SEPARATING PRESENTATION AND CONTENT

Our approach is to separate presentation from the rest of the page and to separate the rest of the page from content. Why?

The main reason to separate presentation from the rest of the page is to simplify design changes, from a slight adjustment to a full redesign.

The reason to separate the rest of the page from content is, likewise, to facilitate updates while maintaining presentational consistency throughout the site.

More importantly, separating presentation from content allows us to easily reuse modules for our dynamically generated pages. Not having to re-write this code for every project reduces development time. Reducing development time saves money for our customers.

This approach also facilitates presentational consistency throughout our modules.

HOW WE DO IT

There are many ways of creating web pages. They include: HTML, SHTML, XHTML, JSP, ASP, Perl, PHP, Cold Fusion, etc. We want our sites to be able to run on any Web server. Our customers shouldn't be married to a specific platform. (For example, ASP can only run in a Microsoft environment.) This would limit the sites portability. So we use XML pages for the content, XLS "skins" for the layout and CSS Style Sheets to control the display of the elements. For the development of our applications we use a Java based Web Development Library that we've created ourselves.

SCOPE INTERACTIVE CSS XSL PHILOSOPHY

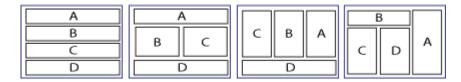Let's start with a few definitions. What is CSS and what is XSL?

CSS stands for Cascading Style Sheets. Style sheets define how HTML elements are to be displayed. Styles are normally saved in external .css files. External style sheets enable you to change the appearance and layout of all the pages in your Web, just by editing one single CSS document.

XSL stands for eXtensible Stylesheet Language. It's a family of transformation languages which allows you to describe how files encoded in XML are formatted or transformed. The syntax of XSL is valid XML. We have at least one XSL skin and sometimes several for each project.
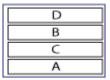
WHY BOTH CSS AND XSL?

Why do we need both CSS and XSL? Can't you do this with just CSS?

While it is true that you can control how something displays with CSS, the one thing you can't do is change the order in which elements display. To illustrate this point all of the following represent the same 4 HTML elements whose layout is changed with a different CSS style sheet.

This certainly gives you many possibilities. But here is something you can't do with CSS.

You can't change the order in which the elements are presented.

That is where the XSL skins come in. Using them you can change the order in which the elements are presented, you can add elements that weren't there or you can remove elements. (Note: You can't remove elements using CSS, but you can set them to not display, in effect removing them.)

WHY NOT JUST XSL?

Why use CSS at all? Although XSL skins can control the layout, you still need the CSS styles to control the display of the elements. XSL can't tell the browser what font to use, what color to make it or what the margins should be. All of that and more must be controlled using CSS styles. Also if you use many XSL skins instead of CSS styles, you loose the ability to change one file and have those changes cascade down to all the pages.

THE SCOPE OF THINGS

When the World Wide Web started, page content was "hacked" into layouts using tables for something they weren't intended for, forever marrying the data with the presentation. That makes changing it a nightmare.

Using XML, XSL and CSS gives us the ability to separate design from content. This allows us to make design changes when wanted, create dynamically generated modules that reflect the flavor of the site and make sure everything validates according to W3C standards. The result is a site whose cost is reasonable to build and maintain, that works now and will work in the future; no matter what the next great technology will be.