

Framework Presentations: COOP and ColdBox

Posted At : November 30, 2007 8:00 AM | Posted By : Steve

Related Categories: Frameworks

The [Online ColdFusion Meetup Group](#) had two presentations yesterday and I was fortunate enough to be able to attend both. The first one was from John Farrar about his COOP framework and the second was from Luis Majano about his ColdBox framework. I haven't used either of these, but I have been interested in both for some time.

COOP

Much of [COOP](#) deals with form handling. Some of it fills the same space as [cf_sebForm](#), and we have even made similar decisions on how to solve similar problems.

His approach centers largely on a co-processor. This is similar to how I use [page controllers](#). Both are files named the same as the file called from the browser except using a ".cfc" extension. His framework calls the file implicitly, whereas I call the page controller explicitly from each file. His approach saves code by using a convention, where I opted for slightly more code to allow for a bit more flexibility. I don't think either approach is really better than the other, each have advantages and disadvantages.

With his co-processor, he has a concept that he calls pre-DOM coding. This essentially allows him to set values and attach event handlers (and perhaps more) to DOM elements in the co-processor. The idea is to allow programmatic control from the co-processor so that the .cfm page remains clean from anything that looks like coding logic. This allows a designer to edit the .cfm page without being exposed to logic that they could find confusing and with which they could potentially cause problems.

I like this concept, but I don't like that it takes priority over settings created directly in the elements themselves. This violates the standard of having the closest setting win (common in HTML and CSS, for example). I have the same issue, and address it with page controllers. I do think that the pre-DOM aspect is helpful, but would be better if the code in the page had precedence.

He also had several examples of different forms and widgets that could be created with COOP. I tried to download them this morning from the [COOP Google Group](#), but had no luck.

I liked what I saw in COOP and I thought it was an impressive approach.

Would I use it? I'm not sure. In page controllers and my own custom tag set, I have a similar solution set already. Still, the pre-DOM coding concept looks interesting.

Would I recommend it to others? Absolutely, I like the way he solves several problems. I think anyone considering page controllers or my custom tags should also look into COOP and choose which approach best suites them (or perhaps a combination of both as the two could potentially work together despite some overlap).

ColdBox

I have been watching the development of [ColdBox](#) for some time. I have talked to a few people who have tried it out and heard unanimously positive comments about it. As such, I was looking forward to the presentation.

The presentation did not disappoint. Unfortunately, I was unable to take good notes during this presentation. I had a few interruptions on my end during the presentation. What I saw, however, was very impressive.

Among the features that he covered is his "Smart Cache" caching mechanism. This sounds very impressive and deals with a problem that I have long known that I should deal with, but that I have yet to get to in an elegant manner. I don't remember the details on how it worked, but it his highly configurable and sounded really impressive.

He says that he has written over 250 pages of documentation for ColdBox. That is extremely impressive. I was proud of myself for having over 15 pages of documentation for DataMgr, but 250 is really impressive. I think this is something that most open source developers (myself included) should be better about - hats off to Luis for helping to make adoption easy.

Another thing that really impressed me about ColdBox is that several of the features dealt with visibility of the processing and data happening under the covers. It really does a good job of letting you know what is happening under the hood.

All in all, ColdBox looks really impressive. I expect that it will quickly rank along [Model Glue](#), [Fusebox](#), and [Mach-II](#) as one of the major MVC frameworks in ColdFusion.

Would I use it? I wouldn't use it personally only because I don't like hub-and-spoke frameworks (where all requests go through one file). I know that it has SES URLs, but that isn't enough for me.

Would I recommend others use it? Depending on their situation, I would absolutely recommend this to others. I think hub-and-spoke is wrong for my situation, but I think it provides no drawback at all for many others. Outside of the drawback of hub-and-spoke, ColdBox looks like a beautiful framework.

I will definitely download it to get ideas on how to better improve my own code structure.

Hopefully I provided an accurate portrayal of both frameworks. If anyone notices and inaccuracies, please let me know.