

## ContentUnification\_gmuslera

This is not intended for actual Tikiwiki, maybe could be used for 2.0 if is mostly a rewrite, or for another similar program that as Tikiwiki, combines in a single interface a lot of the tiki features... in fact, i could be perfectly describing how other existing system works, but not being aware of it. Also, to make things simple, i will focus mainly in the text-based content, like wiki pages, articles, forums/comments, blogs and html pages, but probably could be extended to most tiki content features.

### The problem

Tikiwiki have a lot of text-based content features, but because of its architecture, each one is different of the others. Use different tables, the text could have different syntax and rules (for articles i have a side graphics, for wiki have a history of changes, some are html, others wiki, other can call plugins, in blogs i could choose to have a wysiwyg editor, etc).

Those exceptions make the system as a whole more difficult to use and understand, less intuitive, and sometimes leave users wanting to have in one content type features of another.

Also there are things that are not so easily portable to other features, i.e. multilingual pages. Or content that is in a category and not in others, i.e. if you can go back and forth with a content between wiki pages (for syntax, collaborative edition, etc) and the article system or a blog (i.e. most of what i have wrote here could fit in another systems blog, if I add a "publication date" and see them ordered by date).

### Unifying content

Now, what in a content makes it to need to be a wiki page, or a blog entry, or an article, and not all the others? What if we separate the actual content from the way it is displayed or related to others? That comes a bit from the hand of ContainersVsContent\_gmuslera, and a bit from what is discussed in TikiNamesRevisited\_gmuslera, specially the part of embedding related with namespaces.

### Content table

Lets suppose that we have an unique texts table, where you have as attributes the text itself, the format (that could be plain text, wiki, html, even other wiki syntax/structured text/etc), language, author, title, name, etc.

That is the content itself, and how and where is shows depend of upper level containers.

### Containing content

Thinking more in data representation than in programs that shows it, a particular piece of content could perfectly be a wiki page, but also shown in my personal blog, announced in the site's today articles belong to several namespaces/categories/structures. The same could be applied even to non-text content, i.e. images, flash animations or other kinds of files.

The content is normally entered in certain environment, i.e. im starting to write a wiki page or a forum thread, but why it should be only that? Could be complex for the programs to do that now or soon? ok, but that the data structures be able to have that kind of flexibility opens a lot of future possibilities.

### Containers

With that schema, what would be the containers? For some things is clear to be seen, like articles, or blogs, or forums that are actually containers. But what about wiki pages, if the content is in fact a wiki page or at least most of what is it? We have already wiki pages containers on structures and categories, another container could be a namespace (if it is plain different from a category, else could coincide), or even a "wiki page" container that holds special attributes for a piece of content that makes it a wiki page. It could even be a "collage" container something that picks bits and pieces from the system and makes a

system or section homepage showing a "whats happening here" like newspaper's frontpage.

The container could have the attributes that the piece of content don't have by itself, i.e. other content pieces that it is related to in a forum thread, publication date for an article, permission for the entire container and so on.

Even could be used trackers for extending the attributes of certain container, or even to have a "generic" container, and the attributes that makes a difference be hold in trackers or something similar (check ContainersVsContent\_gmuslera) for maximum potential of expansion of software without changing the db structure.

## Applications

If the previous idea have some kind of sense, what about the programs that work with that?

They could behave a lot like the existing, current ones, picking the information from other places, or maybe with extra functionality related with what enables this other way of store information.

They could ask for all the possible information, or taking "defaults" related to what they will do, i.e. for the wiki page editors resetting the available text format options to maybe wiki and wiki with html, or asking for an image if is an article.

They even could "build" a new object type, take some content, adding some attributes for it to the right tracker, know how to show and create it, and voilà, new feature, no db changes required. That could be a good help for "pluggable" external modules

## Permissions

With such flexibility, the permission system could evolve a bit too. A piece of content could have permissions attached, but also the containers. And to see if i can do some action over a container or a content, i could check in all the related pieces.

Also, if object types could be created "on the fly", the permission system and object types tracking will have changes. In some way i should have registered what kinds of objects have, and what kind of actions can do with them so i could give object+action permissions.

## End words

I'm not sure if this kind of things could be done with Tiki in a future version, unless there is a rewrite, or even if i proposing here something very known and almost an standard on some category of CMS. But I think i'm not speaking about other software, but about Tikiwiki, something that have as strength the integration of forums, wiki pages, articles, blogs and a lot more, and this tries to be an step forward in what it already do, a way to advance in the direction of TikiIntegrationDev.

As I said in the beginning, this is centered in er... "text-box" based content, things like polls, or maps, or homework, or directories, don't match exactly with this model so they could still be like they are more or less