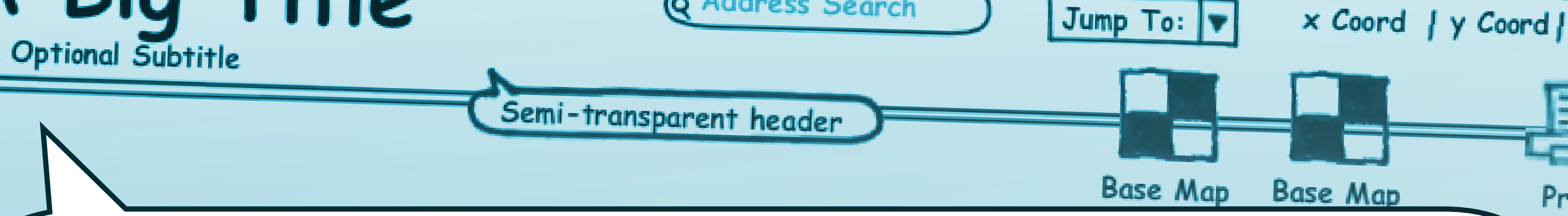

Remix Those Sample Viewers

Elevate your RIA and stand out from the crowd

By Brian Noyle and Dave Bouwman, DTSAgile

Ladies and gentlemen, we are facing an epidemic…an epidemic of "sameness."

ESRI has provided the development community with starter kits for ArcGIS Server. These sample viewers, available for the JavaScript and Flex APIs, are a great starting point for creating map-centric Rich Internet Applications (RIAs). The pluggable nature of starter kits facilitates significant reuse of components. There are dozens of plug-and-play components available to developers in the ESRI code galleries.

While the speed and ease with which these sites can be built and deployed is excellent, the downside is the wave of cookie-cutter sites emerging on the Internet. It is clear that in many cases, little or no thought has been given to originality, branding, or how the user should interact with the site.

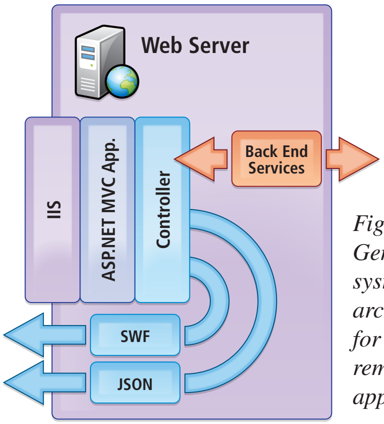

Figure 2: Generalized system architecture for our remixed application

In comments made on a blog post on this topic by one of the authors (see blog. davebouwman.com/index.php/2009/11/re-mixing-the-flex-sample-viewer/), Stu Rich reminded us that the power of the sample application is that it proves the critical path and provides us with many different tools to choose from when building applications in the RIA space.

At the same time, the job of the thoughtful designer/developer is to choose just the essential tools and design them so their use will be intuitively obvious. At the same time, the designer and developer should take care to make each site distinctive. Sites should reflect the brand and intent of the organization. You wouldn't lift someone else's site en masse for your corporate site, so why would you simply copy another site's mapping apps?

Since ESRI has done most of the heavy lifting and provided very pluggable, reusable frameworks in multiple Web technologies free of charge to the development community, we assert that a little extra time and effort is all it takes to elevate offerings to the next level and provide a thoughtful, attractive, and unique experience to users. In this article, we'll point out a few of our beefs with cookie-cutter sites and illustrate the results of our efforts to take a starter kit and remix the interface to produce a more highly polished site.

Remix from a Wish List

As we've noted in our published articles, blogs, and presentations, we like our applications to be as focused as possible. Every mouse click and drop-down menu that we can remove from an application improves the user's experience. Here are some items selected from our (incomplete) user interface (UI) and application behavior wish list.

- We want top-level menu items that do something on-click as opposed to always revealing a pull-down menu. A top-level print menu should open the print dialog box when it is clicked rather than reveal a single menu item that is also called Print.
- Many RIAs rolling out the door these days are information exploration tools. In the age of Google, search is king. A search box should always be available and occupy a prominent place in the UI at all times.
- If this site will have a full bleed or wall-to-wall map, we want a full-width banner with plenty of room for commonly used top-level items.
- We want a limited set of basemaps available and want them displayed as top-level icons on the menu. Clicking them will turn on the layer. The active layer should have a glow indicating that it's…well…active.
- We never want to hide navigation down a level in the application. Navigation and bookmarking should be front and center.
- We want widgets that can be set to never close. We've frequently seen cases where users can't figure out how to reopen a widget that has been closed.
- In numerous other usability rants, we've expressed the philosophy that map content should be limited to a basemap and some minimum number of operational layers so that legends and layer lists become superfluous. If they must be shown, table of contents features should be inconspicuous.
- Whichever starter kit is chosen, the look and feel must be customized for each client or delivery. It's not enough to simply replace the ESRI logo and move on. If possible, use custom icon packs and unique skins.

The Remix in Action

In our example scenario, we'll start with the Flex Starter Kit and build a data visualization application that allows foresters and policy makers to explore data pertaining to the effects of common forest pests in the United States. Conversations with target users convinced us that navigation should be minimal and search is paramount. In addition, we're willing to sacrifice map space to accommodate a panel on the right side due to the richness of the data. Finally, since we've been around the block a time or two, we've got some ideas for a novel way to visualize pest impacts on the landscape in time series so that needs to be worked in as well. This example isn't going to use all the items in our wish list, but we'll pull a few of the ideas in and let the rest of the list serve as fodder for future sites.

Sketch It Out

Whenever we start out on a new project or new set of features for an existing project, we like to develop wireframes (among other things) for the simple reason that debugging and accommodating changes on paper is much cheaper than fixing things once the site is built. Depending on the project and client, we'll create pencil and paper sketches or use Microsoft Visio or products like Balsamiq or the Pencil Project to create mockups. Figure 1 shows how our Flex Starter Kit remix looked on paper.

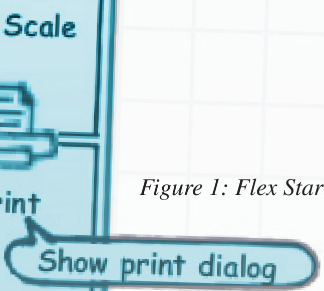

Figure 1: Flex Starter Kit remix on paper

The Architecture

The technology stack for our remix came together as shown in Figure 2. Our ASP.NET MVC application has the primary job of returning our Flex widget (*.swf) for rendering in the browser when a user lands at our site. In addition, a variety of other resources have been written into one or more MVC Controller classes to access backside services and return information needed by the Flex application as JSON. For example, as the user zooms and pans around the map, the application will request updated pest statistics and additional data for the current area of interest.

The Final Result

Figures 3 and 4 show the results of our remix and skinning efforts using the Flex Starter Kit as the starting point for our application. Note that the UI includes the full-width banner, the address search, zoom-to drop-down, an "always open" widget, two basemap icons, and a top-level print widget. The tiles in Figure 3 are from the CloudMade Midnight Commander cache. Figure 4 is a map of Colorado based on OpenStreetMap tiles. Note the Pest Conditions Explorer widget including options for changing the year of the data being displayed as well as selecting an individual pest map to display.

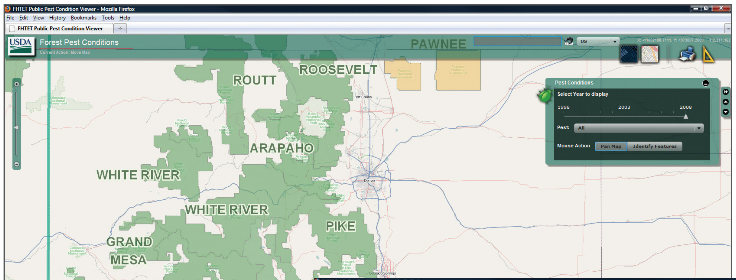

Figure 4: State-level view of forest pest data for Colorado is based on OpenStreetMap tile cache.

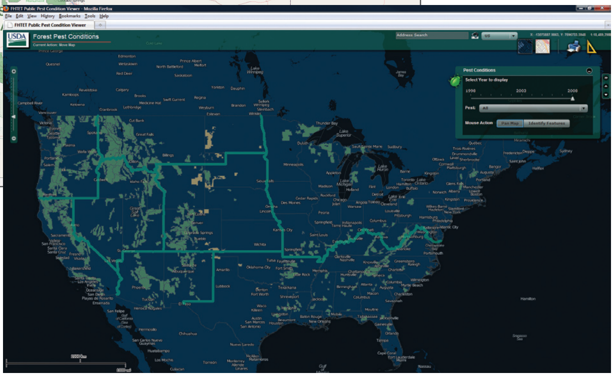

Figure 3: Nationwide view of Pest Data Viewer Flex Application shows regional-scale data.

Our Conclusions

In an environment where getting up and running in the RIA space quickly is easier than ever, differentiating your application from the cookie-cutter sites on the Internet is critical in distinguishing your organization or client's site from all the others. As developers, we can dramatically improve the impact of sites based on the ArcGIS Server Web API starter kits by spending a little time identifying where the user experience can be improved by undertaking a modest development cycle, dedicating resources to UI mockups, and making efforts to skin and brand sites. ESRI has done the bulk of the heavy lifting to make development against ArcGIS Server easy, efficient, and reusable. What's left is for the development community to spend some time remixing the current starter kit offerings and elevating the content and quality of the sites produced.

Once the developer has gotten used to working with an ESRI starter kit, a more concerted effort can be made to do a fully custom implementation. After releasing the publicly accessible site described above, we went back and built a customized site that fully leveraged the richness of the data available. Key data exploration features include a dynamic vertical bar chart showing both total acres affected and the total number of counties affected by major forest pests. Up to five pests can be shown at any given time. The bar chart includes a slider control allowing the user to change the year for which data is being viewed. When a user is viewing the entire United States, clicking on a region polygon generates a dynamic vertical bar chart of total acres and total counties affected by major forest pests. As the user zooms in on the map, more granular detail is shown. Region and county selection can be done via the map or via pick lists in the search pane located at left pane. Data summaries and links to external information are also shown in the left pane. Included in the tabular data summaries in the left pane is a function to generate a chart showing pest damage trends for all years in the system and view information on pests of interest.

What's nice about the application now is that the user does not need to go digging around in a menu system to access the functions to efficiently explore data. The template we've come up with will likely be the starting point for many other focused applications. "Do one thing and do it well" is the mantra. Keeping critical functionality up front and conveniently located is of paramount importance.

For more information, contact

Brian Noyle, Senior Software Architect
E-mail: bnoyle@dtsagile.com
Or
David Bouwman, CTO and
Lead Software Architect
E-mail: dbouwman@dtsagile.com

About the Authors

Brian Noyle, originally trained as a global change biologist and tundra botanist, has nearly 10 years of experience as a GIS software developer and architect. His professional and technical interests are primarily focused on moving clients toward more standard architecture and development practices and patterns to facilitate a closer integration of GIS with the standard IT enterprise. Noyle has extensive experience in full software lifecycle management with a focus on delivering through Agile project management methods.

Dave Bouwman has been designing and developing GIS software for the last 12 years with projects ranging from small Web sites to statewide enterprise forest management systems. Over the last few years he has been leading a team of developers in the pursuit of great software built in a sane manner. The combination of an Agile process with pragmatic development practices taken from extreme programming has led to a highly optimized methodology of creating solid software.