Technical Opportunities to Help with the Year 2000 Problem

Final Report

1997

ARPA Order: E496
ONR Contract: N00014-97-1-0114
Period: November 1, 1996 to September 30, 1997
Total Support: $125,000

Thomas W. Reps, Principal Investigator
Computer Sciences Department
University of Wisconsin
1210 West Dayton Street
Madison, WI 53706

1. Summary of the Completed Project

The "Year 2000 Problem" (Y2K problem) concerns how to avoid the possible breakdown of computer systems due to the use (in both code and data) of only two digits to represent the year in dates. The purpose of this project was to to plan a project aimed at reducing the impact of the Y2K problem on the Department of Defense. Some of the technical issues that I considered included:

- How to conduct experiments that involve setting dates ahead to determine impact.
- How to locate places in a piece of code where dates are used.
- How to perform impact analysis (either statically or dynamically) to determine what other parts of the code are affected by date manipulations.
- Database reformatting and conversion.
- Testing.
- Dealing with multi-lingual systems.
- Dealing with binary-code systems (where the system is written in assembly code, the source code has been lost, or the source code was never delivered).
- "Sand-boxing" or other techniques for isolating the effects of bad/old date formats.

I also investigated problems, opportunities, special risks, possibilities for high-payoff, *etc.* that were specific to the DoD context.

New Techniques for the Y2K Problem

DARPA also asked me to investigate whether there were "any techniques in the research community—ideas not necessarily originally developed with the Y2K problem in mind—that could be applied to the Y2K problem and have impact beyond present commercial Y2K products and services." The most exciting of the ideas that turned up concerns a method for using path profiling as a heuristic to locate some of the sites in a program where there are problematic date manipulations. It works as follows:

> In path profiling, a program is instrumented so that the number of times each different loop-free path executes is accumulated during an execution run. With such an instrumented program, each run (or set of runs) of the program generates a path spectrum for the execution—a distribution of the paths that were executed. Path spectra can be used to identify paths in a program that are good candidates for being date-dependent computations by finding differences between path spectra from execution runs on pre-year-2000 data and post-year-2000 data. By choosing input datasets to hold all factors constant except the way dates are used in the program, any differences in the spectra obtained from different execution runs can be attributed to date-dependent computations in the program. Differences in the spectra reveal paths along which the program performed a new sort of computation during the post-year-2000 run, as well as paths—and hence computations—that were no longer executed during the post-year-2000 run.

With some further analysis of the spectra, for each such path that shows up in the spectral difference, it is possible to identify the shortest prefix that distinguishes it from all of the paths in the other path set (see publications [2] and [1]).

For the Y2K problem, the path-spectrum comparison technique may provide help with two aspect of the problem:

- Determining the sites at which date-manipulation code occurs.
- Post-renovation testing.

Of course, the path-spectrum comparison technique is not guaranteed to uncover all sites of date manipulations. No technique can do this; all one can hope for are good heuristics. However,

because path-spectrum comparison involves a different principle from the principles that lie behind the heuristics used in commercial Y2K tools, it should be a good complement to current techniques. Furthermore, the path-spectrum comparison technique is actually applicable to a much wider range of software-maintenance problems than just the Y2K problem; it offers new perspectives on program testing, on the task of creating test data, and on what tools can be created to support program testing.

A prototype tool for gathering and comparing path spectra, called DynaDiff, was during the project built at the University of Wisconsin. (DynaDiff works on programs that run under Solaris on Sun SPARCstations.) The idea was also written up and reported in publications [3], [2], and [1].

The Proposed DARPA Y2K Problem Plan of Action

I gave an oral presentation of my recommendations about the course of action that DARPA should follow to DARPA Director Larry Lynn, ITO Director Howard Frank, and ITO Program Manager John Salasin on November 15, 1996. I subsequently put my recommendations in writing and sent the written plan to DARPA on November 22, 1996 (see publication [4]). The cost of the plan was estimated to be $14M, largely in FY97 ($8–9M in FY97 and the balance in FY98).

From the information that was reported back to me, it sounded like Lynn essentially approved the plan, modulo input from the Pentagon as to what DARPA's role should be. ITO Assistant Director Howard Shrobe used the material from the plan I had prepared to brief Anita Jones (Director of Defense Research and Engineering) and subsequently Paul Kaminsky (Undersecretary for Acquisition and Technology) and Emmett Paige, Jr. (Assistant Secretary of Defense for C3I). Unfortunately, after a long period in limbo (November 1996 – March 1997), Paige and Jones decided not to fund the plan. In an e-mail message to me, Anita Jones explained her reasons as follows:

> I see that there are some interesting technical challenges.
>
> But, if I put this set of problems up against what I am funding and ask what to cut out in order to fund this, I cannot make the rationale for cutting:
>
> biological agent defeat
> defensive information warfare
> fit-in-your-palm flying sensor vehicles
> telemedicine
> counter sniper weapons
> accurate navigation in urban sites
> Next Generation Internet
> hyperspectral sensors
> .....
> and on and on.

Publications [2] and [4] are included with this report.

2. Publications Written Under ARPA Order E496 (ONR Contract N00014-97-1-0114)

The following papers and other documents were written by the Principal Investigator during the period of ARPA Order E496 (ONR Contract N00014-97-1-0114). Papers [1] and [2] are available over the World Wide Web at URL http://www.cs.wisc.edu/~reps/.

Invited Papers

[1] Reps, T., The use of program profiling in software testing. In *Proc. of Informatik '97* (Aachen, Germany, Sept. 24-27, 1997), M. Jarke, K. Pasedach, and K. Pohl (eds.), Springer-Verlag, Berlin, Ger., 1997, pp. 4-16.

Conference Publications

[2] Reps, T., Ball, T., Das, M., and Larus, J., The use of program profiling for software maintenance with applications to the Year 2000 Problem. In *Proceedings of ESEC/FSE '97: Sixth European Software Engineering Conference and Fifth ACM SIGSOFT Symposium on the Foundations of Software Engineering*, (Zurich, Switzerland, Sept. 22-25, 1997), *Lecture Notes in Computer Science*, Vol. 1301, M. Jazayeri and H. Schauer (eds.), Springer-Verlag, New York, NY, 1997, pp. 432-449.

Patents

[3] Reps, T., Method of troubleshooting data-dependent anomalies in computer programs. U.S. Patent filed January 8, 1997.

Other Publications and Reports

[4] Reps, T., DARPA Year 2000 Plan of Action. Unpublished report, November 22, 1996.