Philo - Bug #15

Blogs should have a ManyToManyField to BlogEntry.

08/10/2010 06:46 AM - Joseph Spiros

Status:	Closed		**Start date:**	08/10/2010
Priority:	High		**Due date:**	
Assignee:			**% Done:**	0%
Category:	Penfield		**Estimated time:**	0.00 hour
Target version:				

Description

Rather than BlogEntry having a ForeignKey to Blog, Blog should have a ManyToManyField to BlogEntry.

Pros (and Neutrals):

- Philosophically, only users with permissions to edit a Blog should be able to add entries to that blog, but anyone with permissions to do so should be able to create BlogEntry objects. This helps with per-object permissions, too.
- Allows users to create entries and keep them unpublished in a "private" or "draft" state by simply not having them added to any blogs.
- Allows entries to appear in multiple blogs. (I'm not sure that this is a particularly good idea, but it's worth noting as a side-effect.)

Cons:

- Difficult to administer the list of entries in a blog, due to limitations in the admin, though raw_id_fields can help.
- Changes schema. If this change is made, it should be made sooner than later to avoid causing problems for more deployments down the line.

History

#1 - 08/17/2010 06:55 PM - Stephen Burrows

This seems like a strange idea to me. The main idea seems to be making permissioning easier, since the second pro could easily be achieved by having a "draft" field on the blogentry or making the foreignkey accept a null value (which was done in commit bb938c5fa1db5fe9133c12385526b99fce343f87). The first pro also seems unnecessary, since we could simply limit the foreignkey's choices to blogs the user is allowed to edit.

Also, philosophically speaking, we seem to be coming at this from the idea of a normal user accessing the django admin, which really shouldn't be happening. The django admin is not supposed to be used to manage things in an extremely user-friendly way - it's supposed to be nitty-gritty.

#2 - 01/06/2011 10:07 AM - Joseph Spiros

- Status changed from Feedback to Closed

I'm closing this for now, though we may want to reassess in the future.