# Comparing Social Networking Modules for the drupal.org Redesign.

by fago alias Wolfgang Ziegler
Vienna, 02.03.09

## Comparing friendship modules

### *Requirements*

This module should allow d.o. users following other users.

- One-way friend connections
- Ability to block following users
- Views integration

### *Modules*

First off I'm selecting modules that will be compared lateron.

- Ajaxified Friends: No release yet.
- Buddylist: No drupal-6 version.
- Buddylist2: No ready drupal 6 version yet.
- Flag: One can use user-flags for doing one-way relationships between users.
- Flag Friend: Builds 2-way relationships upon the flag module, but it doesn't seem to handle one-way relationships.
- Friend: Dead due to unpatched security hole.
- Friendlist
- User Relationships

So for one-way friend connections possible choices are the flag, friendlist and user relationship modules. All of those modules offer views integration, but none seems to have the ability to block users. However all of the modules provide hooks, which makes it possible to build that functionality on top of them.

It follows a comparision table for the remaining three modules. The criteras are evaluated with ++ (very good), + (good), o (average), - (bad), -- (very bad). The number in the square brackets „[]“ reference to a note below.

| | Flag | Friendlist | User Relationship |
|---|---|---|---|
| Popularity | ++ (1,476 users) | O (440 users) | + (944 users) |
| Project maintenance | ++ [1] | + [2] | + [3] |
| Drupal compliance | + [4] | - [5] | + [6] |
| Code quality | ++ | + | + |
| Test drive | ++ | - [7] | + [8] |

Notes:

1. There is high activity in the issue queue and the maintainers care. Bugs are fixed and new patches are discussed and reviewed! As the flag module is well established one can expect the  project to stay well maintained in the future.
2. Friendlist is developed by multiple developers and gets a lot of bug fixes.
3. The maintainer looks at the issues and the bugs get fixed. While there were some troubles with the maintainer, it now has a new one.
4. Coder lists some code style issues, but actually the code looks really drupalish. It also integrates really well.
5. Friendlist doesn't follow the guidelines for CVS commit messages and coder also reports quite some issues. The admin ui is not very drupalish and hard to grasp.
6. Coder rates user relationship quite well, also the admin ui and module integration is alright. The code comments of the API are fine, the rest could be improved and more drupalish, e.g. commenting hook implementations.
7. The module generates a lot of PHP notices and also some warnings caused by bad queries. The module seems to work though.
8. The module generates quite some PHP notices, but is working well.

## What were the evaluation critera?

- Popularity: The (latest) drupal 6.x usage statistics from January 25th.
- Project maintenance: Does the module receive bug fixes? Activity in the issue queue.
- Drupal compliance: I tested code style compliance by using the coder module. I also had a look whether the project and module seems to behave „drupalish“ and integrates well into drupal.
- Code quality: I had a short look at the code structure, the APIs, code comments and how well the module is modularized itself.
- Test drive: I've configured a simple one way connection and tried to establish the connection between two users for each module.

### Building the „block user“ functionality

A way to build that would be:

- Create two one-way-relation ships, one for following, one for blocking.
- Once a user blocks another one, delete the „following“ relation between them.
- Once a user follows another one, make sure he hasn't been blocked. If he has been blocked, just revert the created relationship and show the user a suiting message.

I've built a simple prototype using the flag module and two simple rules – which works fine. Look for the prototype directory, if you are interested in that. Of course the rules could be replaced by a simple custom module.

### Results

I try to sum up my results here and try to give a suggestion.

- Friendlist seems to be an active project, which receives quite some development. There are quite some extensions available and it integrates with a lot of modules. However it's doesn't seem to be very solid and it has a small user-base.
- User relationship seems to be the most popular two-way-connection module out there. As friendlist it comes with a lot extensions and module integrations.
- The flag module is the successor of the views bookmarks module, it's popular, well developed and maintained. In contrast to the other modules there are not so many extensions available, but they aren't needed for d.o anyway. However the flag friend module shows how easy the flag module can be extended and also makes a good impression.
  So the flag module appears to be simple, but powerful. It seems to be a reasonable choice for creating one-way user connections.

## Comparing activity log modules

### *Requirements*

This module should provide the „Updates“ as well as the „Friend updates“ blocks for the dashboard.

- Log an extensible list of activites
- Views integration for listing

Features, which would be nice to have:

- Different display styles per message
- Categorizing messages and allow users to filter by category
- Ability to translate messages

## *Modules*

First off I'm selecting modules that will be compared lateron. For a overview of the available modules see also http://groups.drupal.org/node/15207.

- Activity: There is no 6.x-1.0 release out yet and views integration is still worked on. However it still offers blocks on its own.
- Activity Log: Glue module beween views and rules.
- Activity Stream: 3rd party integration only
- Heartbeat: Builds upon rules.
- Lifestream: No releases.
- User activity: Measures how active the user is, but provides no update logs.

So there are the activity, activity log and heartbeat modules left.

It follows a comparision table for the remaining three modules. The criteras are evaluated with ++ (very good), + (good), o (average), - (bad), -- (very bad). The number in the square brackets „[]“ reference to a note below.

| | **Activity** | **Activity log** | **Heartbeat** |
|---|---|---|---|
| Popularity | O (287 users) | - (54 users) | -(47 users) |
| Project maintenance | + [1] | --[2] | + [3] |
| Drupal compliance | + [4] | -- [5] | O [6] |
| Code quality | ++ | + [7] | + [8] |
| Test drive | + [9] | + [9] | + [9] |

Notes:

1. There is quite some activity in the issue queue and maintainers take part, however issues could get sorted out faster.
2. The project looks dead. There was no activity since October 08 and the module seems to be unfinished.
3. There is some, but not much activity in the queue. The maintainer cares and fixes bugs.
4. Everything is fine, coder reports just a few issues, but a README is missing.
5. Coder reports quite some issues, including a criticial one. Docs are missing and the UI is difficult to grasp at it is.
6. Heartbeat doesn't follow the guidelines for CVS commit messages and coder also reports quite some issues. The admin ui integrates fine and provides some simple help, but a guide for setting the module up is missing.
7. While the code looks overall fine, it has unfinished parts as the views integration.
8. Code looks overall fine and is structured in submodules. However the submodules violate the module's namespace.

9. The module generates quite some PHP notices, but the basic functionality seems to be working fine.

## What were the evaluation critera?

- Popularity: The (latest) drupal 6.x usage statistics from January 25th.
- Project maintenance: Does the module receive bug fixes? Activity in the issue queue.
- Drupal compliance: I tested code style compliance by using the coder module. I also had a look whether the project and module seems to behave „drupalish“ and integrates well into drupal.
- Code quality: I had a short look at the code structure, the APIs, code comments and how well the module is modularized itself
- Test drive: I've insallted the module, configured it to log some simple activites and verified it's basically working.

## Features

- *Different display styles per message*
  No module seems to take care of that. The activity module comes close to it – it has two display styles: One for the „acitivity creator“ and one for the rest of the users. However for d.o. there should be two display styles available at the same time for all users, so one could use one style for the generic updates block and one for the friend updates block.
- *Categorizing messages and allow users to filter by category*
  No module provides that feature except for heartbeat, which has three predefined message types. However one cannot define custom message categories, which users can use for filtering.
- *Ability to translate messages*
  Only heartbeat provides that feature, but it passes configurable strings to t() instead of using tt() of the i18n module.

## Activities of multiple sites

As d.o. is going to be built of multiple sites we need a way to log activities of all sites on a single one. No module provides support for that, but each of it could be extended to do so by their API (activity module) or the rules API (the others).

I'm going to start development of web service support for rules, both client and server side[1]. This could be also used for logging „remote activites“. However I don't think it will be ready early enough, so it's not a great help here.

[1] http://more.zites.net/bringing_rules_to_the_next_level

## Results

I try to sum up my results here and try to give a suggestion. However it's difficult to do so as all modules have their drawbacks and none of them can provide all needed features out of the box.

- Activity log looks to be dead and not more than a prototype.
- Heartbeat has quite a small user base and setting up further custom messages is complicated and not well documented. However it offers quite some complex features, which no other module provides. As activity log it works on top of the rules module, so it relies on an existing API, but needs the rules module to work.
- Activity module makes a good impression, however it's not ready for drupal 6 yet, in particular the views integration is still missing. But there is already a patch[2] for that. The idea of having different display styles for one message is great, but it's not implemented in a way it could serve d.o.'s need for generating both update blocks without changing all the activity integration modules.

I also plan to add a generic logging module to rules, that probably allows logging messages for basically any entity – as there was already such a module for drupal 5 (workflow-ng's per entity logging module). By logging messages for users this can also serve as an „activity log“ solution. As of now I plan to support different messages types and arbitrary categories, as it's already the case for 5.x. Furthermore I think of adding translation support, but I don't plan to add in more complex features like detecting multiple messages. I'd prefer to see such things implemented as further add-ons.

So if there is still enough time to wait for that, the „rules logging module“ could be also a way to go. I think I could have it ready in about a month and as I've not started with it yet, I could ensure it fullfills the requirements for d.o.

In the end it comes down to using a rules-based solution or the activity module. As I'm the author of rules I'm of course in favour of the idea of building upon rules. However the existing rules-based solutions are not ideal yet.

[2] http://drupal.org/node/335498