Boston University
Journal of Science & Technology Law

Note

The Year 2000—Delight or Disaster: Vendor Liability and the Year 2000 Bug in Computer Software

Suzanne R. Eschrich

Table of Contents

The Year 2000—Delight or Disaster: Vendor Liability and the Year 2000 Bug in Computer Software[†]

Suzanne R. Eschrich[*]

I. INTRODUCTION

1. The approach of the year 2000 brings with it anticipation and excitement. The new century promises to usher in a host of exciting new advances in technology, and the computer industry has poised itself to be at the forefront of these technological advances. In the last few years alone, developments within the computer industry, in particular the immense growth of the Internet,[1] have dramatically changed the way many people use computers. This rapid pace is likely to continue as companies search for various new ways to integrate computers into everyday life. In the computer industry, however, there is cause to approach the turn of the century with trepidation.

2. While the prospects for future technological developments are great, the prospects for technological failures are even greater. For many businesses and individuals, computers perform critical operations, enabling them to function effectively and efficiently in the Information Age. Computers are the life-line for many business institutions. Computers process transactions, control money, keep inventory, and store an array of other data. In all such transactions, accuracy and reliability are essential. Yet design relics from the early days of computing threaten to impede and even halt the ability of computers to perform the very tasks that businesses depend on them to perform. Enter the year 2000 bug.

3. This Note will begin in Part II with an explanation of the mechanics of the bug and examples to illustrate the bug's impact on ordinary computer processes. Part III will examine the potential theories of liability that could be used to recover damages caused by system failures related to the bug. Part IV will evaluate which theories are likely to be most successful in any potential litigation and recommend

[†] © 1998 by the Trustees of Boston University. Cite to this Note as 4 B.U. J. SCI. & TECH. L. 8 (1998). Pin cite using the appropriate paragraph number. For example, cite the first paragraph of this Note as 4 B.U. J. SCI. & TECH. L. 8 para. 1 (1998).

[*] B.S., 1995, Florida State University; J.D., 1998, Boston University School of Law.

[1] In early 1997, the number of Internet users worldwide was estimated at 57 million. *See* Amy Harmon, *Global Pact Has a Local Impact*, L.A. TIMES, Feb. 17, 1997, at D3. This number is expected to increase to 700 million users by the year 2000. *See id.*

the adoption of computer malpractice to deal with this and similar computer defects.

II. BACKGROUND

4. The year 2000 bug ("Y2K Bug" or the "Bug") is an intrinsic defect in nearly all computer software.[2] The problem exists because of a programming design decision that treats the year in a date string as a two-digit rather than a four-digit number.[3] As a result, the standard date format for most data entry tasks and date calculations is the MM/DD/YY format, or, for example, 03/15/97.[4] This format limits the characters available to store the year, so the year "1997" is stored as "97" and the year "2000" would be stored, in defective systems, as "00."

5. Programmers adopted this date standard because of deficiencies in early computers.[5] Programmers were driven to make shortcuts in their programs for several reasons. The main incentive was financial. In the 1960s and 1970s, storage space on computers was extremely expensive.[6] Because of these high costs, early computers had limited storage space, and calculations with large amounts of data were slow. To increase the efficiency of calculations and reduce data storage costs, programmers developed ways to decrease the storage space required for common functions.[7] For dates, this meant storing only the last two digits of the year.[8] This is the basis of the Y2K Bug. By assuming all dates were in the 20th century, only the

[2] *See* Steven Levy & Katie Hafner, *The Day the World Shuts Down*, NEWSWEEK, June 2, 1997, at 52, 54.

[3] *See id.*; *see also* Peter de Jager, *Doomsday*, COMPUTERWORLD, Sept. 6, 1993, at 105, 105.

[4] *See, e.g.,* ALAN SIMPSON, UNDERSTANDING MICROSOFT ACCESS 127 (1993) (noting date formats must conform to the date format MM/DD/YY); de Jager, *supra* note 3, at 105.

[5] *See* Levy & Hafner, *supra* note 2, at 54.

[6] *See* Steven H. Goldberg*, Managing "Year 2000" Business and Legal Risks for Hospitals and Health Care Systems*, COMLINKS (visited Mar. 24, 1998) <http://www.comlinks.com/legal/gold1.htm>. More specifically, "it costs about 10 cents per month to lease one megabyte of storage space today." *Id.* "[I]n 1963 the cost was $175/month (in 1995 dollars)." *Id.*

[7] *See* Levy & Hafner, *supra* note 2, at 54.

[8] If digits are treated as characters for storage purposes, storing two extra digits for a database of 10,000 records would require approximately an extra 20,000 bytes, or 20K of storage space. Interview with Steven Eschrich, System Manager at the Department of Health Care Policy, Harvard Medical School, in Boston, Mass. (Nov. 1, 1996). For the early computer that had only 640K of memory, the extra digits would take up a full 3% of the computer's total memory. *See id.*

last two digits of the year in the date would be used for data storage or computational purposes.

6. Not only was the storage of dates limited to two digits, but the actual entry of date data was also limited.[9] While this decision was cost-efficient at the time, this Bug will cause serious ramifications if it is not corrected before the year 2000.[10] While the effects of the Y2K Bug are most apparent in legacy systems and software,[11] the use of the old date standard is so prevalent that even software developed and released within the last few years is not immune.[12]

A. Examples

7. The problems related to the Y2K Bug are most apparent in computations where dates are subtracted or compared. Here are a few examples:

1. Happy Birthday[13]

8. How does a typical software program calculate a person's age? The standard method used to determine age is to subtract the person's birth year from the current year. Suppose, for example, that someone was born in 1970. To calculate that person's age in 1997, the computer will subtract 70 (1970) from 97 (1997) to determine a correct age of 27 years old. Now fast-forward to 2005. Eight years have passed so the age of the person would now be 35. What will a computer with the Y2K Bug calculate the age to be? The computer will subtract 70 (1970) from 05 (2005) to come up with an age of -65. Miscalculations of age will cause the most damage where age calculations are relied on to determine product quality,[14] set

[9] For example, many of the screens a user is presented with when entering data only allow data entry of the last two digits. *See* SIMPSON, *supra* note 4, at 127. As a result, it is necessary to change the user interface to programs as well.

[10] A study by the Gartner Group estimates that it will cost as much as $600 billion to correct the Y2K Bug. *See* Levy & Hafner, *supra* note 2, at 54.

[11] The term "legacy" refers to old software or systems that continue to be used because of the high costs of replacement or redesign. *See Free On-line Dictionary of Computing* (last modified Aug. 4, 1996) <http://wfn-shop.princeton.edu/foldoc/cgi-script?query=legacy>.

[12] The first year 2000 lawsuit involves software engineered in 1995 and was unable to process credit cards that had an expiration date in the year 2000. *See* Compl. of Produce Palace Int'l at ¶4, Produce Palace Int'l. v. Tec-America Corp., No. 97-3330-CK (Mich. Cir. Ct., filed June 12, 1997); *see also* Emily Fitzloff & Eric Hammond, *Year-2000 Bug Bites Novell*, INFOWORLD, Oct. 20, 1997, at 1, 1 (stating that as of October 1997, Novell Netware was not year 2000 compliant).

[13] For a similar example, see de Jager, *supra* note 3, at 105.

[14] At one company in Britain, a computer with the Y2K Bug mistakenly ordered the

product maintenance schedules,[15] or compute retirement benefits programs.[16] For example, some Social Security benefits cannot be collected until an individual meets the age eligibility requirement.[17] Age eligibility calculations performed on a system with the Y2K Bug would return incorrect age calculations as demonstrated above and, thus, reject benefit applications. Recognizing the immensity of these problems, the Social Security Administration is devoting a substantial amount of time and resources to fix its Y2K Bug problems.[18]

2. Who Came First

9. Many businesses store and arrange data in sequential date order. Suppose a business has several data records that it needs to sort sequentially.[19] Assuming that the records have dates of 1960, 1900, and 1970, the computer will sort the records by reading the last two digits of the date and return a sequence of 19**00**, 19**60**, and 19**70**. If the business adds a new record in 2010, what happens to the sequence? Because of the Y2K Bug, the computer will incorrectly sort the records, returning the result of 19**00**, 20**10**, 19**60**, 19**70**. Sorting errors of this kind will cause the most problems in order-entry systems, payment systems, and in reservation databases that sort records by date.[20]

destruction of tons of corned beef when it calculated the food to be over 100 years old. *See* Levy & Hafner, *supra* note 2, at 56.

[15] The United States Department of Defense is particularly sensitive to errors in age calculation. For example, if the Army ordered new equipment after the year 2000, it could be identified as 99 years old by computers with the Y2K Bug and tagged, automatically, for destruction. *See* Sharon Machlis, *GAO Slams U.S. Army's Year 2000 Preparation*, COMPUTERWORLD, Oct. 6, 1997, at 3, 3.

[16] *See* Joanne Wojcik, *Employers Warned Not to Wait on Year 2000: Mistakes Could Be Costly,* BUS. INS., Mar. 24, 1997, at 3, 3.

[17] *See* 42 U.S.C. § 402(a) (1994).

[18] The Social Security Administration has been working on Y2K Bug issues for the past eight years. *See* Levy & Hafner, *supra* note 2, at 58. It is one of a select few government agencies and departments that is on track to complete its Y2K Bug conversions before the turn of the century. *See id.*

[19] For a similar example, see de Jager, *supra* note 3, at 108.

[20] *See* Lynda Radosevich, *Millennium Bug Already Taking Its Toll*, INFOWORLD, Jan. 12, 1997, at 1, 19 (discussing the problems that the Y2K Bugs will cause, beginning January 1, 1999, for reservation systems that accept reservations one year in advance).

3. Mortgage Snafu

10. Calculating interest on a loan term is normally accomplished by subtracting the starting date of the loan from the ending date (or current date). If the starting date of the loan was 19**90** and the ending date was 19**95**, the computer would calculate interest for a term of five years (95-90). However, if the starting date of the loan is 19**97** and the ending date is 20**05**, a computer with the Y2K Bug will incorrectly calculate the loan term and, consequently, the interest due. The computer will calculate the loan term to be -93. Either the calculation will fail because the computation returns a negative number or the program will ignore the negative and calculate interest for a term of 93 years, instead of the correct term of eight years. At an interest rate of 7.5% on an eight-year loan of $100,000, the consumer pays a total of $33,285.16 in interest. If the period is extended to 93 years because of an incorrect calculation by a computer with the Y2K Bug, the consumer would pay total interest of $598,167.10 on the same $100,000 loan.[21] Every outstanding loan with a period extending to or beyond the year 2000 is subject to such incorrect interest calculations unless each lending institution corrects its system in time.

11. This problem is not new to the banking industry because the industry's thirty-year mortgages forced it to address Y2K Bug problems in the 1970s.[22] The industry, however, has only recently launched a full-scale campaign to address Y2K Bug problems in systems that have not yet been affected by cross-century calculations.[23] Thus, home mortgages[24] and savings account balances[25] are susceptible to incorrect interest calculations because of the Y2K Bug.

[21] The figures in this example were calculated using the Loan Manager tool in Microsoft Excel for Windows 95 version 7.0. Interest calculations were computed using compound interest.

[22] *See* Brian Robbins, *Year 2000 Issue: Not a Race for One Institution to Win*, BANK SYS. & TECH., Sept. 1, 1997, *available in* 1997 WL 11805457.

[23] It has been estimated that only 40% of banks in the United States had started a serious evaluation of their Y2K Bug problems as of March 1997. *See Bank and the Year 2000 Computer Problem: Hearing Before the Subcomm. on Fin. Serv. and Tech. of the Senate Banking, Hous. and Urban Affairs Comm.*, 105th Cong. (1997) (prepared testimony of Jeff Jinnett, President of LeBoeuf Computing Technologies, L.L.C. and Of Counsel to LeBoeuf, Lamb, Greene & McRae, L.L.P.) [hereinafter *Jinnett Testimony*] (discussing effects of the Y2K Bug on the financial industry).

[24] *See id.* (discussing the effects on consumers of Y2K Bug-related bank problems including incorrect interest calculations for savings accounts). In the first eleven months of 1997, approximately 749,000 homes were sold in the United States. *See* United States Bureau of the Census, *New One-Family Houses Sold: November 1997* (visited Jan. 12, 1997) <http://www.census.gov/prod/3/97/pubs/c25-9711.pdf>.

[25] *See Jinnett Testimony*, *supra* note 23 (discussing the effects on consumers of Y2K Bug-related bank problems including incorrect interest calculations for savings accounts).

4. Happy New Year

12. It is 11:51 p.m., December 31, 1999. An online countdown to the new century is taking place and a user logs on to an Internet provider to join the fun. The user stays online until 12:02 a.m. on January 1, 2000, a total of twelve minutes. The formula for calculating the billable time online subtracts the login year (19**99**) from the logout year (20**00**).[26] The billable time for this session is twelve minutes and 99 years!

13. As these examples illustrate, the Y2K Bug has the potential to disrupt many transactions that many people currently take for granted. As a result, industry analysts are concerned about the devastating ramifications the Y2K Bug could have on businesses that depend on accurate date calculations.[27] While some have been proactive in their development and implementation of solutions,[28] many have not taken steps to address the issue.[29] Within the computer industry, leading companies are beginning to address the issue,[30] and dozens of companies have emerged that specialize in providing solutions to the Y2K Bug.[31]

[26] The main problem for telephone companies is non-compliant billing systems that could incorrectly calculate the billable time for any voice or data transmission that starts in 1999 and ends in 2000. *See* Matt Hamblen, *Bomb Ticking for Voice, Data Nets*, COMPUTERWORLD, Oct. 27, 1997, at 2 (describing the Y2K Bug pitfalls in the global telephone network).

[27] For example, some experts have predicted that the fallout from the Y2K Bug problem has the potential to lead to a global recession. *See* Laurence Zuckerman, *Many Reported Unready to Face Year 2000 Bug*, N.Y. TIMES, Sept. 25, 1997, at D3; Robert L. Scheier, *Economist Predicts Y2K-Based Recession*, COMPUTERWORLD, Sept. 22, 1997, at 12 [hereinafter *Y2K Based Recession*].

[28] For example, Merrill Lynch has an eighty member division working in shifts, twenty-four hours a day, seven days a week to correct its Y2K Bug problems. *See* Levy & Hafner, *supra* note 2, at 57. Similarly, BankBoston has been working since 1995 to correct 60 million lines of code. *See id.* at 57-58. It currently has forty full-time employees devoted to the project. *See id.*

[29] A full 30% of companies worldwide have yet to begin addressing the Y2K Bug problem within their organizations. *See* Zuckerman, *supra* note 27. Similarly, of companies in the United States, 60% are still evaluating the extent of their Y2K Bug problems, 25% are in the process of fixing the code, and only 10% are testing the corrected code. *See* Robert L. Scheier, *As Clock Ticks, Many Still Avoid Year 2000 Work*, COMPUTERWORLD, Sept. 8, 1997, at 10 [hereinafter *Clock Ticks*].

[30] *See, e.g.*, Microsoft Corp., *Year 2000 Date Transition Issue* (visited Mar. 23, 1998) <http://www.microsoft.com/ithome/topics/year2k/2kpc/related/2khome.htm> (addressing Microsoft Corporation's strategies for dealing with the Y2K Bug); *Year 2000 Information Center* (visited Mar. 23, 1998) <http://www.year2000.com/y2k-main1.html> (collecting information on the Y2K Bug).

[31] Solutions 2000 International is one example. *See Year 2000 Services* (visited Mar. 23, 1998) <http://www.solutions2000.com/index.html>. It provides services to companies worldwide that have year 2000 problems. *See id.*

14. With the turn of the century rapidly approaching, the race is on to correct the Y2K Bug before time runs out.[32] Industry analysts predict that companies that have not yet begun to remedy their Y2K Bug issues do not have enough time[33] or money to put an effective solution in place.[34] Even the United States government, which arguably faces the biggest challenge in converting its systems to year 2000 compliance,[35] is in danger of not meeting the deadline.[36] Analysts expect a mass of litigation, in which users will look to the vendors or manufacturers of the Y2K Bug-affected software for compensation for potentially devastating losses.[37]

[32] A recent study by the Gartner Group estimates that 50% of all businesses will not have their systems ready by the turn of the century. *See* Levy & Hafner, *supra* note 2, at 58.

[33] It can take years just to analyze a system and determine the extent of the year 2000 problems present. *See id.* at 54. It takes even longer to actually correct those problems. *See id.* After the system is fixed, it must be tested for accuracy, a process that itself takes a substantial amount of time. *See id.*; *see also* Robert L. Scheier, *Testing Can't Wait*, COMPUTERWORLD, Oct. 20, 1997, at 79, 79 (stating that testing will take from 50-70% of the time and effort of Y2K Bug conversions).

[34] It was estimated in 1993 that Fortune 50 companies would each spend approximately 35 to 40 cents per line of code to correct the year 2000 problem in their systems, adding up to between $50 million and $100 million for each company. *See* de Jager, *supra* note 3, at 108. By 1997, the cost of correcting code had risen to between $1.10 and $1.60 per line of code. *See* Thomas Hoffman, *Silver Bullet Pitched, Dissed*, COMPUTERWORLD, Jan. 6, 1997, at 1, 1. Estimates of the cost of converting the United States government's code vary. *See Year 2000 Price Tag*, COMPUTERWORLD, Feb. 17, 1997, at 63 (contrasting an Office of Management & Budget report estimating the cost to be $2.3 billion with outside industry analysts who estimate the cost to be about $30 billion).

[35] The government has determined that in the executive branch alone there are 8562 mission critical systems that are believed to be affected by the Y2K Bug. *See* Rajiv Chandrasekaran & Stephen Barr, *Four Agencies to Be Barred From Buying Computers Until 2000 Glitch Fixed*, WASH. POST, Sept. 14, 1997, at A10 (discussing how the Department of Agriculture, Department of Transportation, Department of Education, and the Agency for International Development will be prohibited from buying any new machines or software until they fix systems that control critical services including student loan applications and benefit checks).

[36] During joint hearings of the House Science Committee and the Government Reform and Oversight Committee, one representative expressed concern that the government is unlikely to complete all of its Y2K Bug conversions by the turn of the century. *See Year 2000 "Computer Problem": Joint Hearing Before the House Science Comm. and the Gov't. Reform and Oversight Comm.*, 105th Cong. (1997) (statement of Constance A. Morella, Chairwoman, Subcomm. on Tech.); *see also* Robert L. Scheier, *Congress Examines Date-Change Progress*, COMPUTERWORLD, Feb. 24, 1997, at 28.

[37] Estimates suggest that lawsuits involving the Y2K Bug problem will total approximately one trillion dollars. *See* Wylie Wong, *Grocer Registers Year 2000 Suit*, COMPUTERWORLD, Aug. 18, 1997, at 6 (citing one estimate of the total amount that is expected to be spent on Y2K Bug-related litigation).

15. The Y2K Bug is unique among computer bugs.[38] First, a bug is generally unknown to the vendor or developer at the time the product is released.[39] If the defect is known, steps are taken to correct the defect before the product is released.

16. Second, unlike other bugs, the Y2K Bug is not an "unwanted" or "unintended" defect in the product.[40] In fact, the Bug resulted from a conscious design decision to program dates in a way that reduced the amount of necessary storage space.[41] Despite knowledge of the intrinsic limitation in the date mechanisms employed, developers continued this method of date programming because it was the most efficient way to store and calculate dates. Finally, few serious measures to remedy the problem have been taken although analysts continue to discuss the impending crisis that the Bug will bring.[42]

B. Industry Developments and Litigation

17. The market, both inside and outside the computer industry, has responded to concerns about the ramifications of the Y2K Bug.[43] In particular, firms are attempting to shield themselves from the inevitable fallout that will occur when

[38] The first computer bug was "discovered" by a technician working with the Harvard Mark II machine. *See Free On-Line Dictionary of Computing* (visited May 23, 1998) <http://wfn-shop.princeton.edu.foldoc/cgi-script?query=bug>. Admiral Grace Hopper, a pioneer in the computer industry and inventor of the COBOL programming language, coined the phrase "bug" in response to this incident. *See id.*

[39] A bug is defined as "an unwanted and unintended property of a program or piece of hardware, especially one that causes it to malfunction." *Id.*; *see also* MERRIAM WEBSTER'S COLLEGIATE DICTIONARY 149 (10th ed. 1993) (defining a bug as "an *unexpected* defect, fault, flaw, or imperfection") (emphasis added).

[40] In making these date calculations, the computers are performing as designed and how the programmers intended the dates to be calculated. *See* de Jager, *supra* note 3, at 105.

[41] *See id.*

[42] Articles discussing the Y2K Bug problem have been published in a wide variety of publications. *See, e.g.*, Joseph Pryweller, *Car Makers Prepare for Year-2000 Bug*, PLASTICS NEWS, Sept. 1, 1997, *available in* 1997 WL 8489811; Rick Telberg, *Start Worrying: The Year 2000 Bug Is for Real*, ACCOUNTING TODAY, June 2, 1997, at 6; Tim Wilson, *Protect Your Net*, COMM. WK., May 5, 1997, at 1. According to the editor of one industry magazine, *Computerworld*, the magazine has published more than 1000 articles discussing the Y2K Bug problem since 1984. *See* Paul Gillin, *Y2K Pledge*, COMPUTERWORLD, Sept. 22, 1997, at 2.

[43] Many of the major computer software and hardware companies have information sites on the World Wide Web devoted to the discussion of their companies' responses to the Y2K Bug crisis. *See, e.g.*, IBM, *IBM Year 2000 Home Page* (visited Mar. 23, 1998) <http://www.ibm.com/IBM/year2000/>; Sun Microsystems, *Welcome to Sun's Year 2000 Information Site!* (visited Mar. 23, 1998) <http://www.sun.com/y2000/>.

systems fail to operate after the turn of the century.[44] Practitioners suggest that the best way to protect a company against year 2000 non-compliance is to obtain written guarantees[45] from every software vendor that both the new and currently used software it provides will be year 2000 compliant.[46]

18. Another alternative is to buy "business disaster insurance" to cover unforeseen business disruptions that may occur as a result of year 2000 conversions.[47] Policies limit coverage to between $100 million and $200 million and cover failures due to in-house programming mistakes or problems caused by third parties.[48] A company can acquire additional coverage for its officers and directors to protect them against shareholder suits.[49] The acquisition of this type of insurance may increase in importance as several insurance companies consider dropping coverage for losses related to Y2K Bug problems.[50]

19. While the majority of Y2K Bug lawsuits are likely to occur after the turn of the century, the first lawsuit for damages resulting from computer-related year 2000 problems was filed in June 1997.[51] Produce Palace International, a Michigan produce store, is suing the developers of its cash register software for damages caused when the cash registers failed to process credit cards with an expiration date of 2000.[52] Between April 30, 1996, and May 6, 1997, Produce Palace owners say that

[44] *See* Levy & Hafner, *supra* note 2, at 58 (suggesting that 50% of companies world-wide may not be year 2000 compliant at the turn of the century).

[45] For an example of a year 2000 warranty, see Timothy J. Feathers, *Year 2000 Warranty* (visited Feb. 2, 1997) <http://www.year2000.com/archive/warranty.html>.

[46] BankBoston is an example of a corporation that has aggressively sought written guarantees from its software providers. *See* Thomas Hoffman, *Lawyer Disputes Year 2000 Warranties*, COMPUTERWORLD, Mar. 17, 1997, at 24. BankBoston has received a written guarantee from a vendor that its software will function in the year 2000. *See id.*

[47] Several insurance companies, including Marsh & McLennan, Inc., American International Group, and Minet, Inc., are insuring companies against Y2K Bug-related business problems. *See* Thomas Hoffman, *CIOs Wary of Year 2000 Insurance*, COMPUTERWORLD, Feb. 3, 1997, at 1, 1.

[48] *See id.* at 1, 103.

[49] *See id.* at 103.

[50] *See* Stacy Shapiro, *Insurers May Try to Exclude Cover for 'Year 2000' Liabilities*, BUS. INS., June 16, 1997, at 25 (discussing movement in the insurance industry to limit or exclude coverage for Y2K Bug-related losses).

[51] *See* Compl. of Produce Palace Int'l, Produce Palace Int'l v. Tec-America Corp., No. 97-3330-CK (Mich. Cir. Ct., filed June 12, 1997). *See generally* Wong, *supra* note 37, at 6 (discussing the potential ramifications of Y2K Bug litigation).

[52] Many credit card companies recognized that cards with an expiration date of 2000 created

their cash registers crashed 105 times during attempts to charge sales to credit cards with year 2000 expiration dates.[53] In its complaint, Produce Palace seeks recovery under several theories, including breach of contract,[54] breach of warranty,[55] and misrepresentation.[56]

20. This lawsuit is interesting because it challenges the assumption that only legacy software is afflicted by the Y2K Bug.[57] Much attention has focused on legacy systems as the source of Y2K Bug problems. The Y2K Bug problem is indeed immense on many legacy systems,[58] but the practice of two-digit date functions has long been the standard in the computer industry.[59] The software and hardware at issue in *Produce Palace International v. Tec-America Corp.* was installed in 1995, however, demonstrating that even relatively new software has the potential for causing year 2000 problems.[60]

problems and stopped issuing cards with that date until the Y2K Bug problems were corrected. *See* Wong, *supra* note 37, at 6. Both MasterCard and Visa say that they have corrected their Y2K Bug problems and both companies planned to begin distributing credit cards with year 2000 expiration dates by October 1, 1997. *See* Thomas Hoffman & Wylie Wong, *Visa, MasterCard Say Year 2000 Problem Is Fixed*, COMPUTERWORLD, Aug. 25, 1997, at 10. Despite assurances by Visa that 99.7% of the 14 million locations accepting its cards were Y2K compliant, validation systems are still rejecting credit cards with an expiration date of 2000. *See* Radosevich, *supra* note 20, at 19.

[53] *See* Erich Luening & Mike Ricciuti, *First Year 2000 Case Goes to Court* (visited Nov. 19, 1997) <http://www.news.com/News/Item/0,4,13213,00.html>; *see also* Compl. of Produce Palace Int'l at ¶ 11, *Produce Palace* (No. 97-3330-CK) (describing how the cash registers were down over 100 of the first 500 days the system was in place).

[54] *See* Compl. of Produce Palace ¶¶ 79-84, *Produce Palace* (No. 97-3330-CK).

[55] *See id.* ¶¶ 22-34.

[56] *See id.* ¶¶ 69-78.

[57] *See* Wong*, supra* note 37, at 6.

[58] *See* Levy & Hafner, *supra* note 2, at 54 (noting that in many mainframe programs, the date appears in approximately every 50 lines of code).

[59] *See id.*

[60] *See* Compl. of Product Palace Int'l ¶ 4, *Produce Palace* (No. 97-3330-CK).

III. THEORIES OF LIABILITY[61]

21. As the previous section suggests, the repercussions from Y2K Bug system failures are bound to have far-reaching effects.[62] Inevitably, those who suffer from Y2K Bug losses will be looking to the vendors and developers of their software and systems to compensate for these losses.[63] The legal theory or theories on which a Y2K Bug suit is based will be important to the ultimate determination of the litigation. For most Y2K Bug suits, there are five different theories a litigant could assert as the basis for liability. These include breach of warranty,[64] negligence, computer malpractice, misrepresentation, and strict liability. The availability of each theory will depend, in a large degree, on the particular facts surrounding the acquisition of the system and software. To that end, most business transactions are likely to be made under a specific contract negotiated between the vendor and the user. For these cases, this contract will determine what rights the user has against the vendor. Since these contracts are likely to differ widely in their terms, there is no section of this Note devoted specifically to breach of contract claims. However, it is important to remember that for many transactions, the terms of the contract will govern. In addition, most contracts will attempt to disclaim all available warranties. Such disclaimers are almost universally upheld[65] and are an effective mechanism for preventing any damage recovery.[66]

22. The purpose of the following section is to give the reader an overview of each available legal theory along with the potential obstacles to recovery that each theory presents.

[61] For discussions of the legal theories of liability for Y2K Bug problems, see generally Gary E. Clayton et al., *The Year 2000 Headache "Two Thousand Zero-Zero. Party's Over. Oops, Out of Time.",* 28 TEX. TECH. L. REV. 753 (1997); John F. Cooney, *Who Will Pay for the Year 2000?,* COMPUTER LAW., Sept. 1997, at 1; Robert G. Gerber, *Computers and the Year 2000: Are You Ready?*, 30 J. MARSHALL L. REV. 837 (1997); Stephen D. Kahn, *The Millennium Bug: Legal Ramifications of Year 2000 Compliance,* 9 J. PROPRIETARY RTS. 6 (1997).

[62] *See* Levy & Hafner, *supra* note 2, at 54; *Y2K-Based Recession*, *supra* note 27, at 12.

[63] *See* Wong, *supra* note 37, at 6 (estimating that Y2K Bug lawsuits could cost as much as one trillion dollars).

[64] These warranties may be express contractual warranties or implied warranties under the Uniform Commercial Code.

[65] *See, e.g.*, Laidlaw Envtl. Servs. (TOC), Inc. v. Honeywell, Inc., 966 F. Supp. 1401, 1412-13 (D.S.C. 1996) (upholding contractual disclaimer of warranties); Hi Neighbor Enters., Inc. v. Burroughs Corp., 492 F. Supp. 823, 826 (N.D. Fla. 1980) (upholding disclaimer of warranties).

[66] Many software contracts limit recoverable damages to cost of repair and exclude any liability for consequential damages. *See, e.g.,* Mecklenburg Furniture Shops v. MAI Sys. Corp., 800 F. Supp. 1328, 1331 (D.N.C. 1992).

A. Uniform Commercial Code

1. Application

23. The Uniform Commercial Code ("UCC") is a model statute that governs commercial transactions. Article 2 of the UCC deals specifically with the sale of goods.[67] Most states have adopted some version of the UCC, including Article 2.[68] As a result, its provisions will govern many software transactions, particularly software purchased over the counter.[69] There is a growing consensus that the UCC is applicable to computer software transactions, because the software is a "good" under the UCC.[70] Goods are defined as "all things (including specially manufactured goods) which are movable at the time of identification to the contract for sale."[71]

[67] *See* U.C.C. § 2-102 (1995) ("Unless the context otherwise requires, this article [Article 2] applies to transactions in goods.").

[68] The UCC, including Article 2, has been adopted in 49 states (every state except Louisiana), the District of Columbia, and the Virgin Islands. *See* William A. Schnader, *A Short History of the Preparation and Enactment of the Uniform Commercial Code*, 22 U. MIAMI L. REV. 1, 10 (1967) (discussing the history of state adoption of the UCC). For a list of state variations in the UCC text, see UCC REPORTING SERVICE, STATE UCC VARIATIONS (Pike & Fischer, Inc. 1997).

[69] For a discussion of the treatment of computer software as a good under the UCC, see generally Bonna Lynn Horovitz, Note, *Computer Software as a Good Under the Uniform Commercial Code: Taking a Byte Out of the Intangibility Myth*, 65 B.U. L. REV. 129 (1985) (arguing that software should be treated as a good under the UCC).

[70] *See, e.g.,* ProCD, Inc. v. Zeidenberg, 86 F.3d 1447, 1450-51 (7th Cir. 1996) (stating that computer software is a good within the meaning of Article 2); Advent Sys. Ltd. v. Unisys Corp., 925 F.2d 670, 675 (3d Cir. 1991) (computer software a good); RRX Indus., Inc. v. Lab-Con, Inc., 772 F. 2d 543, 546 (9th Cir. 1985) (software package a good); Triangle Underwriters, Inc. v. Honeywell, Inc., 604 F.2d 737, 742-43 (2d Cir. 1979) (purchase of turnkey computer system was the sale of a good); NMP Corp. v. Parametric Tech. Corp., 958 F. Supp. 1536, 1542 (N.D. Okla. 1997) (licensing agreement is a sale of goods); Colonial Life Ins. Co. v. Electronic Data Sys. Corp., 817 F. Supp. 235, 239 (D.N.H. 1993) (license for computer software covered by Article 2); Harper Tax Servs., Inc. v. Quick Tax Ltd., 686 F. Supp. 109, 110 (D. Md. 1988) (computer software designed for preparation of tax returns a good); Systems Design & Management Info., Inc. v. Kansas City Post Office Employees Credit Union, 788 P.2d 878, 882 (Kan. Ct. App. 1990) (computer software is a good); Communications Groups, Inc. v. Warner Communications, Inc., 527 N.Y.S.2d 341, 344 (N.Y. Civ. Ct. 1988) (lease of computer software within Article 2); Schroders, Inc. v. Hogan Sys., Inc., 522 N.Y.S.2d 404, 406 (N.Y. Sup. Ct. 1987) (license for computer software is a good); Barazzotto v. Intelligent Sys., Inc., 532 N.E.2d 148, 150 (Ohio Ct. App. 1987) (software package a good). *But see* Computer ServiCenters, Inc. v. Beacon Mfg. Co., 328 F. Supp. 653, 655 (D.S.C. 1970) (agreement for performance of data processing services outside the scope of Article 2); Data Processing Serv., Inc. v. L.H. Smith Oil Corp., 492 N.E.2d 314, 318-19 (Ind. Ct. App. 1986) (contract for development of electronic data processing system not within Article 2).

[71] U.C.C. § 2-105(1) (1995).

Because computer software disks "are movable," the application of the UCC is proper to computer cases.

24. Before looking at specific provisions of the UCC, it is necessary to first address the software industry's attempts to limit liability for defective software products.[72] Software is inherently unstable in that defects are continually discovered in released software products.[73] Software bugs are not only prevalent,[74] but are also expensive.[75] As a result, manufacturers and vendors attempt to limit their potential liability from performance defects through a variety of mechanisms. In negotiated transactions, the contract normally contains provisions limiting or excluding warranties and limiting liability.[76] In individual consumer transactions, liability is limited through the use of "shrink-wrap" licenses.[77] A shrink-wrap license is typically printed on or referenced by[78] the plastic shrink-wrap bag in which software diskettes are shipped.[79] Typically, shrink-wrap licenses state that the consumer, by opening the package containing the diskettes, consents to and accepts

[72] *See generally* Mark Lemley, *Intellectual Property and Shrink-Wrap Licenses*, 68 S. CAL. L. REV. 1239 (1995) (discussing the arguments for and against enforcing shrink-wrap licenses and concluding that they should not be enforceable).

[73] It is recognized that the quality of software products is poor and does not seem to be getting better. *See* Gary H. Anthes, *Quality?! What's That?*, COMPUTERWORLD, Oct. 13, 1997, at 75, 75.

[74] Software defects affect all sectors of our society. Two examples are: a software defect at the British National Lottery that prevented the identification of jackpot winners and a software problem at a brokerage firm that briefly credited customer accounts with $19 million each. *See id.* at 76. A defective software program controlling a baggage-handling system delayed the opening of the Denver International Airport for 16 months. *See* Miryam Williamson, *Quality Pays*, COMPUTERWORLD, Aug. 18, 1997, at 78, 78.

[75] The United States Department of Defense estimates that fully half of its software costs are attributable to correcting bugs. *See* Bill Laberis, *Software Flaws, Lemon Laws*, ENT, Oct. 8, 1997, at 70. The delay at the Denver International Airport cost a million dollars per day. *See* Williamson, *supra* note 74, at 78.

[76] For examples of contracts that limit warranties and liability, see generally Laidlaw Envtl. Servs. (TOC), Inc. v. Honeywell, Inc., 966 F. Supp. 1401, 1405-06 (D.S.C. 1996); Hi Neighbor Enters, Inc. v. Burroughs Corp., 492 F. Supp. 823, 824-26 (N.D. Fla. 1980).

[77] *See* Lemley, *supra* note 72, at 1241-42.

[78] For example, the diskettes for Microsoft Office for Windows 95 were shrink-wrapped inside the software box. Printed on the shrink-wrap was the following statement: "ATTENTION! Use of the software program on the enclosed disks is subject to the terms of the Microsoft License Agreement printed on the license card . . ., or in the user's documentation. You should not open this packet until you have read the Microsoft License Agreement. By opening this packet, you signify that you have read the Microsoft License Agreement and accept its terms." [hereinafter Shrink-Wrap Text] [on file with the *Boston University Journal of Science & Technology Law*].

[79] *See* Lemley, *supra* note 72, at 1241.

the terms of the vendor's license agreement.[80] Through these license agreements, the vendors limit and exclude the warranties that would otherwise be available under the UCC[81] in an effort to limit their potential liability[82] should the software not perform correctly.[83]

25. Courts are split on the enforceability of software shrink-wrap licenses.[84] The courts that have refused to enforce shrink-wrap licenses focus on the buyers' lack of consent to the terms in the shrink-wrap licenses. Although this argument is most persuasive in the context of individual consumers who buy products for home use, it has been successfully employed by more sophisticated consumers as well. For example, in *Step-Saver Data Systems, Inc. v. Wyse Technology,* the user was a sophisticated business that had extensive dealings with the vendor before the final sale was completed.[85] The court analyzed the case under UCC § 2-207[86] and held that the shrink-wrap license contained in the final software was invalid and not a part of the contract because each party did not expressly agree to its terms.[87] An individual consumer transaction will not likely resemble the facts of *Step-Saver*,

[80] *See* Shrink-Wrap Text, *supra* note 78.

[81] U.C.C. § 2-316(2)-(4) (1995) allows a party to limit the warranty of merchantability and the warranty of fitness for a particular use. *See id.*

[82] For example, the Microsoft Office 95 license agreement states that: i. the software has a limited 90 day warranty; ii. customer remedies are limited to replacement of the product or the price paid for the defective product; and iii. all other warranties are disclaimed.

> "To the maximum extent permitted by applicable law, Microsoft and its suppliers disclaim all other warranties, either express or implied, including, but not limited to, implied warranties of merchantability and fitness for a particular purpose, with regard to the software product" [hereinafter License Agreement] [on file with the *Boston University Journal of Science & Technology Law*].

[83] *See generally* Paul S. Hoffman, *Software Warranties and the Uniform Commercial Code*, 6 J. PROPRIETARY RTS. 7 (1994) (discussing the effect of the UCC on software license warranties). *See also* License Agreement, *supra* note 82 (limiting any recovery to price paid or replacement).

[84] *Compare* Step-Saver Data Sys., Inc. v. Wyse Tech., 939 F.2d 91, 102-04 (3d Cir. 1991) (invalidating shrink-wrap license) *and* Arizona Retail Sys., Inc. v. Software Link, Inc., 831 F. Supp. 759, 766 (D. Ariz. 1993) *with* ProCD, Inc. v. Zeidenberg, 86 F.3d 1447, 1449 (7th Cir. 1996) (shrink-wrap license valid unless unconscionable).

[85] *Step-Saver,* 939 F.2d at 93-94.

[86] "[C]onduct by both parties which recognizes the existence of a contract is sufficient to establish a contract for sale although the writings of the parties do not otherwise establish a contract." U.C.C. § 2-207(3) (1995). In that situation, the contract terms will be those that the parties agreed upon, supplemented by terms provided in Article 2. *See id.*

[87] *See Step-Saver*, 939 F.2d at 99, 102-04.

however, because the majority of individual consumer transactions will involve a user going to a retail store and selecting pre-packaged software off the shelf.[88]

26. On the other hand, one court has held that shrink-wrap licenses are enforceable[89] in individual consumer transactions.[90] Comparing such software purchases to the purchase of airline tickets or radios, the court found that the buyer accepted the terms of the contract, in the form of the shrink-wrap license, by failing to reject the software after the buyer had an opportunity to read the license terms.[91]

27. Unlike businesses, consumers do not usually engage in extensive dealings with vendors. The typical individual software transaction begins with the consumer going to a computer store or to a consultant for advice on which program or programs best serves his or her needs. The consumer takes the software box[92] off the shelf, pays for it, and then goes home.[93] It is not until the consumer actually starts to install the software that he or she becomes aware of the existence of a license agreement. No opportunity is given to a consumer to read the license agreement prior to the point of installation.[94] At this point, it is unrealistic for a consumer to return the product after the license agreement is actually discovered. As a result, the shrink-wrap license, including any limitations on warranties it contains, should not be enforceable as a contract against an unsuspecting consumer who had neither the opportunity nor the bargaining power[95] to change any of the terms of the license agreement.[96]

[88] *See generally ProCD*, 86 F.3d at 1450.

[89] *See id.* at 1451-53.

[90] *See id.* at 1452 (distinguishing *Step-Saver* as being a "battle-of-the-forms case," not a consumer transaction).

[91] *See id.* at 1451-52 (noting that in these transactions it is typical for the terms of the agreement to be given to the consumer after purchase of the item).

[92] The actual software box is usually shrink-wrapped itself, making it impossible for a consumer to access any agreements inside.

[93] *See* ProCD, Inc. v. Zeidenberg, 908 F. Supp. 640, 650 (W.D. Wis. 1996) (describing the typical software transaction where the customer goes to a retail store to purchase the software).

[94] *See id.* at 651.

[95] *See id.* at 655.

[96] Unconscionability is another ground that could be used to invalidate vendor software license agreements. Under the UCC, courts have the power to invalidate a contract or a contract clause that is "unconscionable at the time it was made." U.C.C. § 2-302 (1995). Even the Seventh Circuit would invalidate a shrink-wrap license if its terms were deemed to be unconscionable. *See ProCD*, 86 F.3d at 1449. The typical warranty disclaimer and damage limitation provisions are not likely to rise to the level of unconscionability. *See* Harper Tax Servs., Inc. v. Quick Tax Ltd., 686 F. Supp.

28. The enforceability of a shrink-wrap license is important because it will determine whether recovery under any of the UCC warranty provisions will be permitted for transactions conducted under a shrink-wrap license. One of the salient features of the shrink-wrap license is that it limits and/or disclaims the default warranties provided by the UCC.[97] If the shrink-wrap license is an unenforceable contract, the standard UCC warranties will probably apply and any disclaimer will be invalid. If the shrink-wrap license is enforceable between the parties, the extent of any warranty protection is an issue of interpretation under the relevant contract language.

2. Specific Provisions

a. UCC § 2-313: Express Warranties by Affirmation, Promise, Description or Sample[98]

29. The UCC describes how a seller creates express warranties. Section 2-313(1) states that:

> (a) Any affirmation of fact or promise made by the seller to the buyer which relates to the goods and becomes part of the basis of the bargain creates an express warranty that the goods shall conform to the affirmation or promise.
> (b) Any description of the goods which is made part of the basis of the bargain creates an express warranty that the goods shall conform to the description.
> (c) Any sample or model which is made part of the basis of the bargain creates an express warranty that the whole of the goods shall conform to the sample or model.[99]

30. This section establishes three primary ways of creating an express warranty: by promise, by description, or by sample.[100]

31. In a computer software case, express warranties are typically given by sales representatives during the course of negotiations, stated in advertisements

109, 112 (D. Md. 1988).

[97] *See* License Agreement, *supra* note 82.

[98] *See* U.C.C. § 2-313(1) (1995).

[99] *Id.*

[100] For a discussion of software samples as the basis for warranty in computer transactions, see Dennis S. Deutsch, *The "Demo" as the Basis of the Fraud and Breach of Contract Claim*, COMPUTER LAW., May 1991, at 22, 22-23.

about the product, or created by the contract or shrink-wrap license.[101] A typical shrink-wrap license contains an express warranty that the software "will perform substantially in accordance with the accompanying written materials for a period of ninety days from the date of receipt."[102] It is evident from the language that this express warranty will not provide users a basis for liability in the vast majority of Y2K Bug cases because the software will not usually malfunction until after the expiration of 90 days.

32. Custom contracts also frequently contain express warranties. Examples of express warranties in custom contracts include: (i) a warranty that the software will be free from defects when delivered;[103] (ii) a warranty that the software will conform to design specifications;[104] and (iii) a statement that the software is provided "as is."[105] The first two warranties could provide a basis for liability for Y2K Bug defects. Software with the Y2K Bug is not free from defects when delivered to the purchaser and product performance will not conform to the design specifications because the Y2K Bug causes the entire program to malfunction in many cases. As with the shrink-wrap license, however, it is common for many express contract warranties to be limited in duration.[106]

33. Express warranties can be created by sales statements and descriptions in advertisements or product brochures.[107] Competition within the computer industry is vigorous and many companies resort to aggressive sales techniques in

[101] *See generally* Herbert J. Hammond, *Limiting and Dealing with Liability in Software Contracts*, COMPUTER LAW., June 1992, at 22, 22-25 (discussing various types of warranties in computer software contracts).

[102] License Agreement, *supra* note 82.

[103] *See* Mecklenburg Furniture Shops, Inc. v. MAI Sys. Corp., 800 F. Supp. 1328, 1331 (W.D.N.C. 1992) ("error free"); Harper Tax Servs., Inc. v. Quick Tax Ltd, 686 F. Supp. 109, 111 (D. Md. 1988) ("free of program coding errors"); Redmac, Inc. v. Computerland, 489 N.E.2d 380, 382 (Ill. App. Ct. 1986) ("free from defects"); W.R. Weaver Co. v. Burroughs Corp., 580 S.W.2d 76, 81 (Tex. Civ. App. 1979) (programs "free from defects in material and workmanship").

[104] *See* Hi Neighbor Enters., Inc. v. Burroughs Corp., 492 F. Supp. 823, 824-25 (N.D. Fla. 1980).

[105] *See* Krider Pharmacy & Gifts v. Medi-Care Data Sys., Inc., 791 F. Supp. 221, 224 (E.D. Wis. 1992).

[106] *See, e.g., Hi Neighbor*, 492 F. Supp. at 825.

[107] *See, e.g.*, Pau v. Yosemite Park & Curry Co., 928 F.2d 880, 887 (9th Cir. 1991) (finding that statement in brochure describing bike trail as safe could be the basis for an express warranty); Neville Constr. Co. v. Cook Paint & Varnish Co., 671 F.2d 1107, 1110 (8th Cir. 1982) (finding that a representation in a brochure created an express warranty); Interco Inc. v. Randustrial Corp., 533 S.W.2d 257, 261 (Mo. Ct. App. 1976) (finding that a description in a supply catalogue can create an express warranty).

their efforts to persuade consumers to buy their products. Unfortunately, in many cases, it is difficult to categorize statements of sales personnel and advertisements as express warranties because the statements may be regarded by a court as "puffing" and not as an affirmation of fact.[108] Sales talk which relates only to the value of the goods or expresses the seller's opinion is considered to be "puffing" and is not binding on the seller as an express warranty.[109] Clearly, however, not all statements made by sales representatives are puffing,[110] and a court should keep in mind that the "natural tendency" of advertising is to "induce a purchase."[111] If sales representatives state that the software is "right" for a consumer's needs, they have established an express warranty that the product will, in fact, meet those needs.[112] Consumers are naïve about most computer products and have to rely on the advice of the sales staff when deciding between products.[113] It is probable that express statements made by the sales staff will form the basis of the bargain for the ordinary consumer. In such situations, it is proper for courts to find that the seller made an express warranty to the buyer.

34. Most contracts, including custom contracts and shrink-wrap licenses, attempt to disclaim all express warranties. Such a disclaimer will not effectively disclaim express warranties, particularly when the warranty is part of the contract.[114] In addition, a disclaimer is only effective to protect the party making the

[108] *See Redmac*, 489 N.E.2d at 383 (noting that a statement, reported in Olin Mathieson Chem. Corp. v. Moushon, 235 N.E.2d 263 (Ill. App. Ct. 1968), that the user would be pleased with results of a system, was "obviously" the opinion of the seller).

[109] *See id.* at 382. The court found that this was not puffing, but an affirmation of fact. *See id.* at 383.

[110] *See id.* at 382 (observing that a statement that a system would be free from defects on delivery was not puffing, but an affirmation of fact).

[111] Eddington v. Dick, 386 N.Y.S.2d 180, 182 (N.Y. City Ct. 1976).

[112] *See, e.g.,* Wilson v. Marquette Elec., Inc., 630 F.2d 575, 580 (8th Cir. 1980) (finding that oral statements about the capabilities of a system created a warranty).

[113] In making a determination about the existence of an express warranty, the court will look at all the circumstances surrounding the transaction, including the knowledge of both parties. *See* Miller v. Lentine, 495 A.2d 1229, 1231 (Me. 1985).

[114] *See* U.C.C. § 2-316(1) (1995) ("Words or conduct relevant to the creation of an express warranty and words or conduct tending to negate or limit warranty shall be construed whenever reasonable as consistent with each other; but . . . negation or limitation is inoperative to the extent that such construction is unreasonable"); *see also* L.S. Heath & Son, Inc. v. AT&T Info. Sys., Inc., 9 F.3d 561, 570 (7th Cir. 1993) (noting that a warranty disclaimer that is inconsistent with an express warranty is inoperative under UCC § 2-316); Consolidated Data Terminals v. Applied Digital Data Sys., Inc., 708 F.2d 385, 391 (9th Cir. 1983) (stating that an express warranty created by product specifications is not overridden by a disclaimer).

disclaimer.[115] For example, in the majority of individual consumer transactions, the consumer purchases the software from a party other than the original software manufacturer. Any disclaimer contained in a shrink-wrap license will only protect the party making the disclaimer, specifically the manufacturer. The software vendor will not be protected by the software manufacturer's disclaimer and will be open to warranty liability unless the vendor also disclaims such liability.[116]

b. UCC § 2-314: Implied Warranty of Merchantability[117]

35. Warranty of merchantability is implied in every contract for the sale of goods made by merchants.[118] To be "merchantable," goods must meet several criteria.[119] Merchantable goods must be "fit for the ordinary purposes for which such goods are used."[120] Computer software with the Y2K Bug, particularly newer software,[121] arguably, is not fit for its ordinary use because a failure to process date functions correctly typically leads to failure of the entire software package.[122] Although product life cycles are short, vendors should expect that computer software will be used by consumers for extended periods of time.[123] In addition, it is foreseeable that users buy a product that they intend to use, and should be able to

[115] *See* Barazzotto v. Intelligent Sys., Inc., 532 N.E.2d 148, 150 (Ohio Ct. App. 1987) (differentiating a manufacturer's disclaimer from a reseller's disclaimer).

[116] *See id.* (finding that the reseller was not protected unless it had made its own disclaimer).

[117] U.C.C. § 2-314 (1995).

[118] *See id.* (stating that "a warranty that the goods shall be merchantable is implied" in the sales contract); *see also* Neilson Bus. Equip. Center, Inc. v. Monteleone, 524 A.2d 1172, 1175 (Del. 1987).

[119] *See* U.C.C. § 2-314(2) (1995).

[120] U.C.C. § 2-314(2)(c).

[121] As many as 95% of businesses within the United States are affected by the Y2K Bug. *See* William S. Galkin, *Negotiating the End of the Millennium*, COMPUTER LAW OBSERVER (Dec. 1996) <http://lawcircle.com/issue21.html>.

[122] *Cf. Neilson*, 524 A.2d at 1175 (finding that the failure of a software product to meet the needs for which the product was purchased is a breach of the implied warranty of merchantability). *But cf.* Step-Saver Data Sys. Inc. v. Wyse Tech., 939 F.2d 91, 107 (3rd Cir. 1991) (finding that the incompatibility of a computer terminal with an operating system did not breach the implied warranty of merchantability where the terminal conformed to industry standards and competitors' terminals had the same compatibility issues).

[123] *See* discussion of the obstacles to upgrading created by cost and product instability *infra* notes 153-157 and accompanying text.

use indefinitely, especially in light of the enormous costs to change from an existing platform to a new one.[124]

36. There is an argument, however, that it was not foreseeable at the time of purchase that legacy software would be used for so many years. This argument disregards the state of the computer industry. Users, namely institutional users, do not and have not upgraded their large systems because the enormous cost of doing so provides a disincentive to change.[125] As a result, manufacturers must expect software to be used for extended lengths of time. Although some users might have chosen to buy software with the Y2K Bug at a lower cost up front if they had to pay for compliance upgrading later, users were never given a choice. Purchasers should not be forced to pay for the date failures, arguably at a price higher than if the software would have been programmed correctly, when the manufacturers have had the opportunity to correct the bugs for many years. Since products with the Y2K Bug will fail in many of their most critical functions, the implied warranty of merchantability is breached.

c. UCC § 2-315: Implied Warranty of Fitness for Particular Purpose[126]

37. The implied warranty of fitness for a particular purpose applies to contracts "[w]here the seller at the time of contracting has reason to know [a] particular purpose for which the goods are required and that the buyer is relying on the seller's skill or judgment to select . . . suitable goods."[127] Most consumers consider software purchases a solution to data management problems. This is particularly true with institutional buyers, such as banks and corporations, that invest substantial amounts of time and resources into the acquisition of computer systems.[128] In the typical scenario, the consumer will look to software sellers to recommend the package with the required features. Sellers that recommend particular software packages are representing to the buyer that the recommended software will, indeed, work for that purpose.[129] From these facts, a court could find a breach of this representation when the software subsequently fails to perform

[124] All computer upgrades or changes are expensive. For instance, BankBoston is spending $50 million to become year 2000 compliant. *See Clock Ticks*, *supra* note 29, at 10.

[125] *See* discussion of the cost of upgrading *infra* notes 153-157 and accompanying text.

[126] U.C.C. § 2-315 (1995).

[127] *Id.*

[128] *See Clock Ticks, supra* note 29, at 10.

[129] *See* Sperry Rand Corp. v. Industrial Supply Corp., 337 F.2d 363, 371 (5th Cir. 1964) (finding that a salesperson's recommendation of what a user needed created an implied warranty of fitness for that particular purpose).

correctly because of the Y2K Bug.[130] The main obstacle in relying on this warranty is that a determination of whether the warranty exists is highly dependent on the circumstances surrounding the transaction.[131]

3. Effect of Proposed UCC Article 2B[132]

38. For many years, commentators have debated the applicability of UCC Article 2 to computer software transactions.[133] Additionally, as discussed earlier, courts have not been consistent in their application of Article 2 to computer software.[134] Much of the difficulty in applying Article 2 to software transactions stems from the fact that software transactions are not easily compared to normal sales transactions. In response, a new article of the UCC, Article 2B ("Draft Article" or "Article 2B") has been proposed.[135] Article 2B would focus exclusively on information technology licenses.[136] While the Draft Article focuses on licenses,[137] the normal method of transferring rights to use software, it would also apply to sales of computer software.[138]

[130] *See* Wilson v. Marquette Elec., Inc., 630 F.2d 575, 579-83 (8th Cir. 1980) (holding that it was not clearly erroneous for the district court to find that an implied warranty was created where the seller knew of the buyer's requirements and purported to sell a system that would meet those requirements); Neilson Bus. Center. Corp., Inc. v. Monteleone, 524 A.2d 1172, 1175-76 (Del. 1987) (finding an implied warranty was created where the seller knew of specific data processing needs and attempted to customize software to meet those needs).

[131] *See* U.C.C. § 2-315 cmt. 1.

[132] The analysis in this Note was based on the September 1997 draft.

[133] *See generally* Horovitz, *supra* note 69; Holly Keesling Towle, *Licensing and the Uniform Commercial Code*, *in* ADVANCED SEMINAR ON DRAFTING LICENSING AGREEMENTS, at 147, 149-158 (PLI Patents, Copyrights, Trademarks, & Literary Property Course Handbook Series No. G4-3994, 1997) (stating that Article 2 is a poor model for software transactions and describing the changes of proposed Article 2B).

[134] *See supra* notes 69-71 and accompanying text.

[135] *See* U.C.C. Draft Article 2B (Sept. 1997) [hereinafter Draft Article].

[136] *See id.* § 2B-103. Specifically, Article 2B would apply to the licensing of information in the copyright industries. *See id.* § 2B-102(a)(22) (defining the term "information" to include works of authorship including literary and musical works). Patent and trademark licensing transactions are specifically excluded. *See id.* § 2B-103(d)(2).

[137] *See id.* § 2B-102(a)(25) (defining license).

[138] *See id.* § 2B-103(c), (d)(4).

39. Under the traditional sales analysis of Article 2, courts have required perfect tender in the performance of a contract.[139] The perfect tender rule, while fitting for sales of goods, is unrealistic when applied to computer software because software will always have some minor bugs. As a result, the drafters have rejected the perfect tender rule in favor of substantial performance.[140] Under a substantial performance test, a minor defect, such as a bug in a program, does not warrant a rejection of the product or cancellation of the contract.

40. The warranty provisions of the Draft Article would also play a role in determining liability.[141] For computer programs, warranties focus specifically on performance.[142] The relevant portions of Section 2B-403(a) state that "[t]o be merchantable, the computer program . . . at minimum must: (1) pass without objection in the trade under the contract description; [and] (2) be fit for the ordinary purposes for which it is distributed."[143] Since the Y2K Bug inhibits performance, it is likely that the Draft Article would be more advantageous to companies seeking to recover for year 2000 defects that cause their systems to stop functioning.

41. The major change proposed in the Draft Article is the recognition and endorsement of shrink-wrap licenses.[144] Draft Article § 2B-208 deals with mass-market licenses by providing that "a party adopts the terms of a mass-market license . . . if the party agrees, including by manifesting assent, to the license before or in connection with the initial performance or use of or access to the information."[145] One of the main problems with shrink-wrap licenses is that consumers do not know of them until after the purchase. Draft Article § 2B-208 overcomes this problem by finding that review, even in connection with the initial use of the software, is sufficient.[146] A court has a limited power to exclude a term which a consumer had no opportunity to review if such term, "creates an obligation or imposes a limitation that . . . the party proposing the form should know would

[139] *See* U.C.C. § 2-601 (1995) (allowing a buyer to reject if the goods "fail in any respect to conform to the contract"). Other sections of the code, however, mitigate the perfect tender rule. *See, e.g.,* U.C.C. § 2-508 (1995) (governing a seller's opportunity to cure non-conforming performance).

[140] *See* Draft Article § 2B-102(a)(43) (defining substantial performance); *id.* § 2B-110 (defining breach).

[141] *See id.* §§ 2B-402, 403 (providing express and implied warranties of merchantability).

[142] *See id.* § 2B-403.

[143] *See id.* § 2B-403(a)(1)-(2).

[144] Few transactions involving the Y2K Bug would, in actuality, be governed by Article 2B because it has not yet become law in any state.

[145] U.C.C. § 2B-208(a).

[146] *See id.* § 2B-208(a).

cause an ordinary reasonable person acquiring this type of information in the general mass market to refuse the license if that party knew that the license contained the particular term."[147] The court is restricted even in this limited power, however, to exclude only terms that are "bizarre or oppressive by industry standards or commercial practices."[148] Under the Draft Article, consumers in jurisdictions that do not enforce shrink-wrap licenses will be in a worse situation than under the current Article 2. In either case, it is debatable, from a policy perspective, whether software companies should be given such broad power to dictate the terms in their licensing agreements.[149]

42. Because Article 2B has yet to be enacted in any state, it will not govern many software transactions or the resulting Y2K Bug litigation. A court may look to the provisions of Article 2B, however, to inform its interpretation of terms as they are used under existing law.[150] If the UCC warranty provisions apply, software users are in a good position to recover based upon either an express or implied warranty theory. Most contracts and shrink-wrap licenses, however, will disclaim or limit

[147] *Id.*

[148] *Id.* § 2B-208(c) The comment to this section notes:

> The refusal term concept holds that unknown terms that are not assented to specifically are not included if they would have led to refusal of the license. At the Annual Meeting, in discussion of Article 2, a motion was made to clarify the basis of exclusion in that Draft and seemed to receive substantial support, but it was withdrawn on the assurance from the Article 2 Committee that the "message" had been received and that, in two years, adjustments would be made. Bracketed subsection (c) contains language from that motion and the *Restatement* comments. The Committee should consider whether the concept should be expressly adopted.

Id. cmt.

[149] One industry group, the American Committee for Interoperable Systems ("ACIS"), is opposed to the adoption of Article 2B because it allows enforcement of shrink-wrap licenses as contracts. *See* Will Rodger, *Code May Deny Software Liability*, INTER@CTIVE WEEK, Oct. 13, 1997, at 72, 72. Among the members of ACIS opposing the code provision are 3Com Inc. and Sun Microsystems, Inc. *See id.*

[150] *Compare* Draft Article § 2B-102(30) (defining "merchant" as a person that deals in information of the kind involved in the transaction, a person that by occupation purports to have knowledge or skill peculiar to the practice or information involved in the transaction, or a person to which knowledge or skill may be attributed by the person's employment of an agent or broker or other intermediary that by its occupation holds itself out as having the knowledge or skill") *with* U.C.C. § 2-104 (defining merchant as "a person who deals in goods of the kind or otherwise by his occupation holds himself out as having knowledge or skill peculiar to the practices or goods involved in the transaction").

these warranties. As a result, software users might turn to non-contract based theories for recovery.

B. Negligence

43. Negligence is a tort that enables recovery for harm inflicted by unreasonable conduct.[151] It is defined as "conduct which falls below the standard established by law for the protection of others against unreasonable risk of harm."[152] To recover damages, a software user must show that it was unreasonable to manufacture and sell software containing the Y2K Bug.[153] Because the Y2K Bug affects the basic functionality of software, there is an argument that it was unreasonable for vendors to develop and sell products that will not function correctly in the next century.

44. There are, however, several problems with the use of negligence as the basis for liability in Y2K Bug suits. First, it is unclear how the customs of the computer industry would affect a determination of reasonableness. The practice of storing and calculating dates by the last two digits has a long tradition within the computer industry.[154] In many ways, this is market driven because consumers continually want faster and more capable machines. For this reason, it might be reasonable for vendors to make programming decisions that optimize speed and efficiency.

45. Vendors also expect that the life cycle for most software programs is short.[155] The industry knows that it is practically impossible to program a bug-free product, and it does not expect consumers to continue to use old versions when new and improved ones have been released.[156] As already noted,[157] there is an underlying

[151] *See* RESTATEMENT (SECOND) OF TORTS § 281 (1965) (listing the elements of a negligence cause of action).

[152] *Id.* § 282 (1965).

[153] *See* Invacare Corp. v. Sperry Corp., 612 F. Supp. 448, 453-54 (N.D. Ohio 1984) (allowing a negligence suit to proceed where it was alleged that the vendor negligently advised the plaintiff to purchase a program when the vendor knew or should have known that the recommended system was inadequate for the needs of the purchaser).

[154] *See* de Jager, *supra* note 3, at 105.

[155] For example, less than two years after Microsoft introduced Microsoft Office 95, they replaced it with Microsoft Office 1997. *See Microsoft Announces the Immediate Availability of Office 97* (visited Feb. 21, 1997) <http://www.microsoft.com/corpinfo/press/1997/Jan97/97avilpr.htm>.

[156] For example, Microsoft began testing Windows 97 even though the majority of businesses had not upgraded to Windows 95. *See* Ed Scannel & Bob Trott, *The Road to Memphis,* INFOWORLD, Mar. 24, 1997, at 1, 14.

fallacy with this argument. Many institutional computer users contract for large and expensive systems,[158] and many of these systems that were implemented in the 1970s are still in use today. Because of the enormous costs involved in migrating to a new system, these institutions have used the same programs for decades. Computer programs are also unstable, making consumers hesitant to upgrade to new versions that have not been market tested.[159] Consumers would rather use an old "reliable" system than take the chance of upgrading.[160] The problem is that the older the system, the more likely that it will have severe Y2K Bug problems.

46. Another factor affecting the reasonableness of the two-digit date design is feasibility. In determining reasonableness, a court is likely to look at the development environment as it existed when the particular programming decision was made.[161] When many legacy systems were developed, there arguably was no alternative to two-digit date functions. As noted earlier, the design decision was driven by cost concerns.[162] It is possible, even probable, that when the systems were originally designed, no company could have absorbed the cost, either in money or time, of a system that stored all four digits of the year. If this is the case, the only practical and reasonable decision was to shorten the date.

47. The final problem with employing negligence as a basis for recovery for defective software is that the losses from the defects will, ordinarily, be economic losses. Under the economic loss doctrine, many courts will refuse to allow a negligence action when there is no physical injury to person or property.[163] The

[157] *See supra* note 126 and accompanying text.

[158] Corporations are hesitant to upgrade existing software because of the costs involved. *See Executives Show Interest in Windows 95 Upgrade*, HOUSTON BUS. J., May 10, 1996, *available in* 1996 WL 7818437 (describing cost as the largest deterrent in corporate upgrades); Scannel & Trott, *supra* note 156, at 1, 14 (noting that users, including large corporations, cannot invest the money to license new software and then pay for the short-term expenses of learning to use it, when product life cycles are so short).

[159] *See NT Economies and 32 Bit Migration*, COMPUTER FIN., Mar. 1, 1996, *available in* 1996 WL 8619412 (stating that corporations find it too risky when making significant upgrades to use the first version of a new operating system).

[160] A study of Fortune 500 companies, conducted by Forrester Research six months after the release of Windows 95, found that only 15% had upgraded to the new operating system with the vast majority of companies continuing to run Windows 3.1. *See* Scannel & Trott, *supra* note 156, at 14.

[161] *See* JOSEPH W. GLANNON, THE LAW OF TORTS 68-69 (1995).

[162] *See* Levy & Hafner, *supra* note 2, at 54; Goldberg, *supra* note 6.

[163] *See* East River S.S. Corp. v. TransAmerica Delavel, Inc., 476 U.S. 858, 871-72 (1986) (recognizing the economic loss rule in admiralty cases based on negligence or strict liability); s*ee also* Horsell Graphic Ind. Ltd. v. Valuation Counselors, Inc., 639 F. Supp. 1117, 1121 (N.D. Ill. 1986)

application of the economic loss doctrine illustrates the fundamental difference between the policy basis of contract law and tort law.[164] "[C]ontract law, which protects expectation interests, provides the proper standard [for recovery] when a qualitative defect is involved, *i.e.,* when a product is unfit for its intended use."[165] On the other hand, tort remedies, including strict liability, protect against exposure to unreasonable risks of injury, both to persons and to property.[166] Nevertheless, a majority of courts apply the economic loss doctrine to prevent recovery when the injury results in only economic losses.[167]

48. Economic losses have been defined as "damages for inadequate value, cost of repair and replacement of the defective product, or consequent loss of profits—without any claim of personal injury or damage to other property,"[168] and "diminution in the value of the product because it is inferior in quality and does not work for the general purposes for which it was manufactured and sold."[169] If the losses from Y2K problems are lost profits due to system crashes, they would clearly be classified as economic losses and be prohibited as tort damages. If a company's hardware is damaged or if data is corrupted, however, there is a strong argument that property has been damaged. Damage to hardware in particular goes beyond the mere failure of the software to operate. Even if a court were to allow a claim based on damage caused to hardware, the recovery allowed is likely to be significantly less than any lost profits due to Y2K Bug-related failures. Because most losses from the Y2K Bug will be economic, it is unlikely that much recovery will be allowed on a negligence theory of liability. Recovery based on computer malpractice or professional negligence is not likely to be limited by the economic loss doctrine and therefore may be a more viable alternative for recovery.

(prohibiting "recovery for pure economic loss in a tort action" but allowing recovery for negligent misrepresentations that constituted professional malpractice).

[164] *See* Moorman Mfg. Co. v. National Tank Co., 435 N.E.2d 443, 447-52 (Ill. 1982) (discussing the rationale for refusing to allow tort liability for economic losses).

[165] *Id.* at 448.

[166] *See id.* at 447.

[167] *See, e.g., id.* at 450-52; s*ee also* NMP Corp. v. Parametric Tech. Corp., 958 F. Supp. 1536, 1546-47 (N.D. Okla. 1997) (economic loss rule bars negligence claims for defective software); Laidlaw Envtl. Servs. (TOC), Inc. v. Honeywell, Inc., 966 F. Supp. 1401, 1414 (D.S.C. 1996) (disallowing a negligence action for a software transaction); Krider Pharmacy & Gifts, Inc. v. Medi-Care Data Sys., Inc., 791 F. Supp. 221, 226 (E.D. Wis. 1992) (stating that the UCC's legislative protections cannot be augmented by tort principles).

[168] Note*, Economic Loss in Products Liability Jurisprudence*, 66 COLUM. L. REV. 917, 918 (1966).

[169] Comment, *Manufacturers' Liability to Remote Purchasers for "Economic Loss" Damages—Tort or Contract?,* 114 U. PA. L. REV. 539, 541 (1966).

C. Computer Malpractice

49. Ordinary consumers rely on professionals, such as lawyers and accountants, to provide them with accurate, reliable advice. Professionals usually have superior knowledge and expertise in their fields, and so it is reasonable for those outside the profession to rely on their judgments. As a result, courts typically hold these professionals to a higher standard of care when rendering professional services.[170] The malpractice or professional negligence cause of action applies to individuals who are members of a profession.

50. Courts have identified several factors that tend to distinguish a profession[171] from a business.[172] Professions typically exhibit the following characteristics:

> extensive formal training and learning, admission to practice by a qualifying licensure, a code of ethics imposing standards qualitatively and extensively beyond those that prevail or are tolerated in the marketplace, a system for discipline of its members for violation of the code of ethics, a duty to subordinate financial reward to social responsibility, and notably, an obligation on its members, even in nonprofessional matters, to conduct themselves as members of a learned, disciplined and honorable occupation.[173]

51. Several courts have addressed the specific issue of computer malpractice.[174] Based on a strict definition of profession, courts have held that the

[170] *See generally* McCullough v. Sullivan, 132 A. 102 (N.J. 1926) (finding law a profession and holding lawyers to a professional standard); Smith v. London Assurance Corp., 96 N.Y.S. 820 (N.Y. App. Div. 1905) (holding accountants to a professional standard); United Dentists, Inc. v. Bryan, 164 S.E. 554 (Va. 1932) (holding dentist to a professional standard).

[171] A profession is defined as "[a] vocation or occupation requiring special, usually advanced, education, knowledge, and skill." BLACK'S LAW DICTIONARY 1210 (6th ed. 1990).

[172] A business is defined as an "[a]ctivity or enterprise for gain, benefit, advantage or livelihood." *Id.* at 198 (6th ed. 1990).

[173] *In re* Estate of Freeman, 311 N.E.2d 480, 483 (N.Y. App. 1974).

[174] *See, e.g.,* Hospital Computer Sys., Inc. v. Staten Island Hosp., 788 F. Supp. 1351, 1361 (D.N.J. 1992) (holding that computer consultants are not professionals); Invacare Corp. v. Sperry Corp., 612 F. Supp. 448, 453-54 (N.D. Ohio 1984) (holding that negligence in a computer business setting does not equal "computer malpractice"); Chatlos Sys., Inc. v. National Cash Register Corp.,

cause of action does not apply to the computer industry because the computer industry is not a "profession" in the traditional sense.[175] It is undisputed that the computer industry does not satisfy the usual requirements of a "profession," but members of the computer industry are, in reality, looked upon as professionals and should be treated as such.[176] Additionally, there are efforts within the computer industry to establish ethical and professional guidelines, one of the central features of traditional professions.[177]

52. The approach adopted by most courts is not, however, the only approach to computer malpractice. The concept of professional negligence is another possible theory under which computer professionals could be held to a higher standard of care.[178] Under the Restatement (Second) of Torts (the "Restatement") section 299A, professional negligence applies to those "who undertake to render services in the practice of a profession or trade" and requires qualifying individuals "to exercise the skill and knowledge normally possessed by members of that profession or trade."[179] By including members of a skilled trade along with members of a profession, the Restatement would allow courts to hold computer vendors to a professional negligence standard without straining to fit the computer industry into the traditional definition of a profession.[180]

479 F. Supp. 738, 740 n.1 (D.N.J. 1979) (refusing to create a "new tort" of computer malpractice without "sound precedential authority"). *See generally* Ronald N. Weikers, "*Computer Malpractice" and Other Legal Problems Posed by Computer "Vaporware*," 33 VILL. L. REV. 835 (1988); Daniel T. Brooks, *Professional Malpractice Liability of Computer*, *in* 10TH ANNUAL COMPUTER LAW INSTITUTE, at 707 (PLI Patents, Copyrights, Trademarks, & Literary Property Course Handbook Series No. G4-3820, 1988).

[175] *See, e.g., Hospital Computer Sys.*, 788 F. Supp. at 1361; *Invacare*, 612 F. Supp. at 453-54; *Chatlos Sys.*, 479 F. Supp. at 740 n.1.

[176] There are many examples of the computer industry acting as a profession. Recently, the Institute of Electrical and Electronics Engineers ("IEEE"), the preeminent computer professional society, proposed standards for defining year 2000 compliance. *See Year 2000 Standard in Works*, COMPUTERWORLD, Sept. 15, 1997, at 8.

[177] *See* IEEE-CS/ACM Joint Task Force on Software Engineering Ethics and Professional Practices, *Software Engineering Code of Ethics* (visited Oct. 27, 1997) <http://www.computer.org/tab/seprof/code.htm> [hereinafter *Code of Ethics*] (reciting the text of the software engineering code of ethics); *see also* ACM, *ACM Code of Ethics and Professional Conduct* (last modified July 30, 1997) <http://www.acm.org/constitution/bylaw17.html> (detailing the Association for Computing Machinery's code of ethics).

[178] *See* Diversified Graphics, Ltd. v. Groves, 868 F.2d 293, 295-96 (8th Cir. 1989) (holding a company hired for its computer expertise to a professional standard of care).

[179] RESTATEMENT (SECOND) OF TORTS § 299A (1965).

[180] At least one court has applied section 299A of the Restatement to hold computer personnel to a professional standard of care. *See Invacare*, 612 F. Supp. at 453-54 (upholding a negligence claim alleging breach of the standard of care to which those in the computer industry are held). The

53. The drafters of section 299A intended it to apply to those who render services in the practice of a skilled profession or trade.[181] Although no explicit definition of skilled[182] trade is provided, the Reporters list several examples of skilled trades, including airline pilots, electricians, carpenters, and plumbers.[183] Computer vendors, like electricians and plumbers, possess specialized knowledge and are considered experts at what they do.[184] Ordinary consumers contact an electrician if they need electrical work performed just as they contact a computer specialist when they need to purchase computer systems. The art of designing, programming, and testing software applications is a skill that is perfected over many years. For this reason, the computer industry should, at a minimum, be treated as a skilled trade.

54. Ironically, courts apply the professional negligence standard to other trades that are arguably less skilled than the computer industry.[185] Given the extreme threat that the Y2K Bug possesses and the integral role of computers in today's society, there are good reasons to hold those in the computer industry to a professional negligence standard. Vendors need incentives to look beyond simple efficiency concerns in the design of their products. Software is of little use if its core functionality is sacrificed. Vendors should not be able to transfer the costs of careless design decisions to the consumer.

55. Under a computer malpractice theory, it may be necessary to distinguish between legacy and newer software. While there are strong arguments today for treating the computer industry as a profession, these arguments are weaker as one goes further back in time. Most legacy software was developed when there were few, if any, standards governing software development other than cost. It would seem, however, that if computer malpractice were recognized, it would be difficult to

court specifically differentiated the section 299A claim from the new tort of computer malpractice, which it declined to recognized. *See id.*

[181] *See* RESTATEMENT (SECOND) OF TORTS § 299A, cmt. b.

[182] Skill is defined as "[p]ractical and familiar knowledge of the principles and processes of an art, science, or trade, combined with the ability to apply them in practice in a proper and approved manner and with readiness and dexterity." BLACK'S LAW DICTIONARY 887 (6th ed. 1990).

[183] *See* RESTATEMENT (SECOND) OF TORTS § 299A cmt. b.

[184] *See Invacare*, 612 F. Supp. at 453 ("If machinists, electricians, carpenters, blacksmiths, and plumbers, are held to the ordinary standard of care in their professions, the Court fails to see why personnel in the computer industry should be held to any lower standard of care.").

[185] *See, e.g.,* Gerard v. Peterson, 448 N.W.2d 699, 703 (Iowa Ct. App. 1989) (real estate agent); Doupnik v. Usher Pest Control Co., 346 N.W.2d 699, 701 (Neb. 1984) (pest control business); Fantini v. Alexander, 410 A.2d 1190, 1192 (N.J. Super. Ct. App. Div. 1980) (karate instructor); Roizen v. Marder's Nurseries, Inc., 615 N.Y.S.2d 235, 236 (N.Y. Sup. Ct. 1994) (landscaping business).

develop a rule that distinguished between legacy and new software. Although original legacy code was written decades ago in many cases, how old would the software actually have to be? How would maintenance of the code performed within the last three to five years affect this determination? The better rule would be to draw no distinction between legacy and other software, particularly given the fact that the Y2K Bug has been well known for many years. Assuming there were no professional standards to guide the original coding, emerging professional standards should require repair of known defects such as the Y2K Bug. While the recognition of computer malpractice will be costly to computer manufacturers, these costs must be borne by someone. If users continue to absorb losses, either directly or through insurance, the computer industry is not likely to advance beyond its current state, where poor quality is the norm. The industry should bear the cost of programming defects, particularly those that rise to the magnitude of the Y2K Bug.

D. Misrepresentation

56. Consumers depend upon the advice of computer salespeople when selecting computer software. To induce consumers to buy their products, salespeople often make both written and oral statements about the functionality of the software and how it will meet the consumers' needs.[186] Because consumers lack knowledge about the technology that underlies software, statements about the technology often become the basis on which the consumers decide to purchase software.[187] Statements that a particular product will "take the user into the next century" implicitly represent that the product being purchased will indeed function in the next century.[188] If the program has the Y2K Bug, however, it will not operate correctly in the next century.

57. Several courts have addressed misrepresentation in the computer sales context.[189] Elements necessary to establish a fraud claim include:

[186] *See, e.g.,* Glovatorium, Inc. v. NCR Corp., 684 F.2d 658, 661 (9th Cir. 1982) (involving a computer system sold to perform a function that it was never designed to perform); Dunn Appraisal Co. v. Honeywell Info. Sys. Inc., 687 F.2d 877, 882-84 (6th Cir. 1982) (concerning a statement that a product was suitable for the customer's intended use).

[187] *See, e.g.,* Sperry Rand Corp. v. Indus. Supply Corp., 337 F.2d 363, 369-70 (5th Cir. 1964) (finding that the buyer relied on the seller's judgment in making a computer software purchase).

[188] The first lawsuit filed involving year 2000 problems is partially based on an allegation of misrepresentation. *See* Wong, *supra* note 37, at 6. Surprisingly, the complaint does not allege misrepresentations of the initial transaction; rather, the complaint alleges that misrepresentations were made about the ability of the defendant to correct the problems. *See* Compl. of Produce Palace, Int'l ¶ 73-74, Produce Palace, Int'l v. Tec-America Corp., No. 97-3330-CK (Mich. Cir. Ct., filed June 12, 1997).

[189] *See* Graphic Sales, Inc. v. Sperry Univac Division, Sperry Corp., 824 F.2d 576, 580 (7th Cir.

> (1) a misrepresentation by defendant of a material existing fact,
> (2) with knowledge that it was false or with reckless disregard as to whether it was true,
> (3) with intent to deceive plaintiff, and
> (4) plaintiff acting upon the misrepresentation in reasonable reliance upon its veracity in a manner which caused proximate injury.[190]

58. In many cases, courts have rejected a plaintiff's claim of misrepresentation.[191] For the Y2K Bug, however, a consumer should have more success establishing the elements of misrepresentation.

59. First, vendors misrepresent a material existing fact when they pitch their products as useful in the twenty-first century. The functionality of the software, both now and in the twenty-first century, is material to the bargain. If a consumer knew that the product was not year 2000 compliant, it is unlikely that the consumer would have gone forward with the purchase.

60. Second, most vendors know whether their product is capable of functioning as represented in the year 2000. Even if the salesperson, who typically does the misrepresenting, does not know of the problem, that person could still be in reckless disregard as to whether the fact is true. Additionally, most marketing brochures and advertisements are approved by those in a position to know if the product is in fact year 2000 compliant.

61. Finally, it is necessary that consumers rely to their detriment upon the misrepresentation. Since most consumers do not have the knowledge to evaluate independently whether a particular software program will meet their needs, they rely on the advice of those in the computer industry to aid them in making this judgment. It is reasonable for them to rely on these representations because they have no other mechanism for making software purchasing decisions. Of course, if the

1987) (finding that the facts did not support a finding of misrepresentation); Accusystems, Inc. v. Honeywell Info. Sys., Inc., 580 F. Supp. 474, 480-81 (S.D.N.Y. 1984) (stating that an element of negligent misrepresentation is a "special relationship of trust or confidence," and that the buyer-seller relationship in this case did not entail a sufficient degree of trust); Management Assistance, Inc. v. Computer Dimensions, Inc., 546 F. Supp. 666, 670-75 (N.D. Ga. 1982) (finding no fraud in the inducement because the buyer was not prevented from reading the contract prior to purchase and the contract clearly limited express warranties); Chatlos Sys., Inc. v. National Cash Register Corp., 479 F. Supp. 738, 748-49 (D.N.J. 1979) (finding that statements were "overly optimistic, not fraudulent").

[190] *Management Assistance, Inc.*, 546 F. Supp. at 671.

[191] *See id.* at 675; *Graphic Sales, Inc.*, 824 F.2d at 580-81. *But see Accusystems, Inc.,* 580 F. Supp. at 481-82; *Glovatorium,* 684 F.2d at 660; *Dunn Appraisal*, 687 F.2d at 882-84.

consumer is knowledgeable about the information technology industry, the reasonableness of his or her reliance on any statement will be examined against the heightened degree of knowledge the consumer possesses. Thus, of the fault-based tort theories, misrepresentation is the theory most likely to be successful in potential litigation.[192]

E. Strict Liability

62. Strict or products liability is a tort that, unlike negligence, does not require fault as a predicate for finding liability.[193] In the typical products liability case, a seller or manufacturer is held liable for product defects or hazardous products that threaten the personal safety of consumers.[194] Large manufacturers, who are in the best position to discover and fix product hazards, should shoulder the monetary burden for the harms that their products cause. The difficulty in applying strict liability to computer bugs is that computer software with the Y2K Bug, while certainly defective, is not inherently dangerous. Nor will any system failures, in most cases, cause consumers physical harm.

63. Nonetheless, there are arguments that can be made for the application of strict liability to software. Software vendors are in the best position to prevent defects in their products and are solely responsible for the products. Software users ordinarily do not posses the requisite skills or knowledge necessary to change software products. There is a concern that the application of strict liability will hinder development, but this does not apply to the Y2K Bug because although most bugs are, by definition, unknown to the vendor at the time of distribution, this is not the case with the Y2K Bug.

64. Despite these arguments, if strict liability is ever applied to computer software defects, it should be limited to those cases where physical harm to a consumer has resulted. There are cases where computer software bugs have caused physical injuries and even death to individuals.[195] Examples of such defects include a computer malfunction that triggered a fatal administration of radiation therapy or computer errors that interrupted the proper functioning of navigational information

[192] The economic loss doctrine is a potential bar to innocent misrepresentation claims. *See* Moorman Mfg. Co. v. National Tank Co., 435 N.E.2d 443, 453 (Ill. 1982). There would not, however, be a bar to claims based on negligent or intentional misrepresentations. *See id.* at 452.

[193] *See* RESTATEMENT (SECOND) OF TORTS § 402A (1965).

[194] *See id.* §§ 402A, 402B.

[195] A software defect was blamed for the administration of incorrect doses of radiation to cancer patients resulting in six deaths or injuries. *See* Williamson, *supra* note 74, at 81.

to airline pilots.[196] For policy reasons, it is unlikely that this tort would or should be extended to situations arising from the Y2K Bug.[197]

IV. CONCLUSIONS AND RECOMMENDATIONS

65. Contract theories of recovery are the obvious starting points for any Y2K Bug litigation. Breach of contract based on breach of contractual warranty can be employed when the software was designed under contract for a particular user. Such contracts are likely to have a general warranty clause that might provide a basis for relief. Most custom contracts, however, limit liability[198] and most software is purchased through the mass market where shrink-wrap licenses are frequently used.[199] The success of any warranty claim will depend on the language of the particular warranty and the specific facts and circumstances surrounding the transaction.

66. The UCC and its provisions relating to warranties are bound to be the focus of much Y2K Bug litigation. The trend in the case law is to treat computer software as a "good" and apply the provisions of Article 2.[200] Additionally, if Article 2B becomes effective, the UCC will govern software transactions directly.

67. When software products are sold, there is a general representation, unless disclaimed, that the product will function correctly. Misrepresentation becomes an issue with the Y2K Bug because many software products today are sold under the marketing gloss that the product is on the "cutting edge" and will bring the user into the twenty-first century. Any software product with the Y2K Bug that has been marketed in such a way is likely to be subject to misrepresentation claims. Vendors know the operation of date functions and they know that the two digit calculation will fail in the year 2000. As a result, they are fraudulently misrepresenting the capabilities of the software. By representing that the product will take the user into the twenty-first century, the vendor is representing that the product will work in the twenty-first century. This is clearly not the case with the

[196] For an account of instances where computer defects have had tragic consequences, see IVARS PETERSON, FATAL DEFECT: CHASING KILLER COMPUTER BUGS (1995).

[197] The economic loss doctrine also applies to claims based on strict liability. *See* Moorman Mfg. Co. v. National Tank Co., 435 N.E.2d 443, 450-51 (Ill. 1982).

[198] For examples of contracts limiting warranties and liability, see generally Laidlaw Envtl. Servs. (TOC), Inc. v. Honeywell, Inc. 966 F. Supp. 1401 (D.S.C. 1996); Hi Neighbor Enters., Inc. v. Burroughs Corp., 492 F. Supp. 823 (N.D. Fla. 1980).

[199] For a discussion of shrink-wrap licenses and their enforceability, see *supra* notes 77-97 and accompanying text.

[200] For a discussion of the applicability of Article 2 to software transactions, see *supra* notes 69-70 and accompanying text.

Y2K Bug. The misrepresentation doctrine is also attractive because it is available even if the shrink-wrap license is unenforceable.[201]

68. Negligence is another basis for recovery. Under this theory, it can be argued that the vendor was negligent in the design and development of software with the Y2K Bug. It is foreseeable that software would be used after the year 2000; thus, the software development firms should have designed and sold software capable of such uses. By not fixing the Y2K Bug problem, the vendor did not act reasonably and therefore, can be liable. As indicated earlier, the problem with a negligence cause of action is that the losses from the Y2K Bug will most likely be economic, not physical. Courts that employ the economic loss doctrine will not allow the cause of action.[202]

69. Strict liability is unlikely to be used as a basis for recovery in Y2K Bug suits. While, in some situations, information has been considered a product, and is subject to strict liability in certain instances, there is no precedent for applying strict liability to software and little support for its application to similar information sources, such as books. In particular, it will be difficult to recover because no physical harm results from the product defect.

70. Given the novelty of the Y2K Bug, there is little, if any, case law on point. There has been much commentary and speculation in legal periodicals, however.[203] Most of the analysis applies the elements of traditional doctrines to software. It is clear that the UCC and misrepresentation are available for suits stemming from the Y2K Bug. Although courts have been reluctant to do so, the tort of computer malpractice or professional negligence should be recognized and applied to the computer industry. Professional malpractice, or computer malpractice, is an attractive alternative to negligence because a higher standard of care is applied.

71. Like most industries, the computer industry has evolved since its inception forty years ago. It is time for the computer industry to be recognized as a profession and to hold its members accountable as such. Both inside and outside the computer industry, its members are considered professionals. Software developers and other computer professionals must acquire specialized knowledge and skills before they can be productive in the industry. Consumers, particularly businesses, who lack such expertise, employ the services of computer professionals to design, develop, and implement a variety of computer systems and software. Consumers, for the most part, cannot provide these services for themselves and must turn to those in the field. As computers become even more integral to the

[201] *See, e.g.,* Financial Timing Pub., Inc. v. Compugraphic Corp., 893 F.2d 936, 943-44 (8th Cir. 1990) (stating that misrepresentation may trump normal contract defenses).

[202] *See supra* notes 159-65.

[203] For some examples see Clayton et. al, *supra* note 61; Cooney, *supra* note 61; Gerber, *supra* note 61; Kahn, *supra* note 61.

functioning of society, it is important for the industry to produce quality products or accept responsibility and liability for product defects. Potential liability for software defects under a malpractice or professional negligence standard might be the incentive that the industry needs to make the commitment to producing quality products. Even without the recognition of a malpractice claim, courts could use the professional negligence standard of the Restatement to hold computer professionals to a higher standard of care. The courts traditionally refused to recognize the computer industry as a profession because software developers and programmers were not licensed professionals and there were no established educational standards or regulations in the field. The computer field has evolved, however. There are several areas where individuals can become "certified engineers" in a specific product.[204] Major industry organizations are moving toward the adoption of ethical standards governing both the quality[205] of software products and their effects on the public at large.[206] Now that the computer industry is holding itself out as a profession, it should be treated as a profession. Courts, in appropriate circumstances, should no longer hesitate to hold the computer industry to the higher standards required of professionals.

[204] *See, e.g.,* Steve Alexander, *Certification Begins to Take Hold*, INFOWORLD, Mar. 2, 1998, at 103; *Draft Software Engineering Accreditation Criteria*, COMPUTER, Apr. 1998, at 73. In addition, organizations such as the Information Technology Institute advertise courses to become a "Microsoft Certified Systems Engineer." *See, e.g.,* Information Technology Institute, *Microsoft Certified Systems Engineer Program*, INFOWORLD, Mar. 2, 1998, at 104.

[205] *See Code of Ethics*, *supra* note 177, princ. 1.

[206] *See id.*, princ. 2.