MediawikiMigration

Rationale

The documentation on the Redmine wiki has lot of duplication of information.

The solution that has been choosen for that is the following:

- Migrate part of the information in Wikidata.
- Use template and/or generate information from Wikidata.

Other solutions were also possible such as migrating to documentation system like pandoc, but doing that would increase a lot the required skills of potential contributors.

Using complicated documentation systems has several issues:

- Practically speaking, it makes it impossible for many people that don't know how to program, to participate in Replicant, fix issues etc. This would be very problematic for diversity and inclusiveness of people as it would unnecessarily discriminate against people without such skills. We could also potentially loose important contributions.
- While it would make the job of contributors way easier than without any templating or ways to programmatically generate documentation, it also increase the dependency on people who knows how to use that documentation system.

Instead it would be better to use a documentation system that enable people without programming skills to easily contribute, while at the same time enabling people with programming skills to take advantage of it as well. Templates in various wikis system like Redmine or Mediawiki enables that.

In additions to wiki systems with templates, enabling to interface the documentation system with Wikidata also has many advantages:

- It enables to reuse the information across different projects with similar goals (libreplanet wiki, Parabola wiki), different goals (for instance we could share the work of documenting hardware with the wikipedia and wikidata community), or through custom made tools.
- It can isolate the tasks requiring some programming to the strict minimum: Using programming in documentation systems can makes it easy to generate huge quantity of information, and Wikidata makes it possible to contribute to the information itself without knowing how to program. The programming is then potentially only required to fetch and show / format the information that comes from Wikidata.

Redmine wiki

Issues:

- The table syntax of Redmine textile format is too complicated for several key contributors like dllud.
- It's probably hard to interface with Wikidata
- Javascript is required for the preview to work. GNUtoo has huge issues with that as it leads him to many bad editions that are fixed in subsequent editions that are fixed in subsequent editions...
- The syntax is less well known and the documentation is available on Redmine website but harder to find.
- No one seem to know the syntax for the templating system nor its limits yet.

Mediawiki

Advantages:

- Can be interfaced with Wikidata in various ways that are used in production on Wikipedia.
- The syntax is easy enough to use by people that don't know how to program, many people are used to it, and at the same time it's well documented.
- The table format is much easier to use, and it's usable by dllud.
- The main functionality, including the preview work completely without JavaScript, which leads to an increase of edits quality by people that don't run JavaScript from remote websites.
- Other projects like Libreplanet, and Parabola uses mediawiki, so we can probably reuse things across different wikis.
- Can be used offline thanks to projects like Kiwix
- Can be more easily backuped by external projects like the archive team
- We can probably reuse many templates from other wikis with compatible licensing, and some Replicant contributors like GNUtoo already know a bit the template language.

- There are probably many more tools compatible with mediawiki than the Redmine wiki.

Issues:

- We need to migrate, if possible in a way that preserves history
- We loose the integration with redmine #<bug number> and will have to address this during the migration

Decision

At several conferences, including the Replicant conference in Paris in Summer 2019, and the FOSDEM, people were in favor of migrating to Mediawiki and didn't have objections to it.

Migration

The Replicant Project's Redminewiki uses Textile markup language, while MediaWiki uses the Wikitext markup language.

It looks like some Redmine developers have recently been working to make it easier to transition from Textile to both generic Markdown and a more standardized flavor of Markdown called CommonMark: https://www.redmine.org/issues/32424
https://www.redmine.org/plugins/redmine_reformathttps://github.com/orchitech/redmine_reformat
https://hub.docker.com/r/orchitech/redmine-gfm

There is currently an RFC at MediaWiki about supporting CommonMark in MediaWiki natively:
https://www.mediawiki.org/wiki/Requests_for_comment/Markdown

The comments section of MediaWiki's RFC page on this topic may be helpful to read for context on this proposal.

If we can find a reliable fork of MediaWiki that uses CommonMark instead of Wikitext, we should at least consider using it for the reasons outlined in the MediaWiki RFC links above and because of the fact that we now seem to have reliable software available to us to transition from Textile to CommonMark.

If we ultimately decide to use vanilla MediaWiki with Wikitext, or if we don't find any forks of Mediawiki that use CommonMark instead of Textile, it is our assumption that it would be easier to transition from the more common Markdown or CommonMark markup languages to MediaWiki's Wikitext markup language than it would be to transition from Textile directly to Wikitext. This assumption has been made based, in part, on arguments made here: https://hub.docker.com/r/orchitech/redmine-gfm

TODO:
-INSTALLATION:

- We find a way to install mediawiki somehow and we do the install **but** we don't use it officially yet
  like there would be no users accounts.
- We can then use it to test migration scripts
- When everything is ready we'd disable wiki editions in redmine and enable account creation in mediawiki and announce it
  -MIGRATION:
- Use pandoc ":https://pandoc.org/" code-base to convert from textile to mediawiki, including child/parent structure of the articles, to transfer all articles to new mediawiki installation.
- Use git-mw to address the loss of history given such conversion and to recreated past contributors and editors accounts, so we make sure that smaller contributors don't get forgotten (look into "ActiveResource" has a simple way to solve the history issue, in order to get timestamps, use https://github.com/CyberTech/WikiText_to_RedMine_Migration as reference)
  (WIP)
  references:
  https://docs.bitnami.com/installer/how-to/configure-advanced-integration-git-redmine/
  https://docs.bitnami.com/installer/apps/redmine/configuration/use-git/
  https://www.redmine.org/projects/redmine/wiki/HowTo_configure_Redmine_for_advanced_git_integration
  look for inspiration in parent-child https://github.com/eugene-sy/redmine_wiki_hierarchical_export
- use py package to acess rest API
  (WIP)
  (1) it's unfinished and has not been tested recently
  (2) he might need to put a bit more work on it before being able to publish them,
  for instance with my work on Guix, I had to remove some sensitive information like SSH keys
  There are also other ways to configure machines than Ansible, Parabola has packages for its configurations for instance
  So it probably depends on the distribution too
  And both approaches (packages and Ansible) aren't incompatible between each other
  (/WIP)
  -Usability
- use sphynx like skin for mediawiki (https://supertuxkart.net/Maintaining_the_Wiki good information refernce on skining)
  -ARCHIVE:

- Contact the Archive team to backup the redmine's wiki
- Make sure it gets archived completely on archive.org/web
- Make sure it gets archived in files that can be restored (but without non public information like password hashes, etc) -SOFTWARE NEEDED:
- https://pypi.org/project/python-redmine/
- https://github.com/schacon/grack
- look into this for inspiration https://github.com/vile/redmine2confluence-wiki
- for usage parameters and variables use https://github.com/likema/redmine-exporter has reference
- this redmine plugin might be usefull Wiki Hierarchical Export Plugin
- inspiration http://stbuehler.de/blog/article/2011/06/04/exporting_redmine_wiki_pages.html
- inspiration https://gist.github.com/tim-jansen/6263586