

11/25/2008

M2-STIC

GOBBY

Rapport pour l'exposé de collecticiels

Valat Sébastien

Table des matières

Contents

1.	Généralités	3
a)	Le contexte.....	3
b)	Le principe.....	3
2.	L'interface	3
3.	Le fonctionnement.....	5
a)	Description générale	5
b)	Modification et version des documents	6
c)	Un exemple.....	6
4.	Gobby par rapport aux modèles collaboratifs.....	7
a)	Application au modèle en trèfle	7
b)	Le modèle de participation	8
5.	Quelques points sur son utilisation	8
a)	Les avantages	8
b)	Des points à améliorer	8
c)	Des améliorations peu être à venir	9
6.	Avis personnel	9

1. Généralités

a) Le contexte

Gobby est un logiciel OpenSource de production collaborative et temps réel de fichier texte. Utilisant GTK+ il est disponible pour Linux/Unix, MacOSX et Windows™. Il s'agit d'un outil simple permettant à plusieurs utilisateurs de partager des documents de sorte que chaque participant dispose à tout instant d'une version à jour.

Sa structure est orientée vers la production de code source, il est donc idéal pour le développement collaboratif ou l'enseignement informatique à distance. Son aspect simple en fait un logiciel réactif et facile à prendre en main.

On peut dire merci à l'équipe de quatre personnes bénévoles qui développent cette application. Les fichiers d'installation et sources sont téléchargeables sur le site des éditeurs : <http://gobby.0x539.de/trac>.

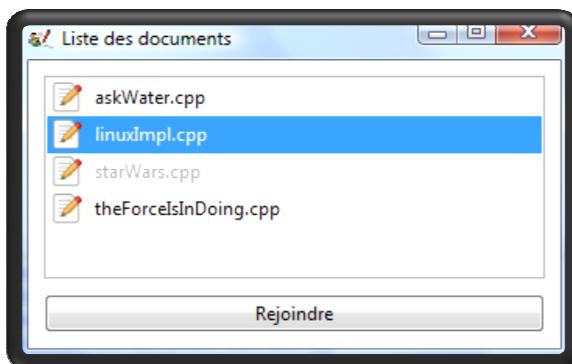
b) Le principe

Les participants se connectent à un serveur ; s'identifient par un login et choisissent une couleur qui sera utilisée pour marquer les portions de textes qu'ils modifient. Tous les participants peuvent partager des documents et accéder aux documents des autres. Un document peut être édité simultanément par plusieurs participants et chacun voit en temps réel ce que les autres écrivent.

2. L' interface

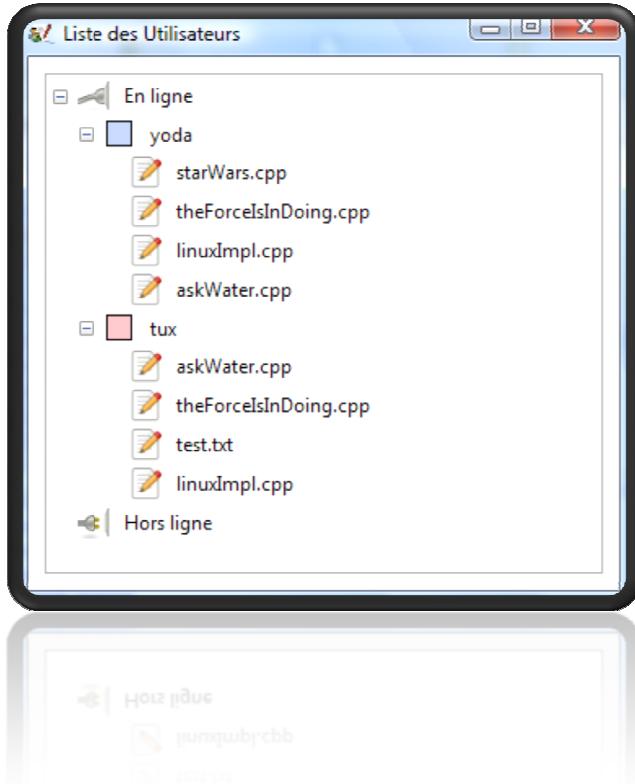
L'interface de Gobby est répartie en trois fenêtres : la fenêtre principale contenant la zone d'édition et le chat, la liste des documents et la liste des utilisateurs. On peut y distinguer les zones actives suivantes :

- *La liste de documents* : elle est fournie comme une fenêtre à part et liste tous les documents partagés sur le serveur. Un double clique sur les noms permet d'ouvrir le

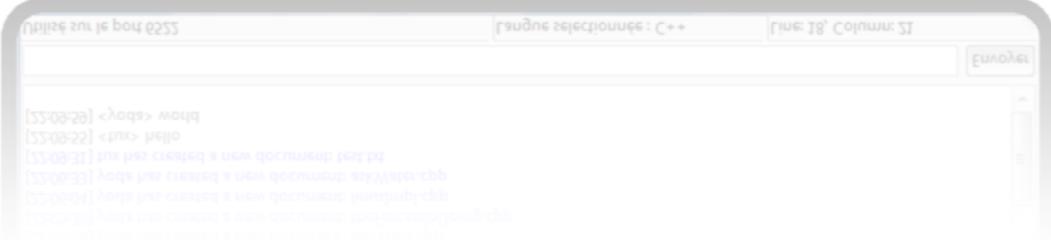
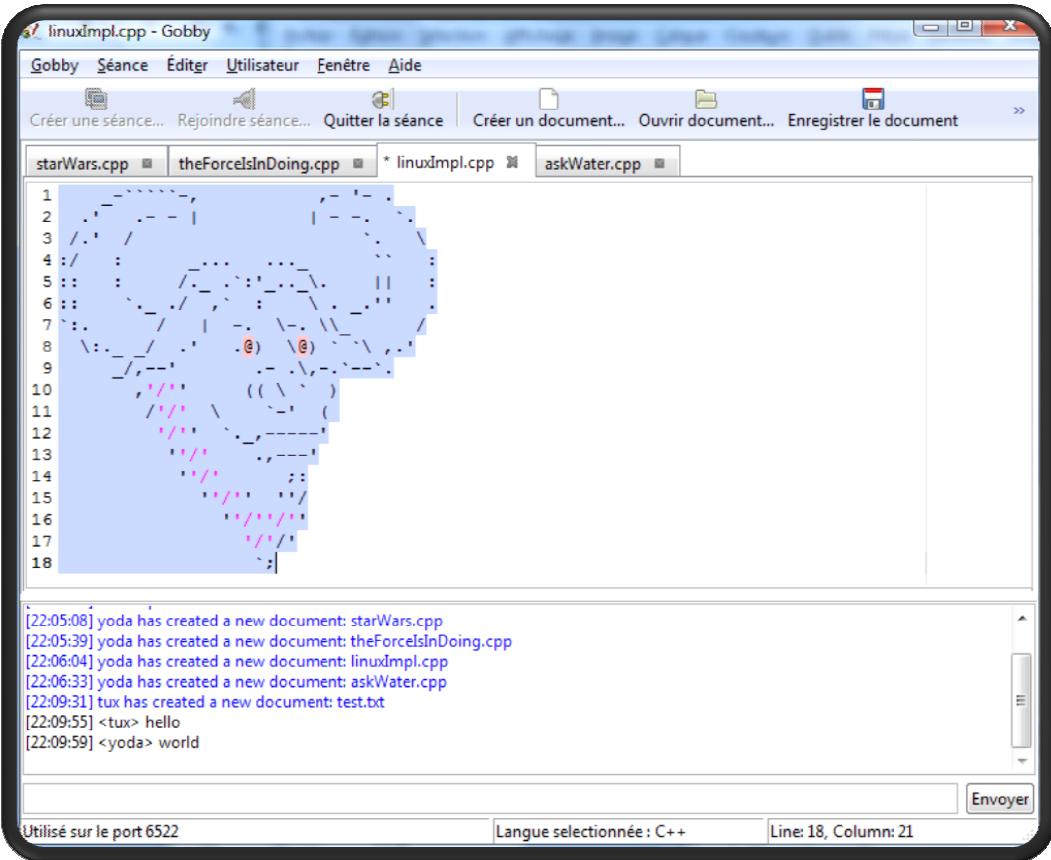


document en édition. Les documents en cours d'édition sont marqués en noir tandis que les autres sont marqués en gris.

- *La liste des utilisateurs* : également fournie comme une fenêtre à part cette zone permet d'avoir connaissance de l'ensemble des utilisateurs présents sur le serveur, des documents qu'ils sont en train d'éditer et de la couleur qui leur est associée.



- *La liste des documents ouverts* : cette liste est placée sous le menu de la fenêtre principale, elle prend la forme d'une liste d'onglet portant le nom des fichiers ouverts. Si le fichier a été modifié depuis son dernier enregistrement le nom est précédé d'une étoile.
- *Le document en cours* : L'espace centrale est dédié au document en cours d'édition. Les caractères du texte qu'il contient sont surlignés en blanc s'il s'agit du texte original ou avec la couleur de l'utilisateur les ayant modifiés. Le logiciel étant orienté développement la partie gauche de la zone fournit un affichage des numéros de lignes.
- *Le chat* : la partie basse de la fenêtre principale fournit aux utilisateurs un moyen de communiquer via un chat ultra simple.



3. Le fonctionnement

a) Description générale

Gobby fonctionne en mode client/serveur, un des participants doit créer une session sur laquelle les autres vont pouvoir se connecter. La session peut éventuellement être verrouillée par un mot de passe commun à tous.

Tous les documents partagés sont enregistrés par le serveur, chaque participant garde également une copie complète des documents en cours d'édition. Lorsqu'un participant demande un nouveau document le serveur le lui transmet et note que le client utilise ce document pour pouvoir l'informer des modifications futures.

b) Modification et version des documents

Chaque document est associé à une révision sur le serveur et sur les clients. Les modifications de documents suivent donc le schéma suivant :

1. Les modifications apportées au document sont notifiées auprès du serveur par un message contenant la *modification* et le *numéro de version* du document dont dispose le client.
2. Le client applique la modification à sa vue et la marque comme *non synchronisée*.
3. Le serveur reçoit la notification et corrige la requête si la version du client correspond à une version plus ancienne du document.
4. La modification est appliquée sur le document du serveur et sa révision est incrémentée.
5. Le serveur envoie une notification à tous les clients en précisant la nouvelle révision du document et la description du changement.
6. Le client met à jour son document, s'il était à l'origine de la modification il marque sa modification comme *synchronisée*.

Le type de modification rencontré correspond pour l'essentiel à l'insertion de nouveaux caractères ou à la suppression de caractères existants.

c) Un exemple

Pour mieux comprendre le fonctionnement du mode de synchronisation des documents, voici un petit exemple à deux participants tiré de la documentation du protocole obby.

Nous prendrons le cas où les deux utilisateurs (client1 et client2) partagent un document contenant la ligne de texte « BBY ». Le document est en version 1 sur le serveur et sur les clients. Imaginons que le client1 ajoute un « G » en première position, l'autre ajoute un « O » toujours en première position. On décompose le dialogue comme suit :

- Envoi des requêtes par les clients
 - Client1 envoi son message « insertion de G en position 0 dans la révision 1» et marque le changement comme *non synchronisé*.
 - Client2 envoi son message « insertion de O en position 0 dans la révision 1» et marque le changement comme *non synchronisé*.
- Le serveur traite les requêtes :
 - Le premier message reçu, disons celui de client1 est traité. Comme la version correspond, le G est simplement ajouté au début de la ligne. Le document est passé en révision 2 et les clients sont notifiés par un message « insertion de G en position 0, document en révision 2 ».
 - Le deuxième message est reçu. Comme la version ne correspond pas les changements antérieurs sont appliqués pour modifier la requête qui devient « insertion de O en position 1 ». Le changement est appliqué de sorte que le

mot devient « GOBBY » et les clients sont notifiés par un message « insertion de O en position 1, document en révision 3 ».

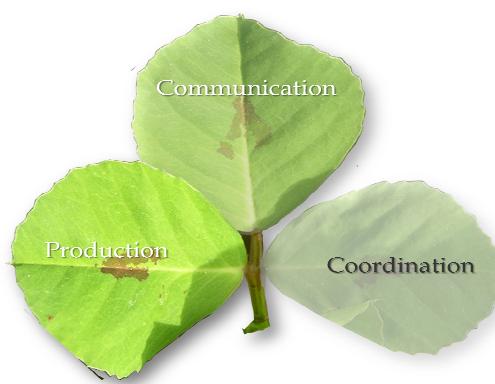
- Le client1 :
 - La première notification est reçue ; comme elle correspond à sa propre requête le changement est marqué comme synchronisé.
 - La deuxième notification est reçue ; comme le numéro de version correspond elle est tout simplement appliquée.
- Le client2 :
 - La première notification est reçue ; comme sa révision entre en conflit avec le changement non synchronisé en attente ce dernier est annulé, le changement reçu est appliqué et celui en attente est corrigé pour devenir « insertion de O en position 1 dans la révision 2 » puis appliqué à nouveau.
 - La deuxième notification est reçue, comme elle correspond au changement en attente ce dernier est simplement marqué comme synchronisé.

A la fin de l'opération les trois entités disposent d'un document contenant la chaîne « GOBBY » en révision 3. Dans le cas d'un tel conflit il est bien évident que l'ordre des deux lettres dépend de l'ordre dans lequel les paquets IP sont arrivés au serveur.

4. Gobby par rapport aux modèles collaboratifs

a) Application au modèle en trèfle

Le modèle en trèfle définit de manière très simple trois zones pouvant être couvertes par les logiciels collaboratifs : production, communication et coordination.



Gobby couvre clairement la zone de production puisqu'il vise à faire écrire des fichiers de manière collaborative. La zone de communication est faiblement abordée par son chat ultra simple.

La partie coordination quand à elle n'est pas du tout abordé par ce logiciel. Chaque utilisateur est en effet libre de ses actes sans aucune restriction, il n'y a aucune notion de rôle ni de fichier ou zone dédiées à certaines personnes. Il peut en résulter une certaines confusion si l'on désire travailler avec un nombre important de personne pour un petit nombre de fichier.

b) Le modèle de participation

Gobby ne fait pas intervenir la notion de scénario et définit de manière statique son enceinte et son unique rôle (participant) il n'est donc pas intéressant de décrire ce collecticiel dans le cadre du modèle de participation.

5. Quelques points sur son utilisation

a) Les avantages

Gobby est un logiciel relativement simple, réactif et rapide à mettre en place. En outre il fonctionne très bien même avec une connexion lente (56k....). Il offre entre autre :

- Une bonne coloration syntaxique pour un grand nombre de langages.
- La possibilité de sauvegarder et reprendre une session
- Des librairies pour manipuler son protocole dans nos propres applications :
 - Obby en C++
 - Obby4j en java (uniquement la version 0.3 du protocole).

Il est très pratique si on utilise en parallèle un logiciel de VOIP pour s'organiser, le chat manque en effet de visibilité dans l'application.

b) Des points à améliorer

La simplicité de ce logiciel ne va pas sans défauts ; son interface s'avère parfois un handicap. Voici quelques points sur lesquels j'ai parfois eu quelques accros :

- Le chat manque de visibilité, on a vite fait de ne pas y voir un message. Nos utilisateurs sont associés à une couleur, c'est étrange de ne pas l'utiliser dans le chat.
- Il est tout à fait possible d'ouvrir deux fois le même fichier, le deuxième sera affiché avec (2) à la fin du nom. Ceci pose problème puisque deux personnes risquent de travailler sur les deux versions.
- Il n'est pas possible de supprimer un fichier de la séance, c'est particulièrement gênant si l'on ouvre un fichier une seconde fois par mégarde car il reste jusqu'à la fin.
- On ne sait pas sur quel fichier l'utilisateur est en train de travailler (à moins d'être également sur ce fichier), c'est particulièrement frustrant dans certains cas avec un

logiciel de VOIP qui nous permet d'entendre les touches du clavier de l'autre alors qu'on ne trouve pas où il fait ses modifications.

- Les fichiers modifiés ne sont pas notifiés dans la liste globale des fichiers. Il faut avoir ouvert les fichiers pour savoir s'ils ont été modifiés (étoile à côté du nom de fichier sur l'onglet) et le sauvegarder si besoin.
- Le principe même du protocole rend impossible la mise en place d'une fonction annulée. C'est parfois gênant si quelqu'un supprime une portion importante d'un document, il faut donc toujours travailler avec une copie de secours (svn...) et la mettre à jour régulièrement.

c) Des améliorations peu être à venir

Une nouvelle version est en cours d'élaboration (0.5) avec un remplacement de la librairie *obby* par *libinfinity* qui fournie un protocole de remplacement basé sur XML. Les spécifications (non complètes sur le site pour l'instant) donnent déjà quelques idées intéressantes avec l'ajout de notions de groupe, dossiers...

A voir si l'utilisation d'un protocole XML ne fera pas trop perdre en efficacité réseau. Mais l'histoire est donc à suivre.

6. Avis personnel

J'ai utilisé cette application à plusieurs reprises pour travailler en collaboration à deux personnes distantes de plusieurs centaines de kilomètres. Avec l'aide de skype l'utilisation consistait à donner quelques cours de développement et développer quelques applications (dont une plus ou moins grosse).

Cette utilisation m'a pour l'instant plutôt laissé sur une bonne impression même s'il y aurait quelques améliorations à faire.

L'utilisation limité à gobby et skype se termine parfois par de l'ASCII-art très amusant dans gobby pour s'expliquer sur les algorithmes.