

Réunion des développeurs Fudaa

11 mai 2009

Compte-rendu

Participants :

- J.M. Battista, Compagnie Nationale du Rhône
- F. Deniger, Genesis Informatique
- M. Fargeix, Centre d'Études Techniques Maritimes Et Fluviales
- A. Hadoux, Genesis Informatique
- G. Hubert, DeltaCAD
- J.M. Lacombe, EDF
- B. Marchand, DeltaCAD

1. Bilan de l'organisation actuelle de Fudaa

Fudaa est une plateforme qui offre de nombreuses possibilités, mais dont l'ampleur prise en neuf ans d'existence pose aujourd'hui un certain nombre de difficultés.

a) Complexité de la plateforme

La plateforme Fudaa est difficile de prise en main pour un nouvel utilisateur, car très vaste, complexe, peu documentée, avec des normes d'écriture hétérogènes, etc. Ce phénomène est accentué par le fait que la conception d'un projet nécessite actuellement la récupération préalable de l'ensemble des projets de la plateforme (10000 fichiers java environ).

b) Difficultés liées aux développements parallèles, problèmes de régression de logiciels Fudaa

Les développeurs écrivent leur code chacun de leur côté, parfois sans :

- connaître l'activité des autres, ce qui peut par exemple avoir pour conséquence des double développements d'une même fonctionnalité.
- appréhender les conséquences que cela peut avoir sur d'autres logiciels ; d'où parfois des problèmes de régression, qui peuvent apparaître lorsque les couches sous-applicatives communes sont modifiées pour les besoins d'un logiciel. L'utilisation de branches de développement ne fait que retarder l'apparition du phénomène.

Exemples récents : la version 0.92 de Fudaa-Prepro qui était très boguée (sur des points qui fonctionnaient parfaitement avec la version 0.91), Fudaa-Refonde qu'il était impossible de recompiler.

2. Évolution de la plateforme

a) Documentation

Afin d'améliorer la prise en main de Fudaa par les nouveaux développeurs, il est souligné l'intérêt de rédiger des documentations d'ensemble (description d'architecture, description de concepts tels que les design patterns ou la diffusion des événements, tutoriels), ainsi que des classes d'exemples, l'avantage de ces dernières résidant dans le fait qu'elles évoluent avec le code.

b) Nouvelle organisation de la plateforme

La plateforme Fudaa sera désormais organisée différemment, de manière à réduire les difficultés de complexité et de régression présentées précédemment.

Chacune des différentes couches de la plateforme (Ctulu, Dodico, Ebli et Fudaa) sera découpée en projets versionnés cohérents et de granularité à définir ; ils seront ensuite mis sous forme de paquetages jar. Quelle que soit la granularité retenue, ces projets ne pourront contenir à la fois des éléments communs et des éléments applicatifs.

Au sein du système de gestion des sources SVN, cette nouvelle organisation se traduira par la création d'une arborescence dédiée aux sources applicatives et d'une arborescence contenant les sources des projets communs. Les paquetages versionnés seront quant à eux déposés sur un repository.

Ainsi, le développeur souhaitant concevoir un logiciel récupèrera les paquetages communs nécessaires sur le repository et écrira les sources de son logiciel. Il pourra également modifier les sources de projets communs et publier ainsi une nouvelle version de ces projets, sans pour autant que cela impacte les autres logiciels. Le développeur d'une autre application pourra bénéficier de ces modifications en utilisant la nouvelle version des projets communs pour son logiciel.

Un nettoyage des classes inutiles pourra intervenir à l'occasion de cette réorganisation. Toutefois, une première version des projets correspondra au contenu actuel de Fudaa, afin de prévenir toute erreur.

Remarques concernant le versionnement :

- Les numéros de version des paquetages seront de la forme « majeur.mineur.bugfix » ; la notion de version majeure sera définie ultérieurement.
- Lorsqu'un développeur réalisera une correction de bug sur une version d'un projet commun, il la répercutera également sur la branche commune et, dans le cas où des versions plus récentes de ce projet existent, il avertira la communauté qui pourra alors répercuter la correction de bug sur ces versions.

3. Définition de lignes directrices

a) Adoption de nouveaux outils

Afin de mettre en œuvre la nouvelle organisation décrite précédemment, l'outil de gestion de projet « Maven » remplacera désormais l'outil « Ant » utilisé jusqu'à présent. Un serveur fudaa.fr sera prochainement opérationnel pour héberger un repository (FTP dans un premier temps) contenant les paquetages versionnés. Le système de gestion de sources SVN sera quant à lui toujours hébergé sur SourceForge.net.

L'outil Maven pourra permettre, dans un second temps, de réaliser des sites projets contenant notamment une documentation ainsi qu'un rapport d'avertissement et d'erreurs.

Un outil de suivi de bogues et de gestion de projet (Trac ou Jira) sera mis en place sur le serveur fudaa.fr. Cet outil devra pouvoir s'interfacer avec le système de gestion de sources SVN, et également permettre de présenter aux utilisateurs et développeurs une roadmap.

Un outil d'intégration continue (Hudson par exemple) permettant, à intervalle régulier, de tester le code sera mis en place sur le serveur fudaa.fr. Dans un deuxième temps, cet outil pourra être complété par une application de contrôle de qualité telle que Sonar.

b) Licence

La licence GNU/GPL2 reste celle utilisée pour l'instant. Une réflexion sur le passage en LGPL pourra avoir lieu ultérieurement, sachant toutefois que ce dernier ne permettrait plus l'utilisation de bibliothèques sous licence GPL (actuellement environ 10% des bibliothèques).

c) Divers

Les standards de code utilisés jusqu'à présent, jugés désuets étant données les améliorations de lisibilité apportées par les environnements de développement intégrés, sont abandonnés au profit des normes Sun. Une réflexion concernant l'uniformisation des règles de formatage (éventuellement avec l'ajout de règles dans les outils Checkstyle et PMD) sera menée.

Les nouveaux projets utiliseront désormais l'outil `java.util.logging`, inclus en standard avec le JRE, en remplacement de FuLog.

Le framework swing Bu est pour l'instant conservé ; le sujet de son remplacement pourra être abordé lors d'une réunion ultérieure.