StyleSystemProposal

Why do we need a new styling system?

TikiWiki's current styling system is riddled with fragmented, sometimes ambiguous code. TikiWiki's current styling system also lacks modularity, W3C web standards (particularly CSS2) validation, a user-friendly interface for modifying current &/or inventing new themes, a well-documented resource for users, admins & developers a-like.

What will the new system do that the old system can't do?

The new styling system aims to provide anyone using Tikiwiki with the ability of customizing Tiki's user interface to the extent they desire, regardless of their skill/knowledge base regarding web development technique's.
The new styling system also seeks to maximize the efficiency & productivity of developers by allowing them to focus more on the code, rather than the presentation.
Using firmly founded web standards, such as remote Cascading Style Sheets (CSS), TikiWiki's code will eliminate localized styling of XHTML elements, thereby allowing for complete customization of Tikiwiki on both an administrative level, and/or a user-level.

Nice Ideal, but how?

As you may know, Tikiwiki utilizes the power & functionality of several components in one comprehensive, feature-packed internet experience. Some of these components include PHP, the Smarty Templating engine, Cascading Style Sheets, and many others. The following sections will outline key topics for discussion & provide one or more possible enhancement for that topic.

The Directory Structure

As the directory structure currently stands, there are four directories worthy of note. There are two directories in the root directory: styles, and templates. The styles directory holds existing style sheets, with subdirectories, each named according to their respective styles. Within those subdirectories are pictures (such as icons) specific to that style.
The templates folder has several folders, and close to 300 templates that work in conjunction with one another, Smarty, and PHP to bring about the fully interactive experience you're probably using to read this document right now. One of the subdirectories within the templates directory is called, "styles"??
Within the $root/templates/styles directory, there is a folder for each the existing themes that have custom built templates. This raises questions regarding performance, as well as directory simplification.
The favorite saying of my former economics/government teacher comes to mind, "Keep it simple, stupid??"
His saying that was the less than subtle reminder that the highest quality solutions are often much more obvious than we originally think.

Directory Structure - Solution

As one might expect, the directory structure can, in many ways, serve as the foundation for a logical, intuitive web application. Therefore, this proposal suggests combining the directories discussed above into a directory called "themes"?? The themes directory will have a templates directory and a style directory.

Levels of Customization

Currently users have two options when it comes to customizing Tiki's appearance. Depending on the permissions set by the administrator, a user is able to modify templates, or create a user-specific CSS file.
Having access to these two methods of customization provides complete access to customizing the user interface. This, however, does not provide an easy, accessible way to fine tune pages, or even sections.

Levels of Customization - Solution

Providing administrators & users with various levels of customization will open Tiki to a completely new & different way of thinking, and working.
First, extend Tiki's current administrative options section to provide the option of how the site "feels??"
For example, setting the entire site to the standard three column layout, a two column layout, one column

layout with distinct top & bottom sections, etc., and also allow the administrator to pass this option along to users as a user-specific setting (i.e. give the user the option of what type of layout s/he prefers). Setting the general layout should then automatically filter the available themes to only those that work with that layout. Themes that work with multiple layouts should obviously be listed accordingly. Once the available themes have been loaded, the administrator should have the option to do each of the following: (a) apply a selected theme as the default theme for visitors/anonymous users, (b) require that users use a specific layout (and thus, only certain themes), and/or (c) disallow users from selecting themes or layouts.
Second, extend Tiki's current user-specific style-related options. This level of customization includes several sub-levels in which several very cool, interactive user-customizations can take place. Please note that these features can be setup for administrative access only, if need be. The following outlines these interactive features:

- Provide users with a 'make it yourself' theme visual theme designer. This concept works off the WYSIWYG framework by providing users with a front-end to building their own theme for TikiWiki. The user/admin is provided with two browser windows. One is probably 'maximized' (depending on the monitor's resolution), while the other works as a remote control for that window. The maximized window is Tiki as you would expect it to be. You can surf, and completely utilize Tiki's full functionality. The second window, however, works via JavaScript to seamlessly create a theme builder that is updated on the fly, and without any downloading of any new pages. When starting the theme builder, Tikiwiki will ask the user if s/he would like to make use of typical styling techniques (i.e. if the background-color is set identically for multiple styles, then continue with that similar formatting). If the user selects yes, s/he will be provided with a condensed remote. Selecting no will allow the user to customize every nook & cranny of TikiWiki's style. I (musus) am currently in the process of building this feature.
- Enhance Tiki's current edit template feature to include it in the above WYSIWYG editor so that users can edit both templates & style sheets on a per page-basis. The WYSIWYG editor should allow editing of content, but not of code (though a back-up editor for complete access to editing the template should still exist). Again, musus is in the process of developing this. Help is definitely welcome though.