TikiObjects_gmuslera

In tikiwiki there are a lot of ways to store information, a lot of ways to interact and add content, to give a more programming oriented point of view, a lot of object where content is stored.

But that objects are not equal. Some of those objects are contained in others (like forum posts in forums, or wiki pages in categories) and others are completely dependant of other objects (like attached files to wiki pages, can´t be seen without watching the wiki page where they belong to). So lets play a bit over that difference.

Objects

There are a lot of tiki objects, and the system is aware of them. There are even unified ways to see some information about them, in example in the users_object_permissions table, where and object kind (i.e. ´wiki page´) and an object ID are related to groups and what they are permitted to do with that object.

To start to categorize objects, we have 3 main kinds of objects

Containers

In this category we have image and file galleries, forums, trackers, blogs, directories, faqs, and categories. They have specific attributes for the entire containers, things that changes it behaviour, permissions that apply for it, but in some way or another, the contained objects could have meaning outside it.

Content objects

Here is where most content is stored. There is a few content in the container itself, more than a name, or a description, most of the content is in the contained objects.

There are a lot of content objects, like files, images, tracker items, blog/forum posts, articles, links, etc. Some of those objects could not have a container (i.e. maps) or could be seen as having an implicit one (i.e. the article collection).

Normally the contained objects have no separate permissions for them i.e. a particular forum post or an individual image inside a gallery.

Also, most of them have names dependant of the container, like the 2nd item of the "Phones" tracker, or the 342th post to the 1st forum.

Attached objects

While they are not usually used yet over the available objects, there are another generic category of object that are normally attached to other objects.

For now there are mainly 2 of this objects (attached files, and rankings), that are more associated content than attributes of that object, and are more related to content objects than with containers.

This attached objects have (or should have) no meaning outside the object is attached to, there is no way actually to say (or see) "the 4th attached file of the UsefulScripts wiki page" without watching that page, and the same should happen with rankings.

Exceptions

Wiki pages have some of the attributes of containers (i.e. proper names, permissions) and some of the content objects (the content itself)

For wiki pages there are 2 content objects that behaves like attachments, but also can be seen as containerless content objects that are illustrations and attached images. As wiki pages, they have an unique namespace (that could make things not so clear in a multieditor environment, the image that attach an editor to a wiki page could overwrite the image of another wiki page, if the image have the same name or anyone that can edit illustrations could edit someone else illustrations), but no individual permissions, owner, description, etc. Those exceptions make tikiwiki harder to use or understand, or give problems very hard to solve with current permission system.

What from here?

The category of attached objects open the door for a list of near future enhancements to tikiwiki.

If we have tables that are specifically for files attached or for rankings, with attributest that says what is the object id and kind like is managed with permissions, we could have this attributes for most content objects in the system with relative ease (will be a common library that manages this kind of objects, and actual scripts and templates should be aware of that capabilities, of course).

Another thing that it opens is a new collection of permissions for all content objects, related to attach files, see attached files, administer attachments and rank, see ranking, etc

And lastly, it open the need of a few new object attributes, like enabling to be ranked and enabling to have attached files.

This model of viewing objects could serve to make things more orthogonal, to round a bit more the framework for creating new content objects and to normalize a bit more the information for things like permission evaluation or creating extensions.