SALVE REGINA UNIVERSITY

IN THE SHADOW OF TECHNOLOGICAL CATACLYSM: THE COMPLEXITIES OF Y2K ALARMISM

A DISSERTATION SUBMITTED TO

THE FACULTY OF THE DEPARTMENT OF HUMANITIES

IN CANDIDACY FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

BY

RICHARD FRANCIS FINN JR.

NEWPORT, RHODE ISLAND

MAY 2002

UMI Number: 3046374

UMI

UMI Microform 3046374

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

SALVE REGINA UNIVERSITY

GRADUATE SCHOOL

The dissertation of Richard Francis Finn Jr. entitled "In The Shadow of Technological Cataclysm: The Complexities of Y2K Alarmism" submitted to the Ph.D. Department in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Graduate School of Salve Regina University has been read and approved by the Committee:

Reader	Ann Nelson, RSM, Ph.D.	March 1, 2002. Date
Reader	Peter Dombrowski, Ph.D.	3/1/02 Date
Mentor	Eugene Lappin, FSC, Ph.D.	5-01-2002 Date
Director, Ph.D. Program	Theresa I. Madonna, Sc.D.	6-6-02 Date
Vice President for Academic Affairs	Barbara A. Kathe, Ph.D.	06.14.02 Date

ABSTRACT

In our modern information age, few technological problems have had as great a potential catastrophic impact on humanity as the year 2000 computer date change (Y2K). Although Y2K was considered initially a technical issue with a straight-forward technical solution, it was later found to have significant human consequences and educed a response unparalleled in the late twentieth century. If left unaddressed, Y2K could have caused a serious interruption to critical elements of our complex national infrastructure. As alarmists consistently warned, these disruptions would have grave repercussions in the social, political, and commercial institutions that rely on them.

Although Y2K is judged to be a nonevent by some, this study demonstrates that the multiple challenges related to the year 2000 computer date change were more than a technical threat. In addition, this dissertation makes evident that Y2K alarmists were effective in highlighting the risks of modern information technologies.

Y2K presented a unique crossroad to examine our attitudes toward advanced technology. The human responses to potential consequences of a cataclysmic event do not always have to be negative. The manner in which these are addressed may also serve as models for resolving future problems.

As the Y2K challenge was recent, scholarly research is still in its infancy to assess the effects that this technological episode has had upon humanity. This dissertation seeks to provide an important contribution to this body of knowledge.

To my wife Rita

and in the memory

of my father

ACKNOWLEDGEMENTS

A number of people deserve my deep appreciation for helping me prepare this dissertation. First and above all, I would like to thank Dr. Eugene Lappin, FSC, my mentor. His exemplary academic discipline, quiet encouragement, and sagacious guidance have been central to completing this work.

I am also grateful to the other members of my committee, Dr. Ann Nelson, RSM, and Dr. Peter Dombrowski for their valuable suggestions, critical review, and professional advice throughout the research and writing of this study.

Additionally, I wish to extend gratitude to my colleagues at Sonalysts, Inc., particularly Thomas McPherson and Donald Estes, for their steadfast support, analytical scrutiny, and endless patience.

Lastly, without the enduring inspiration and academic fortitude of my son Rich, this dissertation would have remained only a pipe dream.

CONTENTS

ILLUSTRATIONS

ABBREVIATIONS

ABC	American Broadcasting Company
ADL	Anti-Defamation League
AIDS	Acquired Immunity Deficiency Syndrome
AM	Amplitude Modulation
AOG	Assemblies of God
ATM	Automated Teller Machine(s)
C3	Command, Control, and Communications
CDC	Centers for Disease Control and Prevention
CEO	Chief Executive Officer
CFO	Chief Financial Officer
CHA	California Humanities Association
CIO	Chief Information Officer
EO	Executive Order (Presidential)
FBI	Federal Bureau of Investigation
FEMA	Federal Emergency Management Agency
FERC	Federal Energy Regulation Commission
FRB	Federal Reserve Board
FRP	Federal Response Plan
GAO	General Accounting Office

GDP	Gross Domestic Product
IEEE	Institute of Electrical and Electronic Engineers
IT	Information Technology
IY2KCC	International Year 2000 Cooperation Center
LC-MS	Lutheran Church - Missouri Synod
NAAWP	National Association for the Advancement of White People
NBC	National Broadcasting Company
NEH	National Endowment for the Humanities
NERC	North American Electric Reliability Council
NSRI	National Survey of Religious Identification
NWO	New World Order
OMB	Office of Management and Budget
PBS	Public Broadcasting System
PY2K Council	President's Council on Year 2000 Conversion
SPLC	Southern Poverty Law Center
SRU	Salve Regina University
TEOTWAWKI	The End Of The World As We Know IT
UFO	Unidentified Flying Object
WTC	World Trade Center
WTO	World Trade Organization
WWW	World Wide Web
Y2K	Year 2000

CHAPTER ONE

INTRODUCTION

Introduction

Throughout the twentieth century, humankind found itself in the shadow of cataclysmic upheaval precipitated by the consequences of the technologies it developed. Many of the technical innovations produced in this period demonstrate that problems generated by technology had effects throughout society that few of their creators had anticipated.

For example, the early days of the atomic era saw an intense effort to exploit nuclear technology in as many ways as possible beyond its use as a weapon. In 1954, advocates suggested that nuclear-generated electrical power would be "too cheap to meter" (U.S. Nuclear Regulatory Commission 1999). Some proponents touted atomic energy as the source of "revolutionary industrial, agricultural, and medical discoveries that would lead to accelerated economic progress, greater leisure, and longer life" (U.S. Department of State 1955, 1). It was believed that this

energy would drive ships, trains, or airplanes, provide heat in colder climates, or distill seawater for drinking and irrigation (Lansdell 1958, 115-123).

These optimistic predictions, nevertheless, dimmed as unanticipated or undesired consequences of nuclear energy emerged. A major concern for the United States was the proliferation of atomic weaponry, first by the Soviet Union and China and then by other countries, heightening the fear of nuclear war. Even without war, the danger from tons of nuclear waste, whose toxicity lasts thousands of years, increased monthly. Finally, there was concern regarding the safety of this power source.

The accident at Three Mile Island nuclear power plant in Pennsylvania (1979) made Americans more conscious of the potential hazards that even "peaceful" atomic energy poses. In 1981, coolant leaks at a plant operated by the Tennessee Valley Authority increased the uncertainty about the value of atomic energy, compared to its risks. The devastating nuclear accident in Chernobyl, Ukraine (1986) compounded this anxiety. More importantly, the long-term ecological effects of neutralizing radiological waste became a problem of immense proportion (Capra 1982, 245-247). Sounding the alarm about these dangers grew a steadfast assemblage of concerned individuals and groups. Notable authors, such as

Fritjof Capra, and environmental organizations, including Greenpeace and Worldwatch, are members of this assembly.

While these problems remain important, this study is not about the effects of an industrial or atomic technology. Rather, this dissertation addresses primarily one consequence of our burgeoning information-age society, namely, the effects of computer technology.

Our modern information age is barely fifty years old. During this period, information technologies such as the computer have significantly influenced our lives. Today, our daily activities depend directly on computer-supported infrastructures that span our country. Yet the incorporation of this technology into our lives has not been without unintended consequences and difficulties. In *The Invisible Computer*, Donald Norman asserts bluntly that the computer is a "disruptive technology," often causing frustrations until we modify it to suit our needs, or we adapt to the technology (Norman 1998, 231-261).

Today, the scale and complexity of modern technologies assure that catastrophe happens more often than in previous centuries. Compounding this dilemma, the nearly instantaneous capabilities of the news media, often disseminating partial facts and half-truths, help fix the adversity caused by these events firmly in the public

consciousness. Naysayers, stressing the dangers of these technologies, often provide fuel for media-spawned alarm. However, the consequences of a cataclysmic event do not always have to be bad. The manner in which disasters or potential troubles are addressed may also serve as models for resolving future problems.

Modern technology-related catastrophes are diffuse processes that seem to develop invisibly and then are discovered too late. A long time may pass between cause and effect. Frequently, the cause is not a single event but the cumulative effect of many small actions that may snowball to near-calamitous results.

Contributing to the fear associated with the consequences of these events is often a small vocal group of alarmists, determined to highlight the risks allied with these technologies. These fears may not always be directly related to the technical aspects of a technology, as with the instance of nuclear power. Rather, greater concern may exist about the cataclysmic impact on infrastructure and social systems should a nearly ubiquitous advanced technology catastrophically fail. This was the case when the software within most computers—the foundational element of our information age—transitioned from the year 1999 to the year 2000.

The Y2K Technical Problem

The year 2000 problem concerned the nearly universal practice of using a two-digit year format in computer programs. Some experts viewed this practice as a technological design flaw (Core 1999). In an effort to conserve expensive computer memory space, early programmers adopted a six-digit code when entering dates in their software: two digits for the year, two digits for the month, and two digits for the day (yy,mm,dd). Thus, the year 1999 was written as "99" (de Jager and Bergeon 1997, 1-5). Two-digit years became the standard for software date coding in 1968 (U.S. Department of Commerce 1968). Using this two-digit format worked for decades since program writers could rightfully assume that the two-digit year, e.g., "78," was always preceded by "19" (meaning 1978). This system succeeded until the year 2000 ("00") approached.

Many feared that programs requiring dates to complete their operations would not be able to discern whether the digits "00" meant the year 2000 or 1900. As a result, some programmers believed that computer systems would reject all dated information, compute erroneous results, or simply not run (Kratzen 1996). Because of its relationship with the year 2000, this predicament had many names, including

"millennium problem," "millennium time bomb," "year 2000 bug," or simply, "Y2K" (the year "2K" or 2000 from Greek *khilioi,* thousand) (Davis, L. 1996).

The Y2K problem began in the earliest years of software development when the high costs of developing and using computer memory required that programmers write applications that minimized memory use. In 1965, when many of today's "legacy" systems were designed, the cost of one megabyte of memory, enough for a good-sized novel, was $761 per megabyte (Jasinowski 1999). In September 2001, the cost had fallen to only 4/10 of one cent (CompUSA 2001). Thus, the costly 1965 price per unit of storage was nearly 200,000 times more than it became in the year 2000. This inexpensive memory is only a very recent phenomenon.

Even Alan Greenspan, Chairman of the Federal Reserve Board, and perhaps one of the most influential persons in the economic world at the turn of the twenty-first century, stated that he might have contributed to Y2K:

> I'm one of the culprits who created this problem. I used to write those programs back in the '60s and '70s and was so proud of the fact that I could squeeze a few elements of space out of my program by not having to put 1-9 before the year. . . . It never entered our minds that these programs would have lasted more than a few years. And, as a consequence, they are poorly documented. (U.S. Congress, Senate 1998c)

The computer date problem was discovered by the activities of a religious denomination. In the 1950's when the Church of Jesus Christ of Latter Day Saints (Mormons) was converting its massive genealogical records into an electronic database, workers found that dates before 1900, such as 1736, were not recognized by the existing two-digit year database. Programmers then deduced that, if software was not able to read dates before the 1900's, these programs probably would not accept coding for dates after 31 December 1999 as well. To accept genealogical records, program writers created the first four-digit year program (Taylor, C. 1999). However, the current standard for date coding did not change from six to eight digits (yyyy,mm,dd) until 1998, less than two years before the Y2K crisis was to occur (U.S. Department of Commerce 1998).

Pervasiveness of the Technical Problem

Although Y2K was generally associated with 1 January 2000, this date was not the only one when computer programs were expected to experience date-related disruption. The millennial leap year, 29 February 2000, was also a significant date that software would have to satisfy in order to operate without disturbance. In fact, computer professionals found that software programs would have to

satisfy forty-two dates, from 2 November 1997 through 29 February 2004, to be considered fully Y2K compliant (Mitre Corporation 1998). This large number added to the complexity of the problem.

Because of the many dates that software would have to satisfy, not all Y2K-related problems would occur on 1 January 2000. The Gartner Group, a technology consulting firm, assessed that about 25 percent of Y2K-related disruptions would take place in the last three months of 1999, 55 percent throughout 2000, with the remaining 20 percent occurring in 2001 (Perez 1999). Gartner estimated that only 10 percent of Y2K failures would occur during the first two weeks of January 2000 ("No All-Clear" 1999).

From a technical perspective, Y2K extended far beyond the older, or "legacy," mainframe computers programmed in the 1950s and 1960s. In these systems, it was estimated that 180 billion lines of code would have to be remediated (CNET 2000). Even newer applications, such as Microsoft Windows 95 or Windows 98, as well as many popular word processing, database, and spreadsheet applications used by business and personal computers were not Y2K ready. Only when software companies released date-conforming editions of these programs did they become compliant. Within legacy systems, recoding to accept dates beyond 31 December 1999

had to occur. Cumulatively these repairs cost billions of dollars and sometimes took months to complete.

Additionally, specialists assessed that the date problem also existed in microprocessors, often referred to as "embedded devices" or "computer chips," used for monitoring and controlling activity in many devices. It was therefore feared that potential Y2K problems were present in items that most people rely on daily, such as automated teller machines (ATM), consumer electronics, oil and gas pipeline control systems, water and electricity distribution, automobiles, aircraft control equipment, military systems, telecommunications appliances, medical devices, manufacturing processes, and others (McCarthy 1996).

The year 2000 date problem was clearly more complicated and extensive than a casual observation would have suggested. Anticipated disruptions were not limited to individual microchips or computers, since problems could be spread throughout networked systems. Computer networks have become part of life's routine for most Americans, despite the disruptions caused by technical difficulties, malicious viruses, or intrusion by hackers (Harmon 2000, A23). Nearly 60 percent of businesses and individuals in the United States are networked either through an