

interview Romain Guy, qui a travaillé tout un temps dans l'équipe Swing de Sun par rapport à l'annonce de Sun de rendre Java Open Source

par [Romain Guy](#)

Date de publication : 29 Septembre 2006

Dernière mise à jour : 29 Septembre 2006

Suite à la publication récemment faites par Sun de sa ROADMAP concernant L'Open Sourcing de Java, nous avons demandé à Romain Guy de nous accorder un interview à ce sujet.

Cet article est également disponible au [format PDF \(mirroir http\)](#)

Le fait que Sun ait annoncé que Java serait open source pour cette année (du moins ses composants principaux) fait beaucoup de bruit. Penses tu que cela va aider Java ? à le rendre meilleur ?

GFX: En ce qui me concerne j'ai un avis partagé concernant l'Open Sourcing de Java.

Je trouve que Java est suffisamment ouvert, et ce depuis un moment.

Le code source est disponible non seulement pour les API mais aussi pour la JVM depuis plusieurs années.

On ne peut certes pas l'utiliser à sa guise mais il permet de comprendre les choses et de faire des bug reports ou de proposer des patches.

Le récent projet PeaBody, qui permet à tout un chacun de contribuer directement au JDK, est une avancée supplémentaire.

À l'instar du JCP, je trouve que ces solutions sont un bon compromis: Elles permettent de conserver la cohérence de la plateforme, et de la marque, Java sans nous empêcher, les développeurs, de consulter le code et d'en tirer certains bénéfices.

Quelle que soit la licence choisie, Sun interdira l'utilisation du nom Java sur des forks incompatibles.

Je doute que quoi que ce soit de grâve arrive à Java mais les vives critiques à son égard pourraient bien conduire à la création de dérivés incompatibles qui ne feront que compliquer la tâche des développeurs et des utilisateurs.

J'espère en tout cas que cela permettra d'améliorer Swing, même si j'espère de tout coeur que personne n'osera introduire un nouveau look and feel horrible dans le JDK (comme Substance par exemple).

Concernant les bug reports et les patches, il semble que Sun ne va tout de même pas donner de droits en écriture à tout le monde. On va donc tout de même se retrouver dans la même situation que maintenant, où la seule chose qu'un développeur pourra faire, c'est remplir des bug reports et/ou de proposer des patches.

Aussi, j'ai noté que la plupart des projets qui ont été lancé sur [java.net](#), comme le [javadesktop](#) et d'autres sur lequel tu travailles, sont généralement sous licence LGPL et tout le monde peut également y participer.

Donc, je ne sais pas trop non plus ce que cela va nous apporter, en tant que développeur Java.

GFX: Avec une licence Open Source on pourra télécharger les sources et créer un nouveau projet ailleurs avec.

Pour moi l'Open Sourcing de Java est plus une question d'image. J'ai surtout peur de ce que va faire IBM...

Tu parles du Projet Peabody. Est ce bien [jdk6.java.net](#) et [jdk7.java.net](#) ?

Ou peabody c'est encore autre chose ?

GFX: PeaBody c'est le projet de collaboration au JDK.

Cela comprend donc [jdk6.dev.java.net](#) et [jdk7.dev.java.net](#) en effet.

Tu disais que quelle que soit la licence choisie, Sun interdira l'utilisation du nom Java sur des forks incompatibles.

Personnellement, je l'espère bien. Sinon, on va droit à la catastrophe.

Mais je tiens tout de même à signaler que lorsque tu installes certaines distributions Linux, et que tu tapes `java -version`, tu as bien souvent un truc du style Kaffe ou gcj qui ne sont pas des implémentations complète de Java. On a donc déjà aujourd'hui des forks incompatibles, et Sun n'a apparemment rien fait pour les empêcher d'utiliser le nom de Java.

GFX: Oui et non. Ces distributions ne portent pas le nom "Java" (qui est une marque déposée).

Si le système de packaging les place sous le nom java, c'est un autre problème.

JGoodies corrige certaines erreurs de look and feel.

Penses-tu qu'il y ait des chances que son auteur le donne pour que cela puisse être intégré dans le JDK.

GFX: Karsten travaille depuis 2 ans avec l'équipe Swing. Il est subventionné par Sun pour corriger des problèmes des LaF.

Quand tu parles de Substance, tu parles bien de substance.dev.java.net ?

GFX: Oui. Techniquement fabuleux, visuellement atroce :(

Aussi, j'ai lu dernièrement que Sun envisageait de retirer le module MIDI du JDK.

Serait-ce les prémices d'un nettoyage de l'Api Java ? Où on verrait enfin les méthodes dépréciées de Date disparaître par exemple ?

GFX: Ce n'est pas un véritable nettoyage de l'API mais plutôt un moyen efficace et simple de réduire la taille du JDK : l'API MIDI nécessite des fichiers ressources assez lourds. Comme personne, ou presque, ne s'en sert. C'est toutefois possible uniquement grâce à la JSR [jesaispluscombien](#) pour gérer les modules. Les méthodes dépréciées ne disparaîtront pas. Cela ne ferait que briser de nombreux programmes et l'intérêt est quasi nul.