

In This Issue

- **A taste of SDI's latest development project, happymappy.com.**
- **Wally takes a look at the pros and cons of ASP.NET MVC**
- **"The Dogs Won't Eat the Dog Food"**

HappyMappy

By Luci Harrell

Much of the work here at Scalable Development includes revamping and updating client systems. We want our clients to be able to utilize database applications with the most efficiency possible, helping them to stay at the top, technologically and performance-wise, of their respective fields.

But there's another side to Scalable Development. Because we pride ourselves as a forerunner of new technologies, this includes working with startup web driven companies to deliver cutting edge products that surpass anything on the web today.

At the turn of the year, Scalable Development's biggest success of '07 was getting HappyMappy.com off the ground. HappyMappy is a free, interactive, online search and mapping database. It allows users to find 'fun things to do' across the U.S. The application is searchable by keyword, establishment name and city, and allows users to zoom and browse on their choice of Google Maps, Microsoft Virtual Earth or Yahoo Maps.

Though HappyMappy is still in Beta format, it already contains over 109,000 points throughout the United States. Points include 'about' information, addresses, phone numbers and website links, if available. For more information about HappyMappy, visit

www.happymappy.com. ■

A Look at ASP.NET MVC

By Wallace "Wally" McClure

A friend of mine recently posted his thoughts on ASP.NET MVC. It got the idea swirling around my head, too. I think that MVC has some notable pros and cons. Some of the things I like are:

- The UI is fairly lightweight with logic. Ideally, it would be only a shell that displays data. Common UI functions are pulled back to a separate layer and are processed within the server. These UI functions can be used across pages.
- ViewState. A developer is now in charge of managing state; this includes the implied statemachine, which managed the events that were formerly fired in an ASP.NET Webform. If you don't want ViewState, there is no ViewState.
- Testability. Ideally, applications are much more testable.

On the other hand, if we look at these items in a little more detail, things might not be quite as nice as we once thought. But that doesn't necessarily mean that there aren't solutions. For example:

- The UI can be fairly lightweight logic, however, someplace there has to be business rules that are implemented and communicated back to the client layer. How am I supposed to do this with MVC? I figure that I am missing something, and that further research is necessary on my part.
- ViewState. Hmm, I've never had a problem with ViewState. If I think ViewState size will be a problem, I can just store ViewState on the server by overriding a couple of methods in a base page and, voila, problem solved.
- Testability. Testability is good, but not great. But that's not the problem. As I've found before, the reality is that some customers are simply not willing to pay for testability. It's sort of like selling 'Total Cost of Ownership' about 15 years ago. Also, please note that testing of Webforms is pretty painful.

See MVC on Page 2

MVC

A couple of myths that deserve to be squashed right now are:

- The payback for using MVC comes for any project that takes longer than two months. Hmm. I don't buy the 'two months' argument. I think that there can be a benefit to using MVC; however, it would depend on the projected project size. I'd like to see an analysis come from someone like Gartner, or some other reputable consulting company, first.
- ViewState is not an evil thing; it is good. It provides a lot of features and functions. It can grow and be very large. We need to acknowledge that. With a few overrides of some methods in a base page, we can work around this. It's not a great solution to the ViewState problem, but it can be made to work.

Having said the above, I'm actually very excited about ASP.NET MVC. I have been and will continue to be looking at it a lot over the next few months. I think that it incorporates some really great ideas, but I don't think it will be replacing Webforms any time soon.

Ok, but what, exactly, is MVC?¹

Model-view-controller (MVC) is an architectural pattern used in software engineering. Successful use of the pattern isolates business logic from user interface considerations, resulting in an application where it is easier to modify either the visual appearance of the application or the underlying business rules without affecting the other. In MVC, the Model represents the information (the data) of the application and the business rules used to manipulate the data, the View corresponds to elements of the user interface such as text, checkbox items, and so forth, and the Controller manages details involving the communication to the model of user actions such as keystrokes and mouse movements. ■

¹ MVC definition obtained from Wikipedia. "Model-view-controller." *Wikipedia, The Free Encyclopedia*. 21 Apr 2008, 13:24 UTC. Wikimedia Foundation, Inc. 21 Apr 2008 <<http://en.wikipedia.org/w/index.php?title=Model-view-controller&oldid=207109857>>.

"The Dogs Won't Eat the Dog Food"

An oldie but goodie (as it relates to software development)

By Wallace "Wally" McClure

Everyone hears the phrase, "The dogs won't eat the dog food." But what does this mean? You hear the term a lot with people from larger software corporations. They talk about 'dogfooding' a product.

I have heard this term, along with the entire story, all of my life. However, I've never seen the full story online. I'd like to take a moment to share the story as it been relayed to me my entire life.

Company A makes dog food. They have just released a new brand of dog food. Unfortunately, the dog food is not selling very well.

"If the customer doesn't like the product, then nothing else matters."

A very upset president of the company calls a special meeting. All of the C level executives are required to attend. The president goes around the table, one executive member at a time, asking for specific information from each. What is the reason that the dog food won't sell?

- The person in charge of marketing talks about how marketing for the product is correct, and that the company is everywhere else that every other dog food vendor is.
- The person in charge of manufacturing talks about how the product is being made per specification.
- The person in charge of distribution talks about how there are no problems in shipping.

Finally, the person in charge of testing stands up and says, "Yes, this is all great, but the dogs won't eat the dog food!"

Moral of the story: If the customer doesn't like the product, nothing else matters. ■

Scalable Development, Inc.

Atlanta Office

2900 Chamblee Tucker Rd.
Building 5
Atlanta, GA 30341
770-458-6590

Knoxville Office

118 Durwood Rd.
Knoxville, TN 37922
865-693-3004

www.scalabledevelopment.com