

Rapport du projet de spécialité

Passerelle Git-MediaWiki, environnement de test



Simon Cathébras, Julien Khayat, Simon Perrat, Charles Roussel, Guillaume Sasdy

Le 11 juin 2012

1 Travail réalisé

Le travail réalisé durant ce projet est un environnement de test pour la passerelle *Git-MediaWiki*, un projet développé depuis plusieurs années à l'Ensimag.

1.1 La passerelle *Git-MediaWiki*

Ce projet a pour but de permettre de gérer un MediaWiki à l'aide de Git. Pour mieux comprendre ce projet, nous allons procéder à une explication de ce que sont Git et MediaWiki.

1.1.1 MediaWiki : des sites communautaires à contribution libre

La meilleure manière de décrire ce qu'est un mediawiki est de prendre un exemple : Wikipédia. Wikipédia est le plus connu des mediawiki, et présente toutes les caractéristiques d'un médiawiki :

- Chacun peut y contribuer de manière libre
- Possibilité d'ajouter / supprimer / éditer des pages

Le principal défaut de ces sites est que leur édition peut s'avérer fastidieuse, on note quelques problématiques à soulever :

- problème avec les éditions concurrentes
- la moindre petite modification de pages engendre une entrée dans l'historique.
- les pages s'éditent une à la fois, donc si on a une modification récurrente sur plusieurs pages, on se retrouve avec un historique compliqué.
- de même une mise à jour importante (ex : ajout de plusieurs pages sur un projet pour l'Ensiwiki) on doit ajouter les pages une à une.

- lors d’une mise à jour importante nécessitant un long travail sur une page, on se retrouve face à un dilemme :
 - soit on enregistre régulièrement et on produit un historique ”sale”
 - soit on enregistre qu’une fois le travail terminé et on s’expose au risque de perte des heures de travail en cas de crash de l’ordinateur, ou de fermeture du navigateur par erreur.

Le but de la passerelle est d’apporter une solution simple à ces problèmes en mettant à jour les mediawiki à l’aide d’un gestionnaire de version : *Git*

1.1.2 Git : un gestionnaire de version

Un gestionnaire de version permet de partager une série de fichiers sur un serveur et de permettre à plusieurs personnes de les éditer en concurrence. En gérant les pages MediaWiki via *Git* on permet à l’utilisateur de résoudre la plupart des problèmes liés aux mediawiki (cf ci dessus). L’idée est d’obtenir une copie locale des pages du wiki pour bénéficier des avantages de ce mode de stockage.

- les fichiers en local permettent d’éditer un fichier en plusieurs fois et de faire une seule mise en ligne. Un historique plus propre et pas de risque lors de l’édition des pages
- On peut facilement faire une grosse mise à jour de plusieurs pages sur le wiki sans problèmes.
- aucun risque de perte de donnée en cours d’édition, la plupart des éditeurs de texte ont un système de sauvegarde automatique.

1.2 L’environnement de tests

Le projet *Git-MediaWiki* est actuellement dans le répertoire *contrib/* de la branche master de Git. Pour pouvoir aller plus loin, c’est à dire se retrouver nativement intégré à Git, la passerelle doit se pourvoir d’une base de tests beaucoup plus approfondie que l’actuelle. Jusqu’à aujourd’hui, les tests de *Git-MediaWiki* étaient effectués à la main en vérifiant le contenu des pages Wiki et des fichiers dans le dépôt Git.

L’objectif de notre groupe est de construire un environnement complet de tests pour la passerelle, avec pour objectif, à terme, de permettre l’avancée de *Git-MediaWiki* hors du répertoire contrib. Le principe est d’avoir toute une batterie de tests s’exécutant automatiquement et validant les fonctionnalités déjà développées, ou indiquant que des fonctionnalités ne le sont toujours pas.

2 La communauté *Git*

2.1 Organisation

2.2 Interactions avec la communauté

3 Difficultés pendant le projet

3.1 Premier Run : Lundi 21 Mai → Mardi 29 Mai

Le premier run de ce projet de spécialité a comporté un nombre conséquent de problèmes, tant au niveau organisationnel que technique

3.1.1 problèmes techniques

Durant le premier run, nous avons eu des difficultés à appréhender le projet. Nous avons dû procéder à beaucoup de changements de structure pour notre code

1. plusieurs fichiers *bash* encapsulant du *Perl*
2. continuer les étapes

3.1.2 problèmes organisationnels

Après une analyse à la fin du premier run, nous avons déterminé plusieurs problèmes d'organisation :

- manque de communication dans le groupe
- mauvaise compréhension de certains points de la charte d'équipe par une partie du groupe.
- plusieurs membres du groupe ont dû s'absenter un après midi ou une journée pour des raisons personnelles
- pas d'évènement de cohésion de groupe

Tout ces points conduisant à un premiers run assez désorganisé, produisant des difficultés à appréhender le projet sur un plan général.

3.2 Deuxième Run : Mardi 29 Mai → Mardi 5 Juin

Pour le second run, on rencontre beaucoup moins de difficultés. Quelques problèmes techniques ont parfois ralenti la progression du projet, mais jamais de manière drastique. Notre plus gros problème a été d'avoir sous-estimé le temps que prendrais d'apporter les correctifs demandés par la mailing list.

Le travail prévu à cette fois été effectué à temps et correctement.

3.3 Troisième Run : Mardi 5 juin → Mercredi 13 juin

4 Bilan

Au niveau de l'équipe

Au niveau des enseignants

L'ensemble de l'équipe s'accorde à dire que l'enseignant de référence (Matthieu Moy) a été d'une qualité exemplaire tout au long du projet. Disponible, réactif sur les questions par mail et présent dans la salle. Il aura été d'une grande aide grâce à sa connaissance des l'environnement du projet, c'est à dire la communauté Git et ses coutumes, le logiciel libre dans un sens plus général, mais également dans sa capacité à nous orienter dans la bonne direction en ce qui concerne l'architecture des fichiers.

Un autre point très positif du projet aura été l'application des méthodes agiles, qui ont permis à l'équipe de travailler plus sereinement. Les passages régulier de XXXXXX nous ont incité à réfléchir à notre organisation, augmentant notre efficacité au cours du projet.

Améliorations

Après discussion dans l'équipe voici quelques pistes d'amélioration pour le projet dans les années à venir :

- donner des informations “technique” sur le projet sur la page de présentation. Il aurait put être intéressant de connaitre les langages utilisés à l'avance.

—