

## Garbage in, money out

One of the most consistently successful ways of screwing up computer programs over the years is to give them silly inputs. In a good computer program, ALL inputs and outputs are checked for both internal consistency (for example has a numeric field been entered as a number ?), and domain sanity, (for example has somebody's age been entered as a negative number ?). Unfortunately this amount of checking takes time and effort so programmers still ignore it and trust the users to input sensible information. Such trust is woefully misplaced however and it doesn't seem to matter what environment programs appear under, Windows, the Web or Whatever, the same deficiencies appear again and again.

If you need a little convincing, then try this. In Word 2000, open the "Replace ..." widget and enter more than 255 characters in the Find field, (just press any key until it beeps at you). Now do the same in the Replace field. It will beep at you once again but when you hit the "Replace" button it crashes Word, (Unhandled exception in Winword.exe 0xC0000005 for those of you who spent Christmas Day programming). A fluke ? Not so. Try inserting a table with more than 50 columns and 32,000 rows. Word accepts it and then hangs. I could pay to upgrade and get a new set of defects but I can't be bothered. Incidentally I am not gratuitously bashing Microsoft here, most computer programs can be broken like this, <http://www.howtobreaksoftware.com/>.

In the Web environment, such problems are even more common. To do comprehensive field checking in a web program, you need to do both front and back end checking. It would be nice to use Javascript on the front end to check everything but this has two problems. First, script kiddies can get round it and second Javascript is full of bugs. For example, the version of Javascript I am using here will accept "3; rm \*" as a numeric field because it only checks the first character, (the rest of this field deletes files incidentally). Web programs should not rely on front end checking alone but on many sites, even this is not done properly. Some of the early credit card sites would accept negative amounts and actually credit the online shopper. A good web-site (and there still aren't that many) will do front-end checking of some kind as well as comprehensive back-end checking using ASP, PHP or similar.

Need a really convincing example ? Well, the whole purpose of this column was to describe the events of December 9th 2005. On that date, Japanese brokerage firm Mizuho Securities lost 224 million dollars because of, "a typing error". One of the traders had mistakenly offered 610,000 shares for sale at 1 yen rather than 1 share at 610,000 yen, (around 5,000 dollars). Fortunately it transpired that the Tokyo stock exchange itself applies bounds on price variation so "only" around 200 million dollars went down the pan.

The article I read was all about the unfortunate trader but it takes two to tango. This was no more than the same old story of sloppy programming and equally sloppy testing. It didn't require much input checking to have picked this up and it is fairly typical of the low professional standards which still haunt IT.

Happy New Year.

l.hatton@kingston.ac.uk