

Compte-rendu de JavaOne 2008

par Yann D'ISANTO ([Autres articles](#))

Date de publication : 30/05/2008

Dernière mise à jour : 30/05/2008

Voici un compte rendu de ma semaine passée à assister à JavaOne 2008, le plus gros rendez-vous annuel autour des technologies Java.

- I - Avant propos
- II - JavaOne - J1 : arrivée et CommunityOne
- III - JavaOne - J2
- IV - JavaOne - J3
 - IV-A - TS-6605 : Deep Inside JSR 296, the Swing Application Framework
 - IV-B - Scripting panel
 - IV-C - TS-6050 : Comparing JRuby and Groovy
 - IV-D - Table ronde NetBeans
 - IV-E - TS-5199 : JMX : Java Management Extensions (JMX) Technology Update
- V - JavaOne - J4
 - V-A - TS-4895 : The NetBeans IDE Compared to the Eclipse Rich Client Platform
 - V-B - TS-5541 : Creating Better Applications at Boeing with the NetBeans Platform Application Framework
 - V-C - TS-6656 : Extreme GUI Makeover
 - V-D - TS-6611 : Filthy-Rich Clients: Filthier, Richer, Clientier
- VI - JavaOne - J5
- VII - Le mot de la fin
- VIII - Remerciements

I - Avant propos

Comme chaque année, Sun organise la plus grosse conférence mondiale Java au Moscone (prononcez "mosconi") Center à San Francisco. Cette année, elle a eu lieu la semaine du 5 au 9 mai et j'ai eu la chance d'être le représentant de developpez.com pour couvrir cette conférence.

L'inscription à la conférence donne accès à une application web permettant de construire son planning avec les sessions auxquelles on veut assister. Il y a énormément de sessions et cela peut donner lieu à des choix cornéliens.

Un peu partout dans le bâtiment, des stations de travail sont accessibles mais elles sont littéralement prises d'assaut.

II - JavaOne - J1 : arrivée et CommunityOne

Comme le veut la tradition, le premier jour est réservé au Community One. Malheureusement je n'ai pas pu assister au keynote du matin, mon avion ayant atterri à San Francisco à 12h30 heure locale.

Après avoir déposé mes affaires à l'hôtel je suis allé récupérer mon badge permettant l'accès aux salles de la conférence et j'ai également eu droit à cette occasion à quelques goodies (un sac à dos, un stylo, un carnet, un CD).

J'ai ensuite assisté à deux tables rondes respectivement menées par Simon Phipps (Open Source) et Zack Urlocker (MySQL).

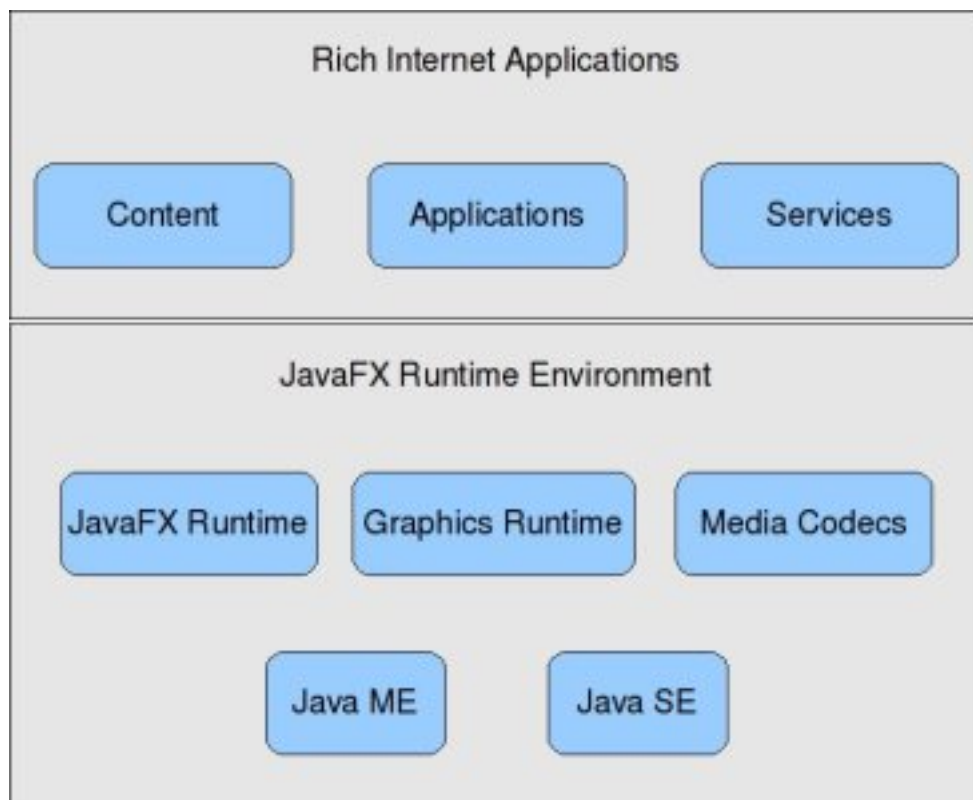
Lors de la première, il a beaucoup été parlé du virage amorcé par Sun il y a quelques années vers l'Open Source. Sa place s'y fait de plus en plus importante (avec notamment le récent rachat de MySQL) et son objectif est de devenir le premier contributeur du monde Open Source. Jonathan Schwartz, le CEO de Sun, a par exemple indiqué que ZFS (système de fichiers de Solaris) devrait bientôt passer sous licence GPL.

Comme on pouvait s'en douter, Sun compte améliorer le support de MySQL dans le monde Java, notamment en rapprochant l'équipe de MySQL de celles du monde Java (on pense notamment aux grands frameworks tels que Spring, etc.). Il a clairement été indiqué que Sun ne vise pas, avec MySQL, le marché des SGBDs client/serveur tel que Oracle et SQL Server (pour par exemple les ERPs) mais reste dans l'optique des applications orientées web. La question de l'intégration de Java à MySQL a également été soulevée (procédures stockées en Java ?) mais cela ne reste à l'heure actuelle qu'au stade d'idée.

III - JavaOne - J2

Cette journée a commencé avec une session générale dont la vedette était JavaFX et où nous avons eu droit à de superbes démonstrations (malgré quelques petits plantages).

Une première démonstration montrait une application tournant dans un navigateur, celle-ci a alors été "transférée" sur le bureau par un simple drag&drop. Nous avons également pu voir une application lançant simultanément la lecture de 200 vidéos en HD et dont les miniatures se déplaçaient en formant une sphère (multimédia + 3D). Voilà ce qui nous attend avec JavaFX. SUN a clairement de grosses ambitions pour JavaFX qui va plus loin que les RIA déjà présents sur le marché. JavaFX est destinée à devenir une plateforme universelle tournant au dessus de Java SE et Java ME.



Cependant il va falloir attendre encore un peu avant de pouvoir y goûter. La version 1.0 de JavaFX Desktop ne sera disponible qu'en automne 2008 et la version 1.0 de JavaFX pour Mobile et TV ne sera, quant à elle, disponible qu'au printemps 2009.

J'ai ensuite pu assister à plusieurs sessions de questions/réponses avec Param Singh (JavaFX), puis Jonathan Schwartz, et enfin James Gosling. Bien entendu le sujet principal de conversation a été JavaFX. Sun souhaite toucher un maximum d'appareils et James Gosling lui-même a avoué que, en ce qui concerne les appareils mobiles, cela n'était pas une tâche aisée et nécessitait encore beaucoup de travail.

JavaFX arrive donc assez tard, et la bataille sur le marché des RIA bat déjà sont plein avec SilverLight (Microsoft) et AIR (Adobe). JavaFX a de quoi séduire, son framework est très riche et orienté multimédia (avec JavaFX, Sun veut "rapprocher le designer du développeur"). JavaFX devrait également supporter un large ensemble de codecs (bien que l'on n'en connaisse pas encore la liste) et intégrer un système de base de données (JavaDB pour Java SE et SQLite pour Java ME).

Un autre point intéressant est le fait que JavaFX soit 100% open source (cela devient une habitude chez Sun) au contraire de ses deux concurrents.

Cependant, son retard est son principal point faible, beaucoup de zones d'ombres subsistent : on ne connaît pas encore les spécifications nécessaires pour faire fonctionner JavaFX et, même s'ils sont prévus, les outils de développement (qui seront sûrement intégrés à NetBeans sous la forme de plugin) n'ont pas de roadmap bien définie.

Sun compte néanmoins sur la large présence de la JVM sur les PC (90%), les mobiles (85%), mais aussi sur tous les lecteurs Blu-Ray pour faciliter l'adoption de JavaFX.

IV - JavaOne - J3

IV-A - TS-6605 : Deep Inside JSR 296, the Swing Application Framework

J'ai commencé ma journée par la session technique TS-6605 : "Deep Inside JSR 296, the Swing Application Framework" présentée par Hans Muller et Tomas Pavek.

La JSR 296 propose un framework pour le développement d'application Swing. Le framework permet de construire facilement une application Swing en se basant sur une structure commune à la plupart des applications desktop. La session était plutôt convaincante notamment grâce à une démonstration s'appuyant sur l'IDE NetBeans qui permet de créer un projet basé sur le Swing Application Framework.

Ce framework apporte un support relativement avancé pour :

- le cycle de vie de l'application,
- la gestion des ressources de l'application (injection),
- la simplification pour la création des actions (annotation),
- l'exécution des tâches de fond (avec notamment la gestion de ce qui s'exécute ou pas dans l'EDT en se basant sur la classe SwingWorker),
- la sauvegarde de l'état graphique (position et taille de la fenêtre, ...).

Son utilisation n'est pas très compliquée, plutôt efficace, et permet d'éviter pas mal de "pièges" relatifs au développement d'une application Swing.

IV-B - Scripting panel

J'ai ensuite assisté à une session de questions/réponses sur le thème des langages de script et langages dynamiques à laquelle participait Charles Nutter (JRuby), Tor Norbye (NetBeans), Ted Leung (Jython), Guillaume Laforge (Groovy), Greg Murray (jMaki) et Tim Bray (Sun).

En effet, les langages dynamiques supportés par la JVM fleurissent (Scala, ..., JavaFX qui possède lui aussi son propre langage de scripting) et leur nombre grandissant soulève des interrogations. Pourquoi multiplier les langages supportés par la JVM ? Quel est l'intérêt des langages dynamiques ? Comment choisir quel langage utiliser ? ...

Avec l'ouverture de la plateforme Java, il est logique de voir le support d'autres langages arriver. L'avantage est que chaque langage apporte sa spécificité, par exemple : Ruby a une syntaxe plus flexible (rapide à écrire), Python est un langage plus concis (rapide à lire), Groovy apporte une haute intégration des fonctionnalités "entreprises", etc.

De plus ces langages peuvent dès lors profiter de toute la plateforme Java (par exemple JRuby est plus rapide que Ruby). Les langages dynamiques sont plus souples, apportent une syntaxe plus simple, et génèrent moins de code. Le choix du langage peut dépendre pour une petite part de spécificités techniques mais se résume généralement plus à une question de goût.

IV-C - TS-6050 : Comparing JRuby and Groovy

Sur le même sujet, la journée a continué avec la session technique TS-6050 sur la comparaison entre Groovy et JRuby animée par Neal Ford. Je ne connaissais pas ces deux langages et cela a été un bon moyen pour moi de les découvrir.

Leur comparaison a été faite sur les points suivants :

- la syntaxe,
- la philosophie,
- les méthodes de test,
- la meta programmation.

Ces deux langages se ressemblent assez malgré quelques différences qui se révèlent surtout être des différences de philosophie. Ce qui est faisable avec l'un l'est généralement aussi avec l'autre, c'est plutôt dans la façon de faire qu'il y a une différence. Même en ce qui concerne les performances ces deux langages se révèlent semblables.

IV-D - Table ronde NetBeans

J'ai ensuite participé à une table ronde sur NetBeans avec David Folk. Il a d'abord été question de la sortie de NetBeans 6.1 avec notamment le support de PHP et Javascript, l'amélioration du support de MySQL, des WebServices et de Spring.

Tout comme la plateforme Java, NetBeans s'ouvre de plus en plus aux autres langages. Après Ruby c'est au tour de Python dont le support est actuellement en cours de travail. Le support d'autres langages (Groovy, Scala, etc.) dépendra de la demande de la communauté.

Bien entendu il a également été question du support de JavaFX, la star de ce JavaOne. NetBeans offre déjà un support pour JavaFX Script. Comme la plateforme JavaFX cible les designers, il a été demandé si Sun fournira des outils de création pour JavaFX dans NetBeans. La réponse est oui : Sun a bien prévu de fournir une suite d'outils, malheureusement le quand n'est pas encore défini. A noter que cela ne fera pas partie du projet NetBeans, mais que cela se présentera plutôt sous la forme de plugin pour l'IDE.

IV-E - TS-5199 : JMX : Java Management Extensions (JMX) Technology Update

Changement de sujet avec ma dernière session technique de la journée qui portait sur JMX (TS-5199), session présentée par Jean-François Denise et Eamonn McManus.

La session a commencé par une présentation de l'API (ce qui tombait bien puisque je ne la connaissais pas) puis par une description des nouveautés prévues pour la prochaine version.

Pour ceux qui ne connaissent pas, JMX est une API (présente depuis Java 5) qui permet de gérer et monitorer une application qui s'exécute dans la JVM. Par exemple, la console Java JConsole exploite JMX pour restituer ces informations et administrer une application. Une version 2.0 de cette API devrait normalement être intégré à Java 7 via les JSR 255 et 262. Les possibilités de cette API nous ont été montrées au travers d'une démonstration plutôt convaincante utilisant le projet **CSPoker**.

V - JavaOne - J4

V-A - TS-4895 : The NetBeans IDE Compared to the Eclipse Rich Client Platform

Fervent utilisateur de NetBeans, je m'étais déjà intéressé à sa plateforme mais avais vite abandonné, surtout par manque de temps. J'ai donc choisi la session technique TS-4895 présentée par Kai Toedter (Eclipse) et Geertjan Wielenga (NetBeans) afin d'en apprendre un peu plus.


La comparaison a été illustrée par une application de démo, "MP3 Manager". Les deux plateformes se ressemblent beaucoup et ont une architecture très semblable (principe des plugins), dans les deux cas il est possible de créer des projet pour développer une application utilisant la plateforme considérée. En fait, leur principale différence réside dans le fait qu'Eclipse utilise SWT là où NetBeans utilise Swing. Mais sinon à peu près tout ce qui est faisable avec l'une l'est aussi avec l'autre, et ceci de façon assez similaire.

V-B - TS-5541 : Creating Better Applications at Boeing with the NetBeans Platform Application Framework

Toujours afin de mieux connaître la plateforme NetBeans, j'ai choisi d'assister à la session technique TS-5541 animée par Bruce Shimmel et Tom Wheeler.

Malheureusement je dois avouer être quelque peu resté sur ma faim, cette session étant bien moins technique que ce à quoi je m'attendais.

En effet, c'est surtout la question du pourquoi choisir la plateforme NetBeans qui a été traitée : ne pas réinventer la roue et se concentrer sur les développements spécifiques à l'entreprise.

Boeing a donc construit une plateforme ( **Boeing NetBeans Platform**) par dessus celle de NetBeans qui apporte notamment le support de JFreechart et d'un viewer Java3D. C'est cette plateforme qui est utilisée par les ingénieurs de Boeing et qui leur permet de produire une application riche en approximativement une heure.

V-C - TS-6656 : Extreme GUI Makeover

J'ai continué ma journée très orientée "desktop" avec la session technique TS-6656 animée par Christopher Campbell et Ben Galbraith. Cette session (ainsi que la suivante) est à recommander aux personnes qui pensent que Java est lent et moche ainsi qu'aux fans d'interfaces graphiques conviviales et évoluées. C'est le genre de session où l'on apprend, ou revoit, plein de petites techniques (en l'occurrence sur Swing) telles que


- choisir entre réécrire la méthode paintComponent où implémenter un UIDelegate pour personnaliser un composant graphique,
- faire son propre Look and Feel, en choisir un déjà existant et le plus proche du voulu pour ensuite le modifier,
- utilisation du glasspane et des JLayeredPane pour les animations,
- utilisation le timing framework pour les animations,
- ...



La démonstration s'appuyait sur une très jolie interface aux effets vraiment 'wouaw !' et montrait pas mal de code.

Un point qui a été souligné est que le design des interactions est différent du design visuel et que c'est souvent là qu'une application pêche.

V-D - TS-6611 : Filthy-Rich Clients: Filthier, Richer, Clientier

En continuité de la session précédente, j'ai terminé ma journée par la session technique TS-6611 animée par Romain Guy et Chet Haase. Là encore on en prend plein les mirettes et tout développeur Swing est aux anges.

Cette session peut être considérée comme un "add-on" à leur livre  **Filthy Rich Clients**, que personnellement je recommande à tous les développeurs Swing.

Il a d'abord été question du problème engendré par un nombre d'animations trop élevé. En effet, chaque animation se base généralement sur un Timer et leur multiplication entraîne une chute dramatique des performances. La solution est de n'utiliser qu'un seul Timer pour toutes les animations en utilisant soit la classe `TimingSource` du  **Timing Framework**, soit le projet `Scene Graph`  **Scene Graph**. Le tout était illustré par du code et une démonstration.

Nous avons ensuite assisté à la mise en place d'effets graphiques relativement simples mais très efficaces pour le drag&drop d'images, là encore démonstration et code à l'appui.

VI - JavaOne - J5

Ce cinquième et dernier jour a commencé par un hommage à John Gage puis par le traditionnel "Gosling's Toy Show" aussi nommé "Extreme Innovation" où James Gosling présente son classement des meilleurs innovations technologiques utilisant Java.

- **Java VisualVM** : *VisualVM* et une application de monitoring, de profiling et d'analyse de vos applications Java. Une sorte de JConsole avancée.

->  <https://visualvm.dev.java.net/>

- **JavaScript Support in Netbeans** : *NetBeans* offre, depuis sa version 6.1, un support complet de JavaScript avec notamment la complétion de code et un débogueur.

->  <http://www.netbeans.org/>

- **Java on NVidia APX 2500** : Support de Java par l'*APX 2500* qui est un processeur pour smartphone pouvant traiter des images d'une résolution de 12 mégapixels et possédant un coeur graphique 3D de classe GeForce 6 avec lequel il devient possible d'exécuter une application OpenGL écrite pour le desktop en utilisant de simples bindings OpenGL.
- **Java Games and Project Darkstar** : Présentation du projet *Darkstar* pour le développement de jeux vidéos en Java, avec en exemple le jeu *Call of the Kings*.

->  <http://www.projectdarkstar.com/>

->  <http://callofthekings.com/>

- **Extreme Java Card Innovation** : Mise en oeuvre de l'API *Robocode* sur une carte à puce au travers d'un concours.
- **Pervasive Java** : L'entreprise *Sentilla* propose une plateforme logicielle basée sur Java pour des applications embarquées.

->  <http://www.sentilla.com/>

- **Never Miss a Word** : Un stylo intelligent capable d'enregistrer ce que vous écrivez, de le traduire, de jouer sur un piano que vous auriez dessiné ...

->  <http://www.livescribe.com/>

- **Industrial Strength Java** : Présentation de l'architecture *Blue Wonder* qui repose sur une combinaison de PC x86, Solaris et Java RTS pour une application de contrôle industriel.
- **Java : Licence to Drive** : *Tommy Junior* est une voiture autonome qui utilise une plateforme logicielle 100% Java (Java RTS, Java SE, Java ME).

->  <http://www.teamjefferson.com/>

- **Java Rocks on Mars** : *JMars* est un système d'information géospatial basé sur la technologie Java utilisé pour faire de l'analyse de données et de la planification de mission par la NASA (Mars Odyssey, Mars Reconnaissance Orbiter et bientôt Lunar Reconnaissance Orbiter).

->  <http://jmars.asu.edu/>

- **CERN : Accelerating Java** : L'*organisation européenne pour la recherche nucléaire* utilise Java pour gérer le LHC (*Large Hadron Collider* ou *Grand collisionneur de hadrons* en français), un gigantesque accélérateur de particules digne des meilleurs films de science-fiction. Avec le détecteur ATLAS on parle d'un flux de données de 2 péta-octets/seconde (soit environ 3 millions de CDs par seconde).



->  <http://public.web.cern.ch/Public/fr/LHC/LHC-fr.html>

Malheureusement, mon avion décollant de San Francisco à 12h, mon JavaOne s'est terminé là.

VII - Le mot de la fin

Ce JavaOne a donc beaucoup tourné autour de JavaFX qui a énormément d'atouts pour faire une entrée remarquée dans le monde des RIA. Cependant, son retard est une pénalité sévère et tout retard supplémentaire pourrait bien lui être fatal (notamment en ce qui concerne les outils de développement).

Un autre point sur lequel Sun a beaucoup communiqué est l'ouverture (au sens le plus général du terme) de la plateforme Java avec notamment un support accru pour d'autres langages sur la JVM.

A noter que Sun commence à publier l'ensemble des sessions sur  <http://developers.sun.com/learning/javaoneonline/> (toutes ne sont pas encore disponibles). Des vidéos des différentes sessions générales sont également disponibles :  <http://java.sun.com/javaone/sf/sessions/general/>

Enfin, il y a des photos :

-  [sur le site de Sun](#)

-  [sur Flickr](#)

Sur un plan plus personnel, ce fût une semaine vraiment inoubliable et j'espère bien avoir la chance d'y retourner une prochaine fois.

VIII - Remerciements

Je tiens à remercier tout particulièrement Bruno Hourdel pour sa gentillesse et son accueil chaleureux à JavaOne. Je remercie également Adrien Artero qui m'a généreusement hébergé sur Paris. Je remercie aussi Ricky81 et bouye pour leurs corrections.

Je remercie enfin toutes les personnes que j'ai pu croiser au cours de cette semaine.

