SUMO Upstream Process

Proposed conference call

What time works for you all? 11 AM ET / 5 PM Europe? Wed May 20 2009

Useful links

SUMO bug tracker (choose support.mozilla.com as the product):
https://bugzilla.mozilla.org/query.cgi?format=advanced

Mozilla SUMO svn: http://viewvc.svn.mozilla.org/vc/projects/sumo/

SUMODEV weekly meeting notes: https://wiki.mozilla.org/Support/SUMOdev_Meeting_Notepad

irc://irc.mozilla.org/sumodev

Architecture Suggestions From Mozilla

Who's who

David Tenser: Firefox Support Manager, a.k.a. the guy responsible for the Sumo project. He wants to make sure SUMO/TikiWiki is generalized enough to be used in other organizations other than SUMO, and wants to make sure lack of upstreaming does not become a problem.

Marc Laporte: TikiWiki admin. Super excited about the SUMO project but sees the danger of not enough upstreaming/synchronizing between the SUMODev and Tiki worlds. Wants to correct this and improve co-operation between the two communities.

Nelson Ko: TikiWiki dev/admin who was the original person who did all the early coding to make Sumo happen, including staging and approval. Nelson admits he subsequently becoming too busy with other TikiWiki projects so did not spend enough time on upstreaming 😕 But he is trying to make amends by getting Martin and Sylvie more involved, and is the creator of this page!

Eric: Originally hired by Nelson to take over his responsibilities for SUMODev as Nelson got busy with other TikiWiki projects, Eric is now a contractor for Mozilla for SUMODev. Eric is a kick-ass developer, and also has his own project http://knowmore.org

Laura: Lead developer of SUMO. Laura works for the Webdev division of Mozilla which takes care of all their websites. Other than SUMO, Laura is very involved with add-ons.mozilla.org, which Mozilla wrote themselves in the CakePHP framework.

Martin Cleaver: Recently started for Nelson on various TikiWiki projects. Has been assigned by Nelson to work on making upstreaming to TikiWiki easier not just for SUMO but all projects that Nelson is involved in, in addition to improving things like Testing, Code Documentation, Interfaces/Hooks for Extending Tiki.

Sylvie GrÃ©verend: TikiWiki developer/admin from the early days of the project. Was contracted by Nelson to help out in upstreaming stuff to TikiWiki. So she knows the difficulty of upstreaming after the fact, as opposed to developing directly in Tiki trunk.

Other Mozilla people heavily involved in SUMODev: Paul, a former QA intern, now contractor developer with Mozilla; Stephen, the QA guy; Cheng and Matthew who heads up live chat and forums but do a bit of coding now an then and a lot of the sysadmin type things. Chris is the editor of the KB so does admin config of Tiki (and reports bugs 😃)

David's initial questions

David's initial questions were:

- How can we make sure our patches are not missed by the TikiWiki community?
- Is there a simple and straightforward way to submit our patches, or point to them, somewhere where you're looking?
- How does your process of accepting patches look like?
- Do we need someone from the TikiWiki side that can work together with us in making sure relevant fixes are upstreamed?
- Does using our patches depend on us upgrading to the latest TikiWiki on SUMO?

Marc's recommendations and Nelson's comments

1- Code directly on trunk
a) So we know the feature is shared, and will be waiting for us when we upgrade in the future.
b) The community knows early about the feature and there is no feature duplication which we need to solve later.
c) If I can run the project on trunk, I do so. There are more bugs but it's the way to keep trunk stable. If not, I backport to to the stable branch.

Nelson: I fully agree with the benefit of coding directly on trunk. Ideally, SUMODev could institute a practice to code on Tiki trunk first, and then "backport" to SUMO, rather than the reverse.
Sylvie: But coding in trunk needs sometimes a lot more effort to generalize. For instance the multiple polls feature in sumo can not be backported as it in tikiwiki because of too many hardcoded settings.

2 - In the case where I have a project that is quite different, I would manage it through an experimental branch in the TikiWiki SVN. Is there a reason why SUMO can't be on an experimental branch of TikiWiki SVN?

The Council of Europe is in a similar situation than SUMO.
1- They want to upstream as much as possible
2- They have high stability needs, because they deploy dozens of TikiWikis.
3- They have some things which are not shareable (tNews)

COE has an experimental branch:

http://tikiwiki.svn.sourceforge.net/viewvc/tikiwiki/branches/experimental/coe/
A benefit is that the larger TikiWiki community receives the commit notifications.

I am very happy when active TW community members are asked to
perform some Tiki code for SUMO and I want more& more of that. However,is
there a reason why SUMO devs (Nelson, Laura, Eric, etc.) don't commit directly
to trunk?

Nelson: I think there might be practical reasons why SUMO might have to be on Mozilla SVN.... Need more discussion on this point.

3 -Again, I am not familiar with specifics but here is what could be
a proposal:

1- Upgrade SUMO to branches/3.0

Nelson: I think doing this is a good first step and IMO is very important.

2- Coordinate with TW quality team to see which fixes should go to soon-to-be-stable branches/3.0 for inclusion in 3.1

Nelson: yes, fully agree.

3- The general-purpose stuff, but not suitable for the current stable branch, committed to TRUNK and will be part of the official 4.0

Nelson: yes, fully agree. The question is only if sumo devs "have time" to do this themselves, or to code on trunk first and then backport to SUMO.

4- SUMO experimental branch: for all the stuff which is very specific

We already have some scripts to merge. Maybe some more scripts will be needed or the current ones deserve a tweak. In any case, COE is in a similar situation to SUMO.

Nelson: I think this makes sense in concept, but there is already SUMO in Mozilla svn, so that will cause duplication. This needs to be resolved. It makes no sense to duplicate. Can we somehow channel Mozilla SUMO svn commit notices to our Tiki svn list?

Nelsons suggestion to use IRC bots as a way to know what's going on?

Bugzilla changes appear in irc.mozilla.org#sumodev through firebot, and also Tiki svn commits appear in irc.freenode.net#tikiwiki through CIA, so maybe if we make use of such IRC bots, it might be a very good way to keep track of what's happening across the "Tiki-SUMO border" which I hope will become less a border 😐

Sylvie's initial answers to David's questions and Nelson's comments:

- How can we make sure our patches are not missed by the TikiWiki community?

- is it possible to send all the commits on a mailing list. It
will be so easier to follow for us.

Nelson: maybe IRCbot is better? Time to get creative. Mailing list may not help I think because SUMO devs already are flooded by bugmail from bugzilla. Or maybe it can work if they use intelligent (like gmail type) filtering like I know some do?
Sylvie: Easy to put a mainling list in a specific folder. You can search in a mailing list. It the irc has a log, yes it will be the same. The need is to be able to search on variable, commit comments, file names- and to have a chronological list. It will be great too that the commit comment is not only bug xxx but also a little summary.

- Is there a simple and straightforward way to submit our patches, or point to them, somewhere where you're looking?

- if you guys want to be developers in the tw team - no problem. You can
propagate yourselves the fixes.

Nelson: This is ideal if SUMO devs can commit directly to Tiki trunk.

- How does your process of accepting patches look like?

We accept every patches - and if it does not look correct - we rollback.

Nelson: The approach in SUMO (or Mozilla in general) is to post patch files on Bugzilla, which are then reviewed then committed. In contrast, the way it's done in Tiki is to commit first and then review within SVN. Hence, the center of activity in SUMO is bugzilla, not SVN. In contrast, the center of activity in Tiki is SVN, not so much the Tiki bug tracker. This is a cultural difference that devs may not be used to...

Do we need someone from the TikiWiki side that can work together with us in making sure relevant fixes are upstreamed?

Not sure somebody has free time to do that on the tw side.... We can look for a volunteer?

Nelson: Possibly.

What has been already tranfered today (05/21/09)

- sefurl in ruules
- sefurl can be localized (lang=en transform into /en/)
- poll: poll date - optimisation
- onthe way: multiple polls for one object
- more post types on forum
- more filter on forum posts
- some improvments in category cache

Conclusions of the conference call

- Everyone agrees on overall goals of getting code bases as aligned as possible and to reduce feature/code duplication/etc.
- Council of Europe & SUMO both explained their internal procedures: they maintain a small list of files that they overwrite for each of their ~60-70 sites (15 on trunk, 35 on stable). Their shareable code changes get sent to tiki trunk. They are presently on 2.x and will move to 3.x shortly.
- SUMO has load (Firefox has over 250 million users, SUMO has 3 million unique visitors per week, and about 10 million page views) and infrastructure (caching, load balanced, master/slave mysql) like no other Tiki site, and things need to be tested more (something may work well elsewhere but not there)
- SUMO releases from dev -> production every 1-3 weeks.
- Quality Team was introduced
- SUMO will upgrade from Tiki 1.x to Tiki 3.x (aiming for Q3, but depends on human resources), and create a list of patches
- All patches/fixes that SUMO deems shareable will go to Quality Team/proposed branch process or to trunk (and thus 4.0). See RoadMap. Possible outcomes for proposed patches in 3.x :
  - Accepted outright
  - Not suitable for branches/3.0 but great for trunk (and thus 4.0).
  - Not suitable, in current state, for general inclusion in Tiki. Typically, more discussion needed on how to make this more generic. Or it could duplicate an existing feature or be out-of-sync with Tiki code (ex.: a patch from 6 months ago).
- Some effort should be done (we didn't get into much details of specifics) to align SUMO Roadmap, Tiki Roadmap, etc.
- David asked a question about a point of contact. In general, discussions should be on tikiwiki-devel@lists.sourceforge.net If this is too slow or not suitable in a particular case, Marc Laporte volunteered to be point of contact.
- SUMO will continue on own svn but will create some way for tiki devs to see what happens will be figured out.
  - (e.g. a hook to send commits to Tiki commit mailing list, or a hook to automatically send commits into a tikiwiki SVN branch)
- By following all this, upgrading from 3 to 4 will be fairly easy for SUMO.

For completeness, I added above some info/links that what not specifically covered in conference call, but relevant.