to do so. Even if those responsible were drained of assets it is unlikely that there would ever be a sufficient amount of capital raised quickly enough. Thus, the $1.2 trillion business of federal government is the ideal place to go for the initial funds to begin work repairing defective software and equipment.

Third, like the bail outs for the Saving and Loans Scandal, the money spent on solving the Y2K problem will be beneficial to a significant number of American citizens. The problems that Americans potentially face due to the Y2K problem, including a large recession/depression, loss of global positioning systems, defense capabilities, etc. are too great to be ignored by the federal government. While the price tag for the bailout is very large, the benefits to all citizens of this country will far outweigh it.

Also, since the government felt obligated to pay for the clean up of the saving and loan debacle, in part, because of the threat to the financial security of American depositors, then it has even greater reason to assist in solving the Y2K problem. The saving and loan scandal, though it cost approximately $200 billion to fix, affected relatively few people when compared to the Millennium Problem. Since the latter will affect banking, defense, and other vital areas, every person in this country will feel its impact. It is government's duty in this case to stave off the worse effects and protect its citizens from as much harm as it can.

Every rational market system, except for the most laissez-faire, recognizes government's role in protecting its citizens from unnecessary harm. In the American system, roughly, the reason that we have a representative government, is for elected officials to represent our interests. Obviously, it is never in the interests of anyone to be unnecessarily harmed. It quickly follows from the above premises that one of government's main duties is to protect us from unnecessary harm. By dealing with the Y2K problem now, government will fulfill at least one of its obligations.
Finally, we cannot forget that the government helped create the Y2K problem by its insistence on the two digit data field for storing years. If the government had been wiser—if it had tried to evaluate technology in light of a diversity of opinions—the Millennium problem could have been reduced, though probably not eliminated. The government's contribution thereby justifies requiring payment from it. To do less is unjust.

VI. The Need for Diversity

Determining who is at fault and why they are at fault is the first step in establishing the importance of a freedom-based ethic of technology. We have seen that the amount of one's obligation to prevent harm and responsibility for harm incurred is proportional to the degree of freedom one has and upon the Kantian principle that one has a duty to not treat people as mere means. But this is not sufficient to establish that the IT industry has an obligation to foster originality, freedom, and diversity in the workplace. We need to add that freedom, or its analogue diversity, in the design of technology is good and that genuine diversity (the kind that is supported by originality and freedom) is lacking in the IT community.

At first glance there seems to be a great deal of diversity in the computer industry. There are numerous computer languages such as COBOL, C++, Pascal, APL, various Assembly languages, FORTRAN, JAVA, and so on. In addition, there are a great number of computer operating systems. These include OS/2 warp, VMS, Microsoft Windows, Unix, etc. Finally, there are numerous computer platforms; including IBM PCs and clones, the various Macintosh computers, Silicon Graphics Workstations, Unisys machines, AS 400s, etc. There certainly is a great deal of diversity in "brands" of information technology and freedom to choose among these various brands. But, that diversity and freedom of choice belies an underlying, and more important, lack of freedom and diversity.

Each of these languages allows programmers to write years using as many digits as they like. Yet programmers do not. Similarly, there is no reason why the chips that make up the various computer platforms or the operating system software that runs on those chips cannot refer to dates in a way that maintains reference to the century. So why do programmers and systems designers use dates that do not maintain reference to the century?

As we described earlier, the problem began innocently enough. In the early 1950s and 1960s saving space on punch cards and in magnetic disk storage was advisable. In order to save money, the first two digits
were removed from the year and it was assumed that dates would be dates in the 20th century. Some programmers recognized that this could be a problem at the turn of the century, but believed that their programs would not be in use at that time. As a result of this lack of foresight, millions (if not billions) of lines of code and billions of computer records were created that took dates to have two digit years. So whenever a new computer program was written, it had to be compatible with these old programs and with all of these old records. This built up a sort of inertia that was difficult, and in many cases impossible, to overcome. The result was that representing years with two digits became part of the IT culture.

If a systems designer (of chips, programs, or operating systems) wanted to modify the IT system so that it incorporated reference to the century, he would face serious pressure not to do so. First, there would be the weight of all of those other programs and systems that did not do so. (Your Y2K compliant program must be compatible with all of those non-Y2K compliant programs and systems.) In addition to this weight, there is a trend towards standardization of programming and systems design in the computer industry.32 For example, there are innumerable books that talk about "structured" programming.33 Students on a massive scale are taught to obey the tenets of these books. Furthermore, companies now hire programmers to oversee other programmers to make sure that they obey the company's programming standards. These programmer watchdogs are required, for example, to pressure programmers to eliminate the use of "goto" type statements in favor of statements that call functions and to eliminate the use of comments in the body of the program. The result is a social structure that requires one to follow the company line. This social structure limits freedom and diversity because it condemns certain practices (having goto's instead of nested functions) and endorses other practices (representing October 5, 1997 as 100597). When a programmer attempts to step outside of the bounds of that structure, regardless of the merits of her doing so, various mechanisms are called into play that "snap" the programmer back into line. It is this lack of freedom that has lead us to the Y2K problem. As Mill stated "The despotism of custom is everywhere the standing hindrance to human advancement, being in unceasing antagonism to that disposition to aim at something better than customary, which is called . . . the spirit of liberty, or that of progress or improvement." [138]

So the type of diversity that is missing (or that is at least in short supply) in the IT industry is a diversity of accepted design practices. You cannot simply write your computer code, program your system, or design your chip in a way that seems right to you (assuming that the
code does the job it is intended to); instead, your program, system, and chip must conform to certain socially accepted (and implemented) forms. In essence, the paucity of diversity in systems design is a consequence of a lack of freedom of choice in designing one's program, chip, or operating system.

The primary reason given for such lack of diversity has to do with economies of scale. In industry, once a uniform structure is in place, it is claimed that it makes economic sense to focus on that structure and design future work so that it complies with that structure. We see this in the development of the car, the "QWERTY" keyboard, and the computer.

The economic advantage of economies of scale in the computer industry is that if everyone uses the same standard, then it will be easier, faster, and cheaper to develop and maintain new software and hardware. Having dates represented in different ways leads to additional work in two ways. First, programmers and systems designers will need to remember that the dates are different in different programs and chips, how they differ, and where they are different. This requires extra documentation and effort on the part of systems designers and programmers. In addition, keeping track of all of these differences will add to the complexity of computer system design and hence to a greater chance of error. If we forget that the one file has a six digit date while another file (that relies on the first) has an eight digit date, then we will have numerous bugs in our programs. Secondly, when dates differ additional code will have to be written because you can not simple move one date to the other nor can you simply compare them (to see which is the earlier date). When dates are in the same format, moving or comparing one to the other is trivial.

The result, the proponent of economies of scale would claim, is that it is very expensive to lack an underlying standard for programs. Having diversity would cost money because it would require additional lines of code and additional documentation, and increase complexity and the chance of making mistakes. These objections, once taken together, imply that the cost of diversity is too high. Furthermore, the proponent of economies of scale would claim that similar problems occur when we attempt to design both computer chips and operating systems without an underlying standard or with diversity of design approaches.

If we accept the complaints of the proponents of economies of scale, then we must accept that in certain (but not all) situations, diversity is an evil. Not everyone agrees. Some people believe that diversity is vital and that failing to have diversity is problematic. These proponents of diversity focus on two points: that diversity has potential economic benefits and that a lack of diversity imposes potential economic risks. Let us turn to the first of these points.
Some proponents of biodiversity argue that diversity is a valuable economic and health resource. They note that plants have been an enormously important source of new medicines. Over one hundred chemical substances have been extracted from plants for use in drug treatments. [Farnsworth et al., 1985] Furthermore, only a small portion of the plants in the world have been examined for their potential medicinal value. If we drive these plants to extinction, by (e.g.) burning down the rain forests that they grow in, then their potential medicinal value (and the related economic value) will be lost forever. Because of this potential loss, many proponents of biodiversity argue that it is in our health and economic interests to maintain biodiversity and that until we have examined plants for their medicinal value, it is unwise to drive them into extinction. [Farnsworth, 1986]

Biodiversity is also economically advantageous because by preserving biodiversity we preserve potential species that have useful traits that we might want to crossbreed with our current crop plants in order to develop a "better" crop plant. As an example of this, Iltis notes how the discovery of two new types of tomato plants (*lycopersicon chmielewskii* and *lycopersicon parviflorum*) led to the development of a new hybrid tomato that increased the value of the crop, for farmers, by approximately $8 million per year. [1986]

Both Farnsworth and Iltis present an economic defense of biodiversity and (where applicable), diversity in general. On this account loss of biodiversity leads to loss of economic value and increased or static levels of biodiversity leads to economic growth. If this position is true, then it seems as if maintaining biodiversity is an extrinsic good.

Proponents of diversity also argue for diversity by pointing out that a lack of diversity could lead to a serious disaster. (As is the case in the Y2K problem.) One example of this argument comes from opponents of the use of genetically engineered crops. They argue that genetic engineering will produce hybrid breeds that, due to their identical genetic structure, will be catastrophically susceptible to disease. [Suzuki and Knudtson, 1989] An example of such a disaster is the 1970 epidemic of Southern leaf-blight fungus that cost the corn industry 12 percent of its crop. A relatively benign fungal infection became an epidemic because of the introduction of a modification to the genetic structure of corn that was intended to make the male portion of the plants sterile. Unfortunately, a hidden trait, susceptibility to the poisons secreted by the fungus that causes Southern leaf-blight, was closely associated with this genetic modification and was "smuggled aboard" with the genetic modification. The result was that approximately 80 percent of the 1970 American corn crop contained this trait and 12 percent fell victim to Southern leaf-blight.
THE MILLENNIUM PROBLEM

A problem with this value-driven approach is that by focusing on the economic good of a particular bit of biodiversity, we can be led to results that suggest that decreased diversity is better than increased diversity. Clark has shown that it is sometimes economically more valuable to take immediate profits from the exploitation, to the point of extinction of a resource (e.g., blue whales), and invest those profits in growth markets than it would be to continue to exploit the resource on a sustainable basis. [1973]

For example, assume that the exploitation of whales, on a sustainable basis, earns us $500,000 profit per year from now on. But, if we exploit the whales to extinction this year, then we will make a $5,000,000 profit this year. So this year we would have made $4,500,000 more than we would have made if we had exploited whales on a sustainable basis. We might think that since there are no more whales, the whalers will have no income in future years; resulting in a net loss to the whalers of $500,000 per year in perpetuity. After 9 years of no income, the whalers would start to lose more than they had gained by hunting the whales to extinction. But, this is not necessarily the case. If we invest the extra $4,500,000 at a sufficiently high rate, then we would gain more each year in investment income than we would lose due to not being able to hunt whales. Imagine that we invest the $4,500,000 in (e.g.) high cap mutual funds (or some other investment). If that investment yields, on average, more than 11.1% per year, then it is financially better to drive the whales to extinction immediately and invest the excess earnings, than to hunt the whales on a sustainable basis. If our investment yields 15% per year, then we make $675,000 per year on the $4,500,000 that we received by driving the whales to extinction. Decreasing diversity, in this case, leads per annum to $175,000 more than maintaining diversity. So, by focusing on the economic value of resources, it can become the case that diversity is not an economic good, but rather an economic evil.

Similarly, it does not follow from the argument that lack of diversity leaves one open to economic disaster, that diversity is an economic good. After all, the economic cost of some disaster might not be sufficient to overcome the economic benefit of the lack of diversity (the economies of scale). The economies of scale might be so great that any economic cost of the disaster would be swamped by the benefit of the economies of scale.

While we agree that on certain occasions it is economically disadvantageous to maintain diversity, that does not mean that as a rule diversity is economically disadvantageous. Rather, diversity tends to produce more economic benefit than a lack of diversity and diversity also tends to prevent disasters like the Y2K problem and the Southern
leaf-blight disaster. Even though the economic proponent of diversity might be wrong on certain occasions, she is right more often than wrong. Because diversity tends to lead to the types of outcomes that people prefer, we take diversity to be an extrinsic good for the IT industry.

Mill would, we believe, agree with this position. He contends that there are four reasons for allowing or supporting diversity of opinion and expression: (i) a silenced opinion might be the truth; (ii) a silenced opinion may be false, but it may also contain some truth; since no opinion is immediately known to be completely true, it is only by the collision of opinions that we may converge to the truth, (iii) even if the received opinion is true, if it is failed to be contested then it will be held without comprehension of the reason for holding it; and (iv) if we silence opinion, then the meaning of the true opinion, its power to improve the character of men, and the growth of real conviction in the true opinion will be lost. [Mill, 85-120] Without diversity (and the freedom to be diverse) and the conflicts it brings, we are unlikely to find the truth and once we find it, it is likely to lose all meaning for us.

The philosopher of science, Paul Feyerabend holds a similar view. [1988] Using a historical analysis of the “growth” and development of scientific knowledge, Feyerabend argues that science is (and must be) an anarchistic enterprise; that it is only when scientists break the methodological rules of the science of their times that progress occurs. Feyerabend recognizes that scientific progress does not always occur when a scientist breaks the methodological rules, but notes that it is the breaking of these rules that is necessary for achieving progress. If this is true and if we desire progress in science, then we must encourage a diversity of methodologies of scientific inquiry and a diversity of scientific theories; even though, it might on occasion not lead to scientific progress.

Foster also holds a view that diversity tends to lead us to truth. [1994] He contends that we should have a diversity of ideas (multiculturalism) and expose students to those ideas in their curriculum. His reasoning is that even if these other ideas are false we may gain some understanding or insight from studying those views. In essence, by being exposed to the ideas of various cultures we might gain insights into nature that we would not gain otherwise.

This (Mill, Feyerabend, and Fosters) argument for diversity is simply that diversity and the conflicts it brings tend to lead one towards the truth, while a lack of diversity tends to slow (or stop) the growth of knowledge. This is both a simple and profound argument for diversity, but how does it relate to the practical sciences of the computer industry?

The extrinsic good that plays a central role in the computer industry is improvement. We want our computer systems to function more efficiently,
at greater speeds, with greater ease, and power. Stagnation along these lines
is not good. We contend, that a lack of diversity in the design of systems
leads to a slowing of computer technological improvement.

Some might argue that we are wrong. That a lack of diversity does
not lead to a decrease in improvement in computer systems. After all,
computer chips are enormously more powerful and faster than they
have ever been. Operating systems are more powerful, efficient, and
can perform a much wider array of tasks than they have in the past.
Programming languages have developed from simple machine code to
object oriented languages like C++ and JAVA. So growth and improve-
ment have occurred.

But note, how did such growth and improvement occur? In precisely
those circumstances that Mill, Feyerabend and Foster point out. When
people go beyond accepted dogmas and methodologies. For example,
most of us have personal computers that work on windows type sys-
tems. Previous to these operating systems there were simple text-line
based operating systems like MSDOS and VMS. Those systems were
efficient, powerful and grew in usefulness over time. But, it is only when
a new approach, that goes against the very core of this text-line based
system, arrives that genuine and dramatic improvement in (e.g.) ease of
use occurs. When Xerox and then Apple introduced windows based sys-
tems, they violated the standards of operating systems and in violating
those standards improvement occurred.

This analysis tells us precisely what is wrong with the argument against
diversity that comes from economies of scale. If we adopt an econ-omy of
scale, then we are locked into a particular strategy of systems design. That
strategy may be good, but it is unlikely to be the best strategy. (In the year
2000 case dropping the first two digits of the year was a good, but not the
best, long-term, strategy.) Furthermore, that strategy may lead to improve-
ments in technology, but its improvement will be less than what it could be
were it to run head to head with a different strategy. It is only when we
allow there to be a market of strategies that the dominant strategy will im-
prove or be replaced at a rate and to a degree that is consistent with the
maximal possible improvement.

Diversity is good because it promotes freedom of ideas and the free-
dom to design systems in a way that the individual thinks is the right
way. This freedom will lead to conflict between various systems. This
conflict will result in learning on the part of proponents of both systems.
And this learning is what facilitates genuine growth and improvement
in technology. Consequently, diversity and freedom, because they maxi-
mize the possibility of technological improvement, are extrinsic goods
of the computer industry.
VII. LIMITS AND SAFEGUARDS OF MORAL TECHNOLOGY

As we stated earlier, Mill limits freedom of action, though not of opinion, by whether or not the action significantly harms another person. No one, for example, may incite a riot against a corn merchant outside of that gentleman’s door for it may lead to grievous injury to the merchant or his family. However, if the action will not severely harm anyone other than the agent, then it is permissible for the agent to do it. On these grounds, prevention of severe harm to others is the only limitation that can be set on business’s freedom and technological diversity. As a result, the marketplace can have virtually unlimited technological diversity and intellectual rigor, in other words, originality.

In order to avoid some of the problems caused by implementing new computer products without adequate foresight, there needs to be some procedure in place that helps prevent dangerous technology from entering the marketplace in the first place. We want to suggest a review board, loosely based on Underwriter’s Laboratory, that will consider new products and determine if they ought to be marketed. If the board decides that a new technology looks to be too dangerous, then it will not be certified as safe for consumer use.

The reason that we selected an Underwriters Laboratory model for the review board is that it has both the flexibility to stimulate new ideas and products and provide important information to consumers, while at the same time encouraging businesses to submit their goods to the review board. We wanted to avoid requiring all new technology from going through the certification process because it would be very costly in time and energy. Also, by requiring all new technology be certified the consumer’s choice in products would have been illicitly limited. Those goods deemed dangerous would not be allowed to enter the market, even if the consumer wanted to take the personal risk of purchasing and using them.

The incentive for businesses to have their technology go through the sometimes lengthy approval process is twofold. First, the fact that the technology was board certified would assist in any lawsuits that may arise from defects in the product. The manufacturer has additional legal evidence that shows they tried to eliminate any problems, thereby reducing their responsibility for any unfortunate oversight to a minimum.

Secondly, the aura of quality that UL approved products have will also attach itself to board certified technology. If given a choice between the UL certified and non-certified products, rational consumers will choose the former for they have more information on the quality of that product. They know that only products which meet the Underwriters Laboratories’ high standards will be approved, while the other is a
mystery. The additional information helps the consumer to make the best possible choice on what to purchase. In the same way, board certified technology helps the consumer make a more informed, and thus better, choice.

In this section, we will first argue for why a technology review board needs representation from all relevant parts of society, and then justify why each area named should have a representative. Finally, we will sketch out the review process that the board may use to make its decisions on what is dangerous technology and whether or not to certify it.

The board will be comprised of members from academia, business, government, and private life who know something about the technological field of the product under review. By having such a diverse group, we can better ensure that everyone’s needs are met without having to worry as much about latter catastrophes. Based on the unique perspectives each member has, the board’s diversity will help it to better identify problems and account for a wide spectrum of needs.

The obvious reason that we need adequate representation from all spheres of society is to prevent overlooking some obvious failing in the technology. Each person observes and processes data from her own perspective. Scientists, for example, will observe based on scientific reasoning, philosophers on philosophical reasoning, and so on. The review board should reflect this fact. By incorporating the diverse viewpoints a more holistic understanding of the overall situation is achieved. This broader perspective will facilitate good decision making by pointing out deficiencies and strengths in the product that could only come from the role-differentiated information.

Business people should be on the review board, in part, because they look at information and make decisions based on business goals and ideas. They will examine the proposal to determine if it is good for business in the short and long terms. More specifically, the business community members will focus on whether or not the technology will unwarrantedly hurt business or the people involved in business. They can draw upon their experiences of other technological innovations to determine whether or not the proposed product is similar to others that benefited or harmed people and corporations.

Secondly, since the technology is sure to affect the business sphere, business people have a right to representation. They will have to deal with any problems that may arise due to insufficient testing of manufacture. Thus, in the interests of fairness, business members should know what they are going to be encountering when they try to operate their respective businesses. To do otherwise would be to illicitly force them to pay for problems not of their own devising; a violation of autonomy.
Public agencies must also be represented. Governments have a responsibility to protect their citizens from avoidable disasters and the harm that accompanies them, when it can reasonably do so. Government is our last stopgap in many cases from dangerous products. (See the FDA, SEC, etc.) Thus, in order to do their jobs properly, they must be represented on any panel that reviews technology. If they should not be allowed on the board, the result would be similar to hamstringing an athlete, and then forcing her to run a race.

Another reason that the federal government should be represented is it often has to be the one that organizes and distributes the relief for crises. If the government is expected to clean up technological messes, then it has to be part of the decision making procedure. It must agree to support the conclusions the board reaches, though it does not have to agree with them, to be accountable for the introduction of the technology and any bad consequences such products produce. By its agreement it takes responsibility for the action because it freely decided to bring about that deed. To force it to be accountable even if it did not have a voice in the decision is a form of taxation without representation.

A review board without adequate university representation would be deficient in at least two important ways. First, universities are, for the most part, the best place to review technology before it is introduced into the open market. Since they are centers for research, universities have the equipment, people, and experience necessary to analyze the new product. They in effect have an expertise that is lacking in business, government, and the private individual to determine how the product actually works. By using their unique technological abilities, deficiencies that the other members of the boards could not obviously see would have a much better chance of being discovered. Some of the very best computer scientists, biologists, and clinicians, among others, work for colleges. Many of these professionals are researching the very technology that is proposed; so they have a great deal of knowledge about it. Technological academics are a valuable resource of information that must be utilized in order to make the best decision.

Secondly, besides those academicians involved in technology, university members in the critical reasoning disciplines of the Liberal Arts will bring a new perspective to the review board. Unlike the business person who analyzes technology for its business effects, or government who considers the technology's impact on its citizens, university people have greater consideration for the overall impact of a product into society. Professors and researchers, especially those in philosophy and sociology, have received an extensive education to help hone their critical reasoning skills. They are better able, in quite a few cases, to perceive and analyze arguments for and against introduction of the product. Since
so much depends on this technology, it is vital to have the best critical reasons in order to make the best decisions, not decisions for the best.

Academics also have had the educational background to broaden them in a variety of areas which will serve them well in the decision procedure. No liberal arts Ph.D. worth the name will lack skills in philosophy, sociology, history and other important disciplines. With this additional information, they can better weigh the technology from the perspective of many different fields, thereby allowing them to render the best decision.

The final group represented on the board should be members of the public for the simple reason that any new technology is likely to impact them the most. More specifically, hazardous technology hits members of the general public directly in the pocketbook by forcing them to ultimately pay for fixing it. The other entities may soften the blow to themselves by passing the costs on to the consumer or taxpayer, which the consumer obviously cannot do. There is no one lower on the financial ladder than the average citizen.

The Millennium Problem is just one such case in point. Though business and government will seemingly pay for the major majority of the solution, private citizens will actually be the ones forced to shoulder the costs. They will have to pay higher taxes so that the government that represents them can assist in the cleanup. Government, though it has a license to print money, must have hard capital to pay its bills. The only way it can collect the vast majority of it is through its citizens.

The private individual will also have to pay higher prices for goods and services that are produced by businesses that have had to fix the Y2K disaster. As anyone who has studied business knows, costs must be passed on to the consumer in order for a company to make a profit. If the company should try to shoulder the entire burden on its own, then it will probably have to choose between paying lower dividends, laying off workers, delaying capital expansion, or a combination of the three. In order not to have to go to any of these three extremes, the corporation must pass the financial burden to its consumers. We must also be aware that the enormity of the problem could force the company into Chapter 11, if it does not make the consumer pay for the costs of reprogramming and new equipment.

Finally, private individuals will have to deal with the new technology whether or not they want to do so. All business the private individual uses have been to some extent computerized or automated. Also, people's homes and cars have a great deal of technology with which they must contend everyday. Since technology is so much a part of their lives, and bad technology such a threat to their well-being, private individuals should have a say on what technology manages to make it to the market. Again, the idea is taxation without representation is unfair.
Perhaps the most difficult task to perform is determining what set of criteria the technological review board must use to help ensure against dangerous technology. What we are striving for are principles which do not unnecessarily stifle the marketplace, while at the same time, respect consumers' choices and helps protect them and others from unforeseen harm. The criteria should also be simple enough to understand and use so that anyone on the review panel could do it with comparative ease. This is a tall order to fill, but we can sketch out some of the main ideas and fill them in at a later date.

The first stage immediately winnows out hazardous products with a relatively quick determination if there are any obvious drawbacks to the technology. Basically, the members must ask themselves if there is a defect that clearly stands out on the product that makes it very dangerous. If there is, then passage is impossible. A genetic manipulation, for example, that is likely to produce dangerous hybrids would automatically fail to make it pass this stage.

The second stage is the prudent determination of need. If the market has no need for the product, that is, there is no demand, then it ought not be distributed. It is at best nonsensical to court disaster when one does not have to do so; it is the height of imprudence. By introducing new technology without there being a real need, we create a situation in which there is greater danger to ourselves than there are foreseeable benefits. The world is too complicated and dangerous a place as it is; thus, we are obligated to prevent unneeded technology from taking root.

The third stage requires the members to consider the short term costs and benefits of implementation. Data must be provided by the company projecting how the market will be impacted by the technology. This way the members can decide if the product is safe for the short term. If there are obvious remedies that will help reduce the short term costs to society, then the technology should be rejected until improvements in this area are made to it. Windows 95, for example, had some flaws in it that should have been worked out prior to the introduction of the product, rather than afterward as indeed did occur. The motivating factor to market the software seems to have been purely financial on the part of Microsoft.

The fourth stage is related to the third. At this level the long term costs and benefits of the technology must be examined. Once again the procedure is based on data submitted from the company. Of course, this will be even less certain than the data from the third stage, but it should give the panel an adequate idea of what can be expected. The panel will be able to see if the technology has a "shelf-life" that at minimum requires a warning to the consumer. Though no absolute guarantee, if this
had been done for computer software and hardware, the Millennium Problem might have been avoided.

The fifth stage asks the panel to determine if there are any moral objections to the technology. The evidence they should consider is whether or not mainstream ethicists have considered the impact the product will have on people individually and as a society. If there is considerable opposition to the introduction of the technology, then it ought not be immediately certified until the moral problems are discussed and guidelines on how to proceed established.

On the other hand, if the product is so new that no one has had the time to ponder its moral implications, it must be detained from entering the market. Quite a few new developments have occurred in the biogenetic field that have left ethicists struggling to catch up. This is not a good condition to have, though some would disagree. However, the mere fact that we can do something does not mean we ought to do so. We could clone a human being, but we ought not until we consider whether or not it is moral. Determining it is unethical after the act is much too late; prudence in our moral actions require us to delay rather than rush in.

This stage obviously rejects the seemingly ubiquitous notion in technological fields that if we can do something, then we should. Mere ability does not imply obligation, otherwise we get the morally repugnant principle of "Can implies Ought" that no one in their right mind would subscribe to. We can, for example, say something degrading to our students just for our own amusement, but we ought not to do it.

Technology that is uncontroversial to the majority of ethicists, of course, passes this test. Controversial technology may also move on to the next stage provided that adequate discussion has been given. The review board can determine when this point is reached based on what they would do as rational creatures. That is, would they want this technology used by them? If the majority would not mind it, then the product may proceed further in the process.

The sixth and final stage requires the panel members to consider if there is adequate diversity if the product is introduced. Lack of diversity is dangerous because without it there is nothing to step into its niche should it fail. We need to have competitors waiting in the wings if not actually competing, so that we can get the best result. The sixth stage is based on a Darwinian idea of diversity being best for evolution. If we had had true diversity in the computer industry, then there would already have been a successor to take over when the Y2K problem was discovered in 1979, instead of being saddled with technology that must only now be patched up or replaced.
On the basis of efficiency, the stages of approval, with the exception of stage one, do not necessarily have to be followed in the order given. Approval processes and panels usually take a great deal of time to make adequate decisions. Based on the need to move technology into the market at a reasonable pace, the various stages can overlap. If it is possible to do all six stages at once, then so be it. A quick but well reasoned decision for the technology's manufacturer allows her to either write off the product, if financially unsalvageable, and begin work on new technology, or immediately start making the necessary adjustments.

However, the main purpose of the review board is to help keep dangerous technology from the market, not to merely be a rubber-stamp for the computer or other technological industries. The board members must always take the time necessary to come to a decision that they will not have any real cause for which to feel remorse. Basically, they must each ask themselves, "If I were the person who would be most harmed by this technology going wrong, would I have wanted it marketed anyway?" If the majority truly thinks that it is better to market the technology than not, the product receives board approval. This is the only criterion they can use to determine when they have considered the product enough. Anything less would be an unwise endangerment of the innocent and unsuspecting public.

VIII. Conclusion: To Encourage Originality, Freedom, and Diversity

Originality is necessary not only in the political realm but also in the business world as well. As Mill says, "In proportion to the development of his individuality, each person becomes more valuable to himself, and is therefore capable of being more valuable to others." [130] Fostering originality, therefore, is good business; the long term interests of a company are served by more valuable employees who are creative, rigorous, and wise. In the marketplace, businesses will create and have access to new, useful ideas and products, that would not exist in a world governed by conformity.

The heightened efficiency of businesses run by people who embody originality would entail benefits for society as a whole. Consumers would not only have additional goods from which to choose, they would be treated as an end in themselves by those who respect the originality of others. Hence, it is in the best interests of all people to guarantee a marketplace with virtually unlimited, safe technology, based on originality. If the business world did this, then the likelihood of technological disasters such as that of the
Millennium Problem would decrease dramatically, while at the same time making the world a better place for everyone.

*East Carolina University*

*Bowling Green State University*

**NOTES**

Research for this paper was supported by Bowling Green State University and East Carolina University. Grateful acknowledgment is also made to Edward C. DeVito for helpful discussion, Robert L. Holmes, Richard Miller, Linda Hill, and an anonymous referee who provided insightful comments on a previous draft.

Authors are listed in alphabetical order.

1. This is based on Capers Jones' report, "The Global Economic Impact of the Year 2000 Software Problem," which is available at http://www.spr.com.

2. For example, United Airlines estimates that a one-hour system delay at Chicago International Airport would result in a cost to (just) United Airlines of over $100,000.

3. Obviously this is an inadequate algorithm as it does not take into account the difference between your birth month and day, and the current month and day. For simplicity's sake we use the inadequate algorithm.


5. What is odd about this way of saving space is that it really isn't all that efficient. If we have a 6-byte-long date field, where each byte contains 8 bits, then we are using 24 bits of space. We can represent numbers between 0 and 16,777,215 using 24 bits. So, we could use the 24 bits to represent the number of days since some particular start date. If we had the start date as January 1,001 AD, then we could represent, using just those six bytes, every day until the year 45,955 AD.


7. A program is Y2K compliant if it is capable of avoiding the Y2K problem.

8. Ibid.


11. Ibid. Rabin and Tiemey.

12. For more details about the effect of dependencies in the Y2K problem, see Leland and Meador [1997].

13. An interesting experiment to try with your IBM personal computer is to change the date and time to December 31, 1999 at 11:58 p.m., then shut off the computer for a few minutes. When you turn the computer back on you may be surprised by the date the computer thinks it is. If your PC runs on DOS, Windows 3x, or has a chip previous to Pentium era chips, then it is subject to the Y2K problem. If you have a Macintosh, try changing the date to February 6, 2040 at 6:26 a.m. and see what happens. (Apple uses a 32 bit value to store seconds that is able to represent date/time between 12 a.m. January 1, 1904 and 6:28:15 a.m. February 6, 2040.)

14. Embedded systems occur in telephone systems, taxes, copiers, mobile phones, video cameras, fire control systems, heating and ventilating systems, elevators, escalators, backup lighting and generators, security systems, door locks, heart defibrillators, pacemaker monitors, patient information systems, patient monitoring systems, X-ray equipment, pharmaceutical control and dispensing systems, planes, trains, marine craft, automobiles, radar systems, traffic lights, cable systems, among other pieces of equipment.

15. Not all embedded systems are subject to the Y2K problem. Those systems that do not themselves refer to date/time do not have Y2K problems (unless the applications software running them do, but then that is, essentially, a software problem).

16. Mill is not as clear as he could be in chapter three on the formulation of the limitation principle. At the beginning of the chapter, freedom of action seems to be limited only when it causes severe harm to another agent. However, later on, there seems to be some modification to this which excludes more actions than those mentioned above.

17. Throughout this section, and the rest of the paper, we will be assuming two uncontroversial moral principles as the basis of our arguments. First, the utilitarian notion that it is wrong to cause unnecessary harm seems to be a virtual truism. Hurting someone either physically, emotionally, or financially if one does not have to do so cannot be moral. Secondly, the Kantian principle of not treating persons as mere means is unproblematic, except admittedly for an unclear definition of “mere means.” Usually, it is wrong to treat a morally neutral or good person as one would a chair or other inanimate object; it disrespects the individual’s true value as a person.

18. The other criteria work much the same way. If there is a strong duty for the agent to prevent the technology from being implemented, then he must have had a great deal of control over the situation and a strong duty to foresee the long term results of the software/hardware. Given these fundamental standards, our task of assigning blame to the myriad of people responsible for the Y2K problem, though still difficult, will now be a little bit easier.


20. Mill is concerned only with harm done to others and not what the agent does to himself. He seems to think this way because the agent has some sort of choice in
the harm done to himself, while the others do not have any say in what is done to them.


22. Except in the principles of mathematics.

23. The Due Care Theory states that a producer has more obligations to a consumer than merely filling the purchase agreement precisely because the seller has special knowledge that the buyer does not or cannot have.

24. One version of the Due Care Theory states that the producer is responsible for product defects even if the buyer has explicitly accepted them.

25. Of course, those programmers who were not adequately trained are responsible only for those errors that they could rationally have chosen to avoid. Ignorance that is not due to an intentional blinding of oneself to pertinent information and ideas excuses the agent from blame. To hold such people responsible would be similar to expecting a small child to perform as a mature moral agent.

26. The executives at Ford decided not to make an $11/vehicle design alteration to fix a flaw in the Pinto even though it would have saved lives and prevented injuries. An internal memo stated that it would be cheaper for the car maker to pay any damage awards that would result from the defect.

27. Oddly enough this mad pursuit for profit may eventually hurt the computer industry more than it helps it. The pessimistic mood generated by the computer industry's rank greed has fostered discontent in our elected officials. They realize that they will have to take part in any practical solution to the Y2K problem. With Congress weighing in on this matter, legislation nonbeneficial to technology companies is likely to be forthcoming. Thus, in the long term, the computer industry may have to face increased Congressional scrutiny and regulation.

28. Mill, as a utilitarian, must reject the Kantian notion of retributive justice. Utilitarianism can only look to the future while retributive justice must look to the past.

29. We are purposively excluding those situations in which companies created products they did not have any reason to realize where defective. An innocent, non-negligent mistake does not cause an agent to be blameworthy for wrongdoing.

30. Perhaps the most practical way is to allow companies to apply for the governmental assistance that will be necessary to help solve the Y2K problem. From those that submit requests, we could determine how much responsibility they bear for creating the problem in the first place and charging them accordingly.

31. The arguments and conclusions apply equally well for state and local governments.

32. We can see this move to standardization in digital video programming [Cosper, 1997], medical computing [McCrone, 1993], and the Department of Defense [Charles, 1996], among other fields. For a short introduction to standardization in the computer field, see Robinson and Cargill [1996].

33. Martin and McClure [1985], Brown [1992], Hughes, Michtom, Michtom [1987], and Payne [1986], to name just a few.
34. The purpose of this example is merely to clarify the point as simply as possible. We have therefore left many possible details out. For a complete analysis, see Clark [1973].

35. On Liberty [123].

36. This applies equally well to the federal government, which is one of the world’s largest producers and consumers of technology.

BIBLIOGRAPHY


THE MILLENNIUM PROBLEM


