The Guru Code

from the editors

Edward Grossman, Editor, *ACM Queue*

No, this is not my attempt at a *Da Vinci Code* parody. The guru code isn't fiction, it's *real*. The guru code permeates everything—OK, maybe not *everything*, but at least the cable TV network in Connecticut.

You see, I was doing a little channel surfing the other day and as I flipped past the community access station I noticed something strange: a blank, black screen with a few letters of red text running across the top. Being a curious sort (and by that I mean “one who is possessed of curiosity” not “one whose behavior is odd,” though that may also be true), I flipped back and had a look at the text. Here's what it said (more or less):

Fatal System Error

Guru Code: #0002000436; #0001000258

What the heck does that mean? I mean, clearly some “system” had had a “fatal error” but “guru code”? Well, at least they're honest, I thought. That code must mean something, perhaps (gasp!) even something having to do with why the fatal crash occurred—but only a guru was going to be able to figure it out. On closer inspection I realized that there were in fact (aha!) *Guru codes* plural, *two* codes... hmm: $2 \times 0 = 0$. I was still clueless.

A couple of months back I asked what you the readers thought this page should be dedicated to—contextualizing the following pages, sounding off on a (hopefully) relevant issue to us all, or perhaps some other yet-unconsidered topics. Thank you all for your many kind responses. I'm only sorry to report that your suggestions broke evenly in opposite directions—50% saying contextualizing the issue was useful and 50% saying that a (longer) repeat of the table of contents wouldn't be as useful as exploring an issue of the day.

Never fear, I will not go the way of the “stupid mule!” (The stupid mule? Ok, here goes: It was related to me that some philosopher once posited as a thought experiment that if you were to put a stupid mule equidistant between two piles of hay, the mule [whose eyes are on opposite sides of its head, each eye seeing only one pile of hay] would starve to death, for due to its limited aptitude it would be unable to decide which pile to eat first, both seeming equally delicious. And the first person who finds and documents the origin of this story gets a free *Queue*

Does anyone

ACTUALLY KNOW WHAT

THESE CODES MEAN?

t-shirt.) So I won't go the way of the stupid mule, I will try *both* to shed some light on the formation

of this issue's contents and to rant about a problem, one that I believe is typified by the guru code.

This month's topic is error recovery—how software should be able to cope with system and pilot error. In my opinion, the guru code is an example of what not to do. It's my experience that even the gurus don't know what to do with such codes. Often the best gurus are left to look up equally obtuse definitions (“TmpVar not valid,” thanks a lot!) of said codes in some documentation a hired tech writer wrote 10 years hence when someone realized the original software author was no longer around, and well, we better have some documentation. Error messages don't help if no one understands them (or if the one person who does is so incredibly busy that you have to schedule a two-minute question with her three weeks out). Or worse, misleading error messages can actually lead well-intentioned people to try fixing things in ways that break them further.

But don't take my word on it, this month's authors are the experts: bad error messages hurt (see Paul Maglio and Eser Kandogan's “Error Messages: What's the Problem?” on page 50); systems should be able to cope with their own errors, as well as pilot error (see Aaron Brown's “Coping with Human Error in IT Systems” on page 34); and get the right error messages to the right people or programs for handling (see Brendan Murphy's “Automating Software Failure Reporting” on page 42).

When we first came at this issue, we invited error recovery guru Bruce Lindsay to come and help us think through the issues. He had such a handle on things, had such insights, we said “Wow, if we could only capture that—now *that* would make an interesting read.” So, that's what we did: *Queue* Editorial Advisory Board chair Steve Bourne got together with Lindsay and we've got it all down for you starting on page 22. Enjoy! ☺

EDWARD GROSSMAN is responsible for *Queue*, so blame him if you don't like it: edgrossman@acmqueue.com.