



Public Affairs, (510) 642-3734

NEWS RELEASE, 8/2/99

---

## UC Berkeley computer scientists attack Y2K bug with new program to find millennium glitches in C applications

**By Robert Sanders, Public Affairs**

- BERKELEY -- With Y2K less than six months away and companies scrambling to fix end-of-the-millennium computer glitches, a team of computer scientists at the University of California, Berkeley, is offering a bit of help - free software that can detect Y2K bugs in many non-business applications, such as computer operating systems.

The majority of programs with Y2K problems are business applications written in a 40-year-old language called COBOL. Sophisticated commercial products are available that help programmers find and fix Y2K bugs in COBOL applications.

However, there is much less industry support for fixing Y2K problems in C programs, a 1970s-vintage language in which the ubiquitous Windows and Unix operating systems are written, as well as many other critical applications. Carillon, the program the researchers developed, applies principles of formal logic to detect Y2K problems in C programs.

"Most of the tools available for finding Y2K problems are really just making educated guesses about where bugs might be in a program," said [Alex Aiken](#), associate professor of computer science at UC Berkeley. "In the past two years, more reliable tools based on mathematical models of program behavior have been introduced for COBOL, but the C tools have lagged behind."

"A lot of critical code is written in C," he added. "C is the dominant programming language for the Windows and Unix environments, which means that C programs are running nearly all PCs and Workstations worldwide. C also has a big presence in embedded computing, the programs that control everything from fax machines to nuclear power plants, and in networking, including e-mail, Web servers, and the infrastructure of the Internet. Y2K problems in C programs potentially affect a lot of people."

Of course, not all programs written in C and COBOL have Y2K problems, but without checking, it's impossible to know until midnight on December 31. Since it took post-doctoral fellow [Martin Elsmann](#) a mere month to create the Y2K debugging software - named Carillon after UC Berkeley's signature campanile or bell tower - he, Aiken, and graduate student [Jeff Foster](#) decided to distribute it gratis, at the [Carillon Web site](#).

"Time is short for Y2K, so we decided the best way to help people was to give it away," Aiken said. "We hope people will find Carillon useful. There are other date-related problems coming after the year 2000, so Carillon may be around for a while."

The Y2K, or millennium, bug is a problem caused by how computer applications, like word processing or spreadsheet programs, represent the year in a date. Instead of using four digits, such as 1999, many programs represent the year as simply a two-digit number - 99. This leads to problems in the year 2000, which these programs can't distinguish from the year 1900. For the past few years businesses and governments have been either scrapping their outdated programs or spending money for programmers to pore over the code to find places where dates are used and convert them from two to four digits.

"These two-digit dates are sprinkled like sand all over the program," Aiken said. "Sorting out which bits of data are dates in a million-line program is a very time consuming and error-prone task to do by hand, but it's something that a computer should be able to do quite well."

Elsman tested Carillon on a very widely used "version control program" for Unix computers. Called Concurrent Version System (CVS), the free, open-source program keeps track of different versions of documents and the time when changes were made, so that people can backtrack and easily find earlier versions. Many software companies use CVS to manage the multiple versions of their products.

"If you have a date problem with CVS, then you're in trouble," Elsman said. "It could take you back a hundred years."

The most recent version of CVS is believed to be Y2K compliant. To test Carillon's ability to find Y2K errors, Elsman tried Carillon on both the latest release of CVS and on an old version of CVS known to have Y2K bugs. After Elsman told Carillon what to look for, it took only a few seconds for Carillon to isolate the date bugs in the old version of the 57,000-line program. A Carillon scan of the most recent version of CVS confirmed that the program is now Y2K compliant.

"Having an automatic assistant that reasons about program behavior is much better than doing Y2K conversions by hand," Aiken said. "Not only is it faster, but you also have much higher confidence when you are done that the program really does use dates correctly. A tool like Carillon takes all of the guesswork and most of the labor out of finding and fixing Y2K problems."

Aiken believes that the Y2K problem for C has not received much attention because of commercial pressures.

"COBOL is 'the' programming language for business and database applications," Aiken said. "There are more lines of code written in COBOL than in any other programming language, and many COBOL programs manipulate dates, so it makes sense that the Y2K companies would focus on COBOL. But a lot of the world relies on C programs, too."

Besides doing a good turn, Aiken, Elsman and Foster had scientific reasons for developing Carillon. They saw the project as a good way to evaluate a larger piece of software they created to help in designing debugging tools such as Carillon. Called the Berkeley ANalysis Engine ([BANE](#)), it consists of software components that can be assembled in different ways to analyze complex programs for any number of bugs, from the Y2K problem to security holes and other programming flaws.

With the help of this general-purpose tool kit, Elsman created Carillon in one month. Commercial programs to detect Y2K bugs have taken man-years and significant development teams to produce.

"This was an acid test for our tool kit, a real-world problem, and we are pleased with the results," Foster said.

The three researchers work in a field called software analysis, a critical area in a time when most software is so complex no one person can understand how it works, let alone predict what will happen in all circumstances.

"We are taking a different approach than is found in most of the industry today," Aiken said. "We use mathematical techniques to prove things about a program, such as that some kinds of bugs can't possibly occur."

"Rather than building a custom tool for each new problem we want to attack, we have built a tool kit for constructing program analysis tools. We build and experiment with tools like Carillon in a matter of a few weeks or months, rather than in the months or years it would have taken us before we had BANE."

Building Carillon was the first problem the researchers tackled with the full BANE system in place, Aiken said, and it is a good demonstration of the usefulness of the technology.

###

---

This server has been established by [the University of California at Berkeley](#) Public Information Office. Copyright for all items on this server held by The Regents of the University of California. Thanks for your interest in UC Berkeley.

---

[More Press Releases](#) | [More Campus News and Events](#) | [UC Berkeley Home Page](#)

*Send comments to: [comments@pa.urel.berkeley.edu](mailto:comments@pa.urel.berkeley.edu)*