

Critical Mission

If you would like to view previous newsletter articles, visit our web site at www.AltoConsulting.com.

FEATURED ARTICLE

By
Craig Yellick, Vice President



Silverlight 4 and RIA Services Architects Rejoice

2010 is the year that enterprise application architects finally received all the tools needed for rapid development and cost-effective maintenance of distributed software solutions. We have all suffered enough with the Frankenstein lash-up of HTML, CSS, HTTP POST and GET, XML and JavaScript. Microsoft chucked the bad parts and kept the good with Silverlight version 4 and Rich Internet Application (RIA) Services.

A Rational Platform for Business Apps

Silverlight version 4 and Rich Internet Application (RIA) Services are the new standard for enterprise application design and implementation. 2010 is shaping up to be a very exciting year for software architects.

Anyone designing distributed software applications for the past 20 years has seen terrific progress but also significant frustration in how applications are put together and maintained. Anyone who has been around the block a few times knows where the pitfalls and shortcomings lie in the user interface, business logic and data layers. Until Silverlight 4 + RIA Services, the vision of how a complex, distributed system was supposed to be implemented usually did not translate into secure, scalable, manageable code because the layers have all evolved more or less independently. There is too much reliance on

developers to write complex code patterns consistently and reliably.

Silverlight 4 + RIA Services bring together a vision of a distributed application architecture that exists within the tools, base classes and generated code rather than in the hands of individual developers.

In short, in properly architected systems based on Silverlight 4 + RIA Services, **the easiest way to do something is also the best way.**

In this newsletter I'm going to highlight four major advances in Silverlight 4 + RIA Services:

- Generated data access code and one-stop declarative validation that works in all layers
- Elimination of the hassle maintaining state in web applications
- Embracing the reality of the need for asynchronous web service calls
- User interface styles and themes formally implemented through object orientation

Data and Validation

Code generators have had a bad reputation in the past, and often for good reason. Nonetheless, you'll want to take advantage of RIA Service's ability to generate an Entity Domain Model directly from your relational database, and then use the EDM to generate a fully-functional Domain Service callable from any client application. Silverlight clients can exploit the Domain Service through declarative binding, meaning programmers do not write the same source code patterns over and over to fetch data, display it, and optionally allow edits. This seamless, repeatable EDM-to-DS process eliminates a lot of tedious coding and makes developers far more productive.

The generated Domain Service classes use the current state-of-the-art Windows Communication Foundation (WCF) model for client-server communications. Your applications get all of the performance and security advantages of WCF without requiring developers to write all the code.

Validation has long been a trouble spot in distributed applications. To provide a good user experience the client needs to validate user entries and provide feedback and error messages. While at the same time the web service-based data service layer has to repeat the same validations because it cannot blindly trust that the client (which may in fact be an attacker) has performed them. Prior to Silverlight 4 + RIA Services the solution was to implement the same logic in multiple places and hope that developers were consistent and complete. Instead, validation logic is implemented once as part of the Domain Service and the Silverlight toolset ensures that the same code is packaged with the client.

Further, common validation rules like “this field is required” and “this value must be positive whole number between 1 and 100” are implemented using declarative syntax. Coding is only required for complex validations that can’t be expressed with the declarative syntax.

No State Management Hassles

Web application developers are all too familiar with the complexity and compromises required to overcome the stateless nature of HTTP GET and POST. ASP.NET has long offered a variety of tricks to get around the limitation, from hidden form fields to session data held at the server. Because Silverlight applications are true client applications that run at the user’s workstation they are free to create and maintain as much data as they desire. The RIA Services Domain Service handles the details of marshalling rich data types and data sets so the developer concentrates on adding value to the application and not worrying about the framework and other plumbing details.

Asynchronous Made Easy

Similar to state management hassles, developers implementing web service-based applications are faced with difficult decisions about when to make an easier synchronous web service method call versus designing, implementing and calling a more complex asynchronous call. Methods found in a Domain Service are all asynchronous all the time. Developers implement service methods and let the framework do the rest. At the client, Silverlight makes it easy to hook into the asynchronous event model so that doing it right is also doing it easily and efficiently. Asynchronous operations stop being difficult, special cases and become routine, resulting in better performing, more scalable applications.

Styles/Themes Through Object Orientation

The slapdash lash-up of HTML, CSS, JavaScript and other styling technologies makes for complex and often even unpredictable user interface designs. Over the years ASP.NET has simultaneously embraced the model and also extended and abstracted it with Themes and Skins. The net result is often so complex that every future change has to be treated as a special case so as not to disturb existing screens, because for non-trivial cases nobody truly understands how it all fits together.

Silverlight user interfaces are based on XAML, a markup language designed specifically to address the many shortcomings of HTML. While you are certainly free to make a total hash of your user interface – developers still have the power to handle special cases – a smart architect will specify that implicit styling be implemented using the object oriented nature of XAML and the Silverlight application framework.

Case Study: Extending SharePoint Online

SharePoint Online, the version of SharePoint hosted “in the cloud” in Microsoft data centers, is a major component of the Business Productivity Online Suite (BPOS). Because the user interface portion of a Silverlight application runs at the client and doesn’t involve custom binary executables or libraries installed

on the server, you can bridge the gap between hosted SharePoint sites and enterprise data and services.

Silverlight is the future of cloud computing, whether you are moving users to BPOS-based application software or working behind the scenes with services like Windows Azure AppFabric and SQL Azure.

Summary

Silverlight 4 and RIA Services is a potent combination for business application development, especially for solutions involving cloud computing.

If you're curious about Business Productivity Online Suite and want to learn more, Alto offers a training class designed to get you up to speed, fast on the concepts and address all your questions and concerns.

Contact Alto if you'd like a demonstration of any of the products or techniques discussed in this article. Our next solution briefing is devoted to these concepts.