Uplink

THE FORUM FOR COMPUTER-ASSISTED REPORTING

September 1992
Volume 3, Number 8

the *seductive* technology

A story of failure and four myths that made it possible

BY RONALD CAMPBELL
The Orange County Register

At 8:20 p.m. on May 9, a sleeping 3-year-old boy was whipped from his mother's lap by an exploding tire tread and dragged to his death beneath the wheels of a bus.

The bizarre death of Ramon Prado provided an occasion – an excuse, really – for The Orange County Register to publish results of its seven-month probe of bus safety in California.

The story showed that the California Highway Patrol routinely fails to meet its legal mandate to inspect buses. But as the story also noted, California's buses have racked up an enviable safety record. And even if CHP inspectors had found and inspected the bus on which Ramon Prado later died, they could have done nothing about a floor so thin that a tire tread could breach it; the floor met federal standards.

So much for seven months work.

The story, and the endless research that went into it, illustrate what I call seduction by technology. Entranced by the siren song of high tech, I pursued a story long after my instincts told me it wasn't there.

Most articles in Uplink tell of triumphs. This is a story of failure and the four journalistic myths that made it possible.

It began in July 1991, with a bus crash near Palm Springs that took the lives of seven people. The Register immediately began casting about for a way to investigate bus safety. We discovered the CHP's Management Information System of Terminal Evaluation Records, aka MISTER.

It sounded perfect: computerized records on 90,000 bus and truck companies, including every traffic ticket, every accident and every CHP inspection of bus or truck terminals. CHP's reaction made the database seem even better: They demanded $2,800. Over the next three months we argued the price down to $81.18.

Enter Myth No. 1: If a bureaucrat hinders or refuses access to public documents, those records must be explosive. That might have been true, sometimes, in the days when governments kept its records in manila folders. Today, when agencies routinely keep gigabyte-sized databases, it's absurd to presume that custodians know what they're hiding.

Under the spell of Myth No. 1, I was easy prey for Myth No. 2: Gather enough facts and you're bound to find a great story. All reporters play hunches, looking under rocks for something odd. With computers comes the temptation to look under every grain of sand on the beach.

That is what I did with MISTER. Had I faced a similar quantity of paper records, I instantly would have turned away for a reality check. I would have called some experts, read some reports and chosen one or two angles to explore. Instead I plunged into the database with all the direction of a near-sighted man on a foggy night.

"How many people die on buses?" I asked the database. Very few. "How often do buses crash?" Not often. "What does 'often' mean?" Good question, since even CHP doubts the mileage numbers in MISTER.

One day I extracted a list of multiple-injury bus accidents. The heaviest toll by far, more than 50 injuries, was for an accident a year earlier within a few miles of the Register. I could not find a word about that accident in the paper. A local CHP flack looked it up for me: The bus was ferrying prisoners

the *seductive* technology

between court and jail; most of the "injured" were inmates complaining of whiplash.

Myths No. 3 and 4 made an already bad situation nightmarish. I suspect both myths are common among computer-assisted reporters.

Myth No. 3: Databases are the perfect reflection of the real world. Think about that one for a second. Intellectually, of course, no one believes it. But emotionally we tend to invest databases with a God-like aura of omniscience. They seem so complete.

In fact, however, databases merely record what their keepers meant them to record, to the degree their keepers can get the information.

It's easy to lose sight of those limitations. MISTER listed hundreds of companies that no longer existed. It omitted bus and truck accidents on local roads, unless police went to the trouble of informing the CHP. MISTER's record of hazardous spills abruptly stopped in 1990 when CHP "temporarily" stopped entering the data.

Before I ever used a mainframe, I was a devotee of Myth No. 4: Mainframes are big, fast and unstoppable – the Incredible Hulk of the Information Age.

The myth is correct on one point. Mainframes are indeed big. So was Tyrannosaurus Rex.

I thought I needed a mainframe to handle MISTER; at 160 megabytes, it would have stretched the capacity of my PC and overwhelmed XDB, the database program I use. It would have taken weeks to get a few simple answers, I thought.

So I used a mainframe instead. And it did not take weeks to get simple answers; it took months.

We lumbered along, the Register's information services department and I, for three months before I framed my first query to the MISTER database. It took time to install the hardware linking my PC with the mainframe, time to teach me how to use the link, time to load MISTER and then to load it again when the first layout didn't work.

Then the real problems began.

Contrary to its legend, I found the mainframe balky. I would ask what I thought was a simple question and then wait and wait – until an irritated computer tech would call, asking why I was consuming 90 percent of the central processing unit's capacity.

I would have to ask four or five questions on the mainframe where one would have sufficed on the PC. And after other users began to complain in droves about that crazy new user monopolizing the computer, the high priests of the mainframe restricted my access.

As the weeks dragged into months, I began to dread afternoons digging into the database. Few findings excited me, and never for long; if I found an intriguing pattern in the river of printouts, I usually found a flaw an hour or a day later.

Ramon Prado rescued me from all this. Months before his death I had noticed that the CHP was failing its mandate to inspect every bus terminal. Absent a healthy dose of mayhem, this was just one more tiny instance of decaying state service. Prado's death, which no inspector could have prevented, make bus inspections newsworthy. In retrospect, my failure with MISTER offers a couple of easy lessons: First, never use a computer you don't control; the control issue alone makes a newsroom PC superior to a mainframe. Second, before you buy a database, before you spend months probing its secrets, find out its weaknesses.

But the major lesson is one that I am not yet sure I have mastered: A database is not a story; it just might lead to a story, but it is not a story. Lose sight of that lesson, and you too can be seduced by technology.

Had I faced a similar quantity of paper records, I instantly would have turned away for a reality check.. . . Instead I plunged into the database with all the direction of a nearsighted man on a foggy night.

Problem solving at U.S. News and World Report

By John Bare
Ph.D. candidate at the University of North Carolina-Chapel Hill

U.S. News and World Report's summer plunge into computer-assisted reporting illustrated a multi-method approach to problems encountered in working with the Food and Drug Administration's Medical Device Reporting system.

As an intern hired to help develop computer-assisted reporting projects at U.S. News, I worked with section editors, senior writers, associate editors and administrative staff. The magazine's library and its corporate data services division devoted substantial resources to computer-assisted reporting projects.

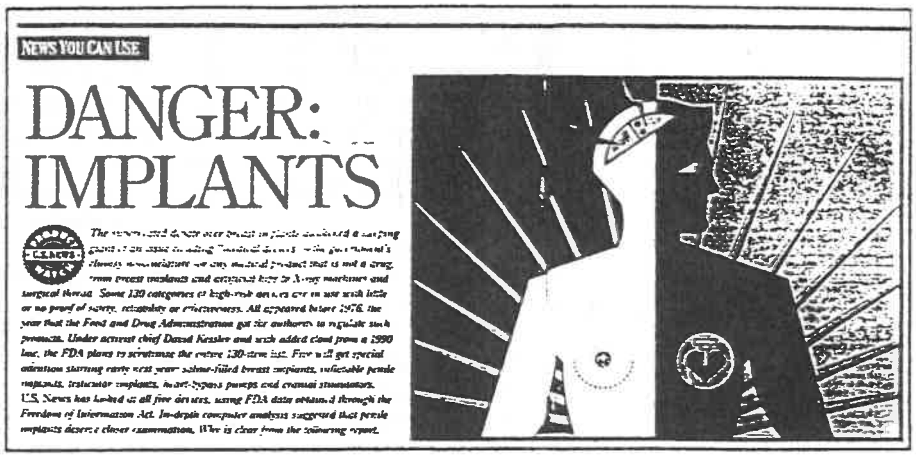
In addition to the magazine's standard word-processing system, ATEX, I used Oracle version 6.0 with SQL Forms, SPSS/PC+ version 4.01, WordPerfect version 5.1 and Lotus 1-2-3 release 3. An AST 386SX/20 connected to a VAX mainframe by an ethernet network was set aside for computer-assisted projects.

The result was "Danger: Implants" (Aug. 24, 1992), a six-page story in the News You Can Use section. Using the MDR computer tapes, U.S. News examined five medical devices set to undergo regulatory scrutiny next year. U.S. News devoted the most attention to penile implants, combining traditional reporting techniques with more scientific methods.

The magazine's corporate data services division loaded the MDR tapes onto the VAX and created a database using Oracle. From my personal computer, I was able to log on to the VAX in just a few seconds. Using Oracle, I could retrieve selected records, sort by fields and produce reports that could be imported into other software packages for additional analysis.

Soon after obtaining the FDA tapes, however, we discovered a major obstacle. When records were sorted by accession number (the MDR equivalent of identification numbers) and product code, the report revealed several problems, including thousands of duplicate records. An FDA programming error had rendered the tapes virtually useless. More meetings with the FDA were scheduled to resolve the problem.

The one thing we learned from the tapes is that key information contained in a text field cannot be sorted. The MDR's massive "event description" field consists of a text paragraph composed by data entry personnel that describes how and why the medical problem occurred.



The one thing we learned from the tapes is that key information contained in a text field cannot be sorted.

With penile implant reports, for instance, the patient's age (if listed at all) is in the event description field, as is the date of the implant operation and the details of the medical device problem. Because this information is buried in a text field, we could not instruct a database program to sort the penile implant records by patient age, implant date or problem type.

Even worse, the form in which the MDR information is archived is inconsistent. The event description field may say that the penile implant patient is 42 years old, or it may say that he was born 8/12/50. One entry might describe the problem as a "leak." Another might say "lost fluid." Still another might say "leakage from reservoir." An implant might protrude, extrude, erode or break the skin. This haphazard style of reporting device problems made it difficult to devise a surefire method of dumping the event description paragraphs into a text analysis or word processing system for more detailed analysis.

So just as social scientists use multiple methods to tackle tough problems, we added another prong to the computer-assisted reporting task. While waiting for a new set of accurate MDR tapes, we completed an old fashioned, hand-coded content analysis of penile implant problem reports, using a sample of 1,196 implant records drawn from microfiche copies of the MDR tapes. We recorded product names, manufacturer names, the report description (death, serious injury or malfunction), report date, date of implant and patient age. In addition, we established nine categories of problems and coded the reports accordingly.

Data from the code sheets were keypunched into Lotus and then imported into an SPSS system file for statistical analysis. With the push of a button, we could find out things such as 8 percent of the penile implants involved infections and the average age of patients with penile implant problems is 56.

By subtracting the implant date from the MDR report date (SPSS "compute" command), we created a variable of the number of years between the implant operation and the problem report. The distribution ranged from zero years (implant and problem in same year) to 16 years but was skewed heavily, with about 62 percent of the problems occurring within three years of the implant date.

By this time, new MDR tapes arrived, and integrity checks indicated that there were no problems. The tapes contained approximately 176,000 medical device problem reports from 1984 through mid-June 1992. Of these, 8,064 records were singled out as penile implant problem reports.

In our continuing effort to cull from the event description field information about why the implant problems occurred, we created separate Oracle reports for each of the 13 penile implant models cited most often in the MDR tapes. The reports contained the full text of the event descriptions.

A macro WordPerfect program was written to search for terms and count frequencies. We counted the number of times such terms as "leak," "lost fluid" and "infection" appeared in the event descriptions. As explained earlier, the FDA's inconsistent reporting style makes this sort of measure imperfect. Here, however, the method served as a backup check to the hand-coded content analysis sample.

Another check came from the FDA, whose analysts are able to sort medical device problems by categories. U.S. News obtained FDA analyses of problem types for the 13 penile implants cited most often. Because the FDA's system did not always fit our needs, it was not desirable to rely solely on FDA data, but again it was a valuable check.

This multi-method approach produced three sources of information regarding specific types of penile implant problems: the content analysis of 1,196 records, the frequency counts from WordPerfect and the FDA analysis. From this we produced a box that ran at the close of the story listing the most common problems associated with various models of penile implants.

For media such as U.S. News that are just starting to utilize computer-assisted reporting techniques, the road to completing projects is filled with an endless string of potholes. As crucial as it is for reporters to narrow their research question and devise an effective methodology, it is just as important to be able to react positively when glitches occur. Whether or not such projects ultimately succeed depends in large part on the willingness of the news organization to assemble a team of players flexible and creative enough to solve the problems they could not have foreseen. For U.S. News and other media that have moved past discussing computer-assisted reporting ideas and actually produced publishable work, they now face new challenges. They must educate editors and reporters about the advantages of computer-assisted methods, provide year-round in-house training and make computer-assisted reporting techniques part of their daily routine.

Real help for data junkies

Keeping up with the wide range of developments in the field of public data can be pretty tough for reporters on the beat.

But the Association of Public Data Users can help with that.

A sort of data-junkie support group, the association provides a forum for interaction between data users, sources and producers through its newsletter and conferences.

The eight- to ten-page newsletter is packed full of information that would be pretty dry to anyone except a data addict.

The front page stories of the May 1992 issue of the APDU Newsletter were esoteric but informative.

"The Mother of All User Fees" updates readers on the latest round in the user fee debate: U.S. Courts might be given authority to fund facility costs through fees on the use of the judicial system.

Sharing the front page was an article about a proposal by the Office of Management and Budget to amend federal policies relating to information dissemination, records management and cooperation between state and local governments.

Inside pages are filled with short descriptions of new government and private data releases available through a variety of mediums. Some of this stuff is really far out, yet fascinating.

The Calendar section lists instructional seminars around the country.

Useful tidbits of information, including legislative updates and technical information, are sprinkled throughout the newsletter. For example, the Census Bureau has announced an error in STF3A data, and it looks like CD-ROMs are falling far short of their expected 20-year lifespan.

It costs $300 to join the Association of Public Data Users. Besides a newsletter, members get a telephone contact list for government data agencies and can share the cost of a database with other members. For information write to **APDU, Princeton University Computing Center, 87 Prospect Avenue, Princeton, N.J. 08544.**

Tech tips: Eliminating happy faces and other garbage characters

By Brant Houston
The Hartford Courant

Subject: Cleaning up data from a file on diskette so you can run a match in XDB.

Problem: You find out that a small state agency has collected information on thousands of businesses and keeps that information on a diskette. You want to merge that information with your own database of a couple of hundred business names.

Before you can worry about spelling differences in the names, you discover that the agency's file is, not surprisingly, fraught with problems, including garbage characters. And your problems aren't over after you import it into database software.

Solution: If you are a journalist who is just getting into computer-assisted reporting, you probably don't have programming skills. But that's not a problem if you have a quick text editor and use a little imagination with XDB.

One commercial text editor is XYWrite III Plus. Copy the agency file onto your hard disk as BIZINFO and look at it by calling it up through XYWrite.

There is the file in all its ugliness with ASCII characters (such as the ubiquitous and ironic happy face) popping up here and there. The nice thing about XYWrite III Plus is that it has an ASCII menu from which you can pluck the offending character and put in the header area.

By typing **CH /{ @}//** in the header area and hitting enter, XYWrite will race through the file and eliminate the happy faces. You can do whatever search-and-destroy missions you need to on other garbage characters. You can also eliminate unwanted spaces with **CH / //**, and add a character, if needed, with the same function.

Another nice feature of XYWrite is that it stores a backup of the original file in case you get excited by all the destruction and blast away an innocent character.

When you import BIZINFO as a comma delimited file, the quote marks around the company names also get imported. (This actually happened to me.) That means there are company names with quotes around them in BIZINFO, and there are no quotes around company names in your own file (call it BIZNAME).

Rather than taking awhile to figure out what is going on (which might mean fathoming bits and bytes, hidden characters and tabs, etc.), there is a quick solution.

Export BIZNAME from XDB as a comma delimited file with quotes around characters. Then re-import it as a fixed file,BIZNAME2. That brings those names back into XDB with quotes. Your company names now have a chance of matching because they have quotes around them in both files.

At this point you might want to use XDB's command **Xleft** to run a join on the first nine characters in name field in each file in order to pick up most of your matches.

For example:

```
Select A.*, B.Name
From BIZINFO A, BIZNAME2 B
Where Xleft (A.Name,9) =
Xleft(B.Name,9)
```

Make sure you index both name fields before running it.

You may have to clean up the data further by using XDB's **Update** command, but you are well on your way at this point.

I'm sure there are other effective text editors and that a little programming knowledge would go a long way toward solving these problems in a better manner. The bizarre importing of names is just an example to show how to clean and shape data by using the import and export functions in XDB. Nonetheless, these tricks do come in handy when importing data from an optical scanner or an online source such as the Federal Election Commission.

Correction

In last month's Tech Tips, Elliot Jaspin made some suggestions for dealing with various spellings of a company, such as "Amalgamated Zoot Suits," within a database called BADSPELL. But Uplink printed one command incorrectly.

The correct command is:

```
Update BADSPELL
Set NewName = "Amalagated"
Where Xleft(name, 11) =
"Amalgamated"
```

Our apologies.

If you are a journalist who is just getting into computer-assisted reporting, you probably don't have programming skills. But that's not a problem if you have a quick text editor and use a little imagination with XDB.

Bits, bytes and nibbles

Elliot Jaspin, MICAR's founder, has left the University of Missouri School of Journalism to take a position with Cox Newspapers in Washington, D.C. We'll miss his smiling mug around the office, but he's promised to continue advising us on technical matters and help out when he can with MICAR's programs. In fact, his first return engagement was last month when he came to help teach a week-long TRI-DART seminar.

MICAR and its programs will continue at the School of Journalism with some help from Investigative Reporters & Editors, a non-profit organization also based at the University of Missouri.

• • • • •

Bruce Melzer at the Anchorage Daily News discovered the feds don't know their state abbreviations.

When Melzer tried to call up Alaskan lending institutions in the Home Mortgage Disclosure Act data, he got Arkansas too.

Apparently whoever entered the data used "AK" as the abbreviation for both states at least part of the time.

HMDA data is divided into two files. "LARS Public" contains information about the loan and loan applicant. This first file lists lending institutions by a numeric code composed of two fields: "respondent id" and "agency code."

This code must be cross-referenced with the second file, "TS Public," which contains the name and address of the lending institutions, including two-letter state abbreviations.

Inaccurate state abbreviations make filtering the TS file by state difficult. But it's not likely anybody would need to do this.

Instead, use the numeric state code in LARS to filter for applicants in the desired state. Then search the TS file using the institution code to get names and addresses of lending institutions.

• • • • •

Uplink needs your input! Send bits to MICAR, 120 Neff Hall, University of Missouri, Columbia, Mo. 65211.

■ Jaspin departs for D.C. and Cox

■ More proof that our government officials can't spelle