The Trouble with OO?

Posted At : May 26, 2009 6:20 AM | Posted By : Steve
Related Categories: ColdFusion, OO Principles

Marc Funaro recently posted a blog entry called "**How OO Almost Destroyed My Business**". It is a good entry, inasmuch as it is thought provoking.

It is a bit long, however, so hopefully he will post a shorter, more organized entry to the same effect later. Even at its long length it is a worthwhile read. The stream of consciousness style does help demonstrate his experience and feelings.

I started to post a comments, but my thoughts ended up running a bit long.

Like the author, I have often felt that the gurus in the ColdFusion universe are pushing an OO solution. This isn't really a fair interpretation of what anyone has actually said, however.

Most ColdFusion gurus are smart and experienced and tend to be working on difficult problems. The kinds of problems that they solve are probably the kind for which an OO solution is the best kind. So, most ColdFusion gurus prefer OO for themselves. Consequently they write about doing that kind of work.

I don't think the intent is to suggest that to be "The One True Path", it is just an honest description of the work that they are doing and find interesting.

Written internet communication tends to be either brief or stream of consciousness. Either one tends to lead to writing that can be interpreted as less nuanced than intended. This blog entry, for example, seems to take the stance that all ColdFusion gurus are pushing OO and OO is always bad. Probably (hopefully), this isn't what the author intended.

In the same way, ColdFusion gurus, I think, don't intend to push OO as the only way to develop.

I have, however, seen writing in the ColdFusion universe (no, I am not going to try to find examples) that indicates that anything other than OO is procedural and implies that anything procedural is spaghetti code. Examples tend to say, "You could do without OO, but you have to copy and paste" or they talking about the benefits of encapsulation as though you can't use encapsulation outside of OO. All of this uses procedural as a straw man for "spaghetti code".

This is unfair, but also just a part of casual writing. The author of the blog, for example, may not have intended to come of as aggressively against ColdFusion gurus and OO as he seemed to do. Even the comments are a mixed bag in this regard. Some are very well thought out (see Mike Brunt, in particular). Others, even by people I know and respect, seem defensive and/or aggressive to the author's experience (he does seem to blame OO for all of his travails, after all) or cheering any stance that OO isn't the way to go.

I think many of the comments supporting the blog entry come from a feeling that OO is the only accepted way to program in ColdFusion today. I think that isn't actually because anyone says that, but rather because nobody is writing about how to develop in ColdFusion without OO.

For my part, I intend to change that.

I have a non-OO framework that I have been using internally. I haven't blogged about it because it isn't OO and it isn't fully baked. I didn't want to defend it against OO purists until it was fully baked.

This is absurd and unfair on my part. I am not giving the ColdFusion community enough credit for being even handed in its assessment of different approaches to solving problems.

I plan to start blogging about these topics soon.

Stay tuned.