



- *LINUX : Choisir sa distribution*

- *Eclipse vs NetBeans : la guerre des IDE*
- *La jungle des licences !*
- *Le danger des rootkits sous Linux*

RÉUSSIR avec l'OPEN SOURCE

LA RÉVOLUTION DE L'INTERFACE

- *Ajax : le futur du Web !*
- *RAILS : le nouveau client riche*
- *Looking Glass : l'interface 3D selon java*

Printed in France - Imprimé en France -
BELGIQUE 6,45 € - SUISSE 12 FS -
LUXEMBOURG 6,45 € - Canada 8,95 \$ CAN

M 04319 - 83 - F: 5,95 €



LINUX : Programmer D-BUS / **.NET** : Découvrir le
P2P avec Groove / Un contrôle utilisateur en C#
PHP : Le framework Noas

PostgreSQL

La base de données open source vraiment sans coût de licence



PostgreSQL versus MySQL

Comparaison complète à l'adresse
www.pervasive-postgres.com/programmez

Des difficultés à choisir votre base de données **open source**?



Contrairement à certaines autres bases de données open source, PostgreSQL vous permet de distribuer et de commercialiser vos applications sans coût de licence

La comparaison complète de [PostgreSQL avec MySQL](#) au niveau licence, technologie et fonctionnalités va vous aider à vous décider.

Visitez www.pervasive-postgres.com/programmez pour la consulter.



sommaire

n°83

/ACTUS

L'actualité en bref.....	6
Agenda	11

/SÉCURITÉ

Faillle WMF : 12 jours d'attente	10
--	----

/TESTS

Eclipse vs Netbeans : match nul entre deux champions	12
--	----

/DOSSIER RÉUSSIR AVEC L'OPEN SOURCE

Le marché Open Source/Linux : dynamique et pragmatique	17
Microsoft mise sur les licences et la communauté.....	24
Développeurs Open Source : un marché porteur pour l'emploi.....	26
Rootkits & Linux : la menace fantôme	28
Les distributions Linux.....	30
Open Source : double licence et modèle économique.....	34
Les meilleurs outils Open Source.....	37
Programmation de D-BUS sous Linux	42

/CARRIÈRES

Sogeti: dans la cour des grands.....	47
--------------------------------------	----

/FOCUS : L'INTERFACE FAIT SA RÉVOLUTION !

Clients riches : vers des interfaces agiles	50
Les interfaces faciles : vers la dématérialisation.....	51
RubyOnRails : du client riche facile.....	53
A la découverte de Looking Glass	56
Les approches de développement avec Ajax	60

/PRATIQUE

Persistance des données avec Noas PHP	64
Groove Virtual Office : une architecture originale, dédiée au travail en équipe	66
Réaliser un contrôle utilisateur sous Visual Studio.Net.....	71
Les bases du langage Erlang (2e partie).....	75

/TEMPS LIBRE

Programmez un jeu pour votre Pocket PC en .Net (2e partie)	78
Livres.....	82

CD-ROM PROGRAMMEZ N°83



LinuxDefender
L'outil de secours
ANTIVIRUS **GRATUIT**
Sur CD-Rom bootable avec GNU Linux/Knoppix

bitdefender
secure your every bit
ANTIVIRUS & ANTISPAM
Serveurs et stations de travail - Linux et Windows
Plètement fonctionnel 30 jours

LinuxDefender
LinuxDefender est un outil qui intègre les dernières solutions de sécurité BitDefender pour Linux proposées via une distribution GNU Linux/ Knoppix sur un CD-Rom bootable.

Ce CD comprend un antivirus capable de scanner et désinfecter les disques durs existants, les partages réseau Samba et Windows ainsi que les points de montage NTFS. Vous pouvez ainsi démarrer une machine qui ne fonctionne plus correctement dans un environnement sain, puis analyser et désinfecter cette machine directement depuis l'interface du CD. Une interface de configuration Web des solutions BitDefender est également disponible.

BitDefender
BitDefender : Solution de protection en environnement Windows et Linux. versions d'évaluations pleinement fonctionnelles pendant 30 jours.

Conseils d'installation
LinuxDefender est un CD bootable et ne nécessite aucune installation préalable. Toutefois, assurez vous que le BIOS est configuré pour démarrer en priorité sur le CD-Rom, avant d'insérer celui-ci dans le lecteur.



Descriptif : un fichier html contenu dans ce CD décrit le fonctionnement des différents outils. Vous pouvez y accéder directement en ouvrant le fichier à l'emplacement suivant : /cdrom/Docs/index.html.

ÉDITIONS PROFIL



ÊTES-VOUS A LA RECHERCHE DU MEILLEUR MOYEN DE DEVELOPPER DES APPLICATIONS MULTIPLATEFORMES PERFORMANTES?

Plus besoin de choisir entre performance et portabilité – les deux sont possibles, grâce à Qt et son environnement de développement C++ multiplateforme. Avec Qt vous pouvez dorénavant développer des applications natives sur Windows, Unix/Linux et Mac OS X à partir d'un seul et unique code source. Sans couche d'émulation ni machine virtuelle, Qt vous permet de développer des applications hautement performantes sur toutes les plateformes.

Qt vous permet aussi d'accroître votre productivité. Plus besoin d'écrire et d'entretenir deux codes sources, vous pouvez dorénavant laisser libre court à votre créativité, Qt et son API simple d'utilisation se chargent des aspects non portables de votre application. Pas étonnant que des sociétés telles que Google®, Adobe®, Synopsys® et Perforce® aient adopté Qt pour leurs développements majeurs.

**Pour en savoir plus ou pour télécharger une version
d'évaluation, rendez-vous sur notre site :**
www.trolltech.com/lookhere.

Qt® BY TROLLTECH
CODE LESS. CREATE MORE.



QUOTIDIEN

www.programmez.com



A visiter absolument!
Actus, Forums, Téléchargement

Programmez!

Rédaction : redaction@programmez.com

Directeur de la Rédaction : Jean Kaminsky.

Rédacteur en Chef : François Tonic.

Ont collaboré : F. Barbin, N. Dasriaux, C. P. de Geyer, R. Djohou, A. Kerhuel, X. Leclercq, J. Lemaître, F. Mazué, C. Pitras, T. Riboulet, J. Saiz, S. Sibué, M. Varandat,

Maquette : PLB Communication.

Publicité : régie publicitaire : ECC, Stéphane Kaminsky - pour la publicité uniquement : Tél. : 01 56 31 21 54 - Fax : 01 56 31 21 59 - publicite@programmez.com.

Editeur : Go-02 sarl, 6 rue Bezout - 75014 Paris
Dépôt légal : 1^{er} trimestre 2006
Commission paritaire : 0707K78366 - ISSN : 1627-0908
Imprimeur : ETC - 76198 Yvetot

Directeur de la publication : Jean-Claude Vaudecrane.

Ce numéro comporte un CD Rom, un encart libre "Memento PHP".

Abonnement : Programmez 22, rue René Boulanger, 75472 Paris Cedex 10 - abonnements.programmez@groupe-gli.com - Tél. : 01 55 56 70 55 - Fax : 01 55 56 70 20 - du lundi au jeudi de 9h30 à 12h30 et de 13h30 à 17h00, le vendredi de 9h00 à 12h00 et de 14h00 à 16h30 - Abonnement (magazine seul) : 1 an - 11 numéros - France métropolitaine : 45 € (295,18 FF TTC) - CEE et Suisse : 51,83 € - Algérie, Maroc, Tunisie : 55,95 € - Canada : 64,33 € - Tom : 79,61 € - Dom : 62,84 € - Autres pays : nous consulter.

Liberté, gratuité, communauté

Nous sommes en plein syndrome du tout gratuit ou presque. On passe rarement une semaine sans lire ou voir un sujet sur Internet, le piratage, le Peer to Peer, les droits d'auteurs. Un opérateur proclame : « ne payez plus votre abonnement téléphonique ! » La valeur est dématérialisée, et le consommateur peut être tenté de croire que vendre c'est voler ! Le « tout gratuit » est souvent le dernier avatar du marketing, et il recèle des effets pervers.

D'un certain côté, l'Open Source contribue à renforcer cette impression du tout gratuit, mais il va plus loin.

Son ancêtre, le freeware, proposait déjà des logiciels gratuits, généralement sans le code source. L'Open Source pourrait se traduire par 3 commandements : gratuité, liberté, communauté. La gratuité car un logiciel ouvert est gratuit (pour la licence d'utilisation tout du moins). La liberté, car l'utilisateur n'est pas lié à un éditeur, même si on n'est jamais certain de la pérennité d'un projet ouvert. La communauté, car utiliser un logiciel ouvert c'est rentrer, au moins potentiellement dans une communauté, sans en avoir toujours pleinement conscience.

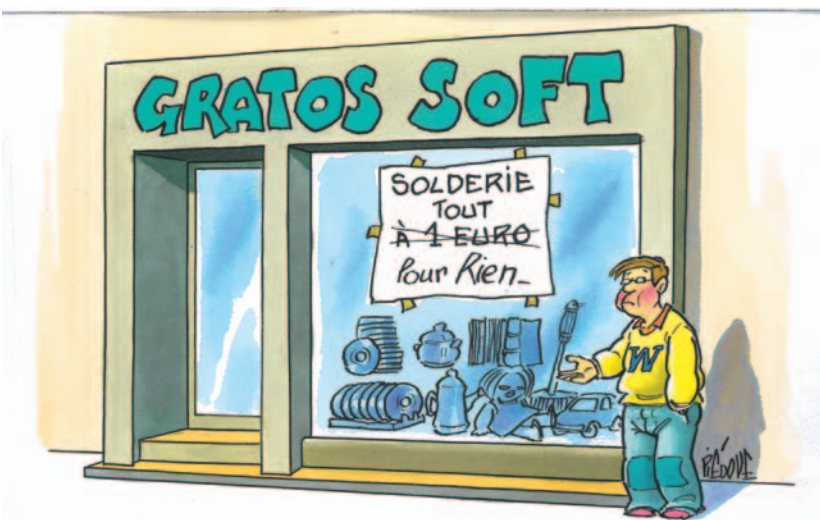
Contre la propriété, il y a la notion d'appropriation. Si on en possède les capacités, on peut participer à l'aventure du logiciel, et se l'approprier d'une certaine façon, ou en tout cas y ajouter sa griffe. Les débuts de l'Open Source et du logiciel libre étaient empreints d'une certaine utopie : dans l'esprit, une sorte de collectivisation du logiciel s'opposait au logiciel capitaliste. Mais il faut bien en vivre ! Dans le modèle économique actuel, les licences demeurent gratuites, mais les éditeurs libres proposent toute une gamme de services et de supports payants.

LOUÉ SOIT-IL !

Aujourd'hui, les éditeurs commerciaux, prenant conscience du problème, tentent de fournir une nouvelle réponse. Vous ne voulez pas d'un logiciel complet et cher ? Louez-le ! Vous trouverez alors des applications ou des fonctions répondant à un besoin. De la comptabilité ou du dessin ? Louez une application directement sur Internet. C'est le modèle de l'application distribuée, de l'ASP, qui va se généraliser. C'est une forme de liberté.

Google démontre cette nouvelle génération de logiciels avec Gmail ou Google Talk. Microsoft réplique avec Windows et Office Live. Imaginez de pouvoir taper une lettre à partir d'OpenOffice ou de MS Office directement sur le Web à partir de n'importe quel ordinateur connecté ? On louera à la durée, ou selon les fonctions désirées, n'importe quelle application. C'est aussi un choix de société. Quelle société de l'information souhaitons nous ? Toujours plus de mobilité, de virtualisation ?

François Tonic



PROCHAIN NUMÉRO n°84 - Mars 2006 - Parution le 25 Février 2006

DOSSIER

SÉCURISER

Votre code et vos applications

S'il faut se prémunir des virus et codes malicieux, le développeur doit aussi protéger son application des attaques. Blindez votre code source !

FOCUS

Le kit de survie du développeur

.NET

Le guide des composants et outils

COMPOSANT

NetAdvantage 2005 disponible

Après une longue phase de développement, d'adaptation et de tests, Infragistics a commercialisé la version .Net 2.0 de ses composants NetAdvantage. NetAdvantage 2005 for Visual Studio 2005 prend en compte les nouveautés des langages .Net 2 et de l'environnement Visual Studio 2005. La migration des anciens NetAdvantage vers 2005 ne devrait pas poser trop de soucis, selon l'éditeur. L'objectif demeure la simplification du travail du développeur. Au passage, l'éditeur a introduit une mise à niveau : dans le WinEditors il est possible d'étendre les colonnes dans un WinForm, format XHTML par défaut pour la documentation, déploiement simplifié des applications ASP.Net. Parmi les nouveautés, on notera un nouveau DataBinding, un nouveau système d'alignement (Base et Snap Lines dans les WinForms), le support des skin ASP.Net. Autre grande nouveauté : le support des smart tags. À partir de 495 dollars US. Version d'évaluation disponible.

Site : www.infragistics.com

Departure_City	Arrival_City	Departure_Date	Arrival_Date	First_Name	Row	Seat	Window
Chicago, IL	Orlando, FL	7/18/2007	7/19/2007	William	1	4	1F
Los Angeles, CA	Orlando, FL	7/21/2007	7/21/2007	Kerry	2	3	2F
Chicago, IL	Orlando, FL	7/18/2007	7/19/2007	William	1	4	1F
Los Angeles, CA	Orlando, FL	7/21/2007	7/21/2007	Kerry	2	3	2F
Houston, TX	Orlando, FL	7/18/2007	7/19/2007	William	1	4	1F
Phoenix, AZ	Orlando, FL	7/18/2007	7/19/2007	William	1	4	1F
Philadelphia, PA	Orlando, FL	7/18/2007	7/19/2007	William	1	4	1F

SOA

Rogue Wave se met à SOA

SOA devient un label incontournable. L'éditeur Rogue Wave va proposer à partir de mars prochain, un nouvel outil : Hydra 3.0 orienté SOA. L'outil prend en compte de multiples langages (C++, Java, BPEL, Web Services). Il supportera les standards du marché, se connectera via des connecteurs aux principaux middlewares et outils de messaging du marché et, bien entendu, s'intégrera avec les applications serveurs, frameworks et SGBD existants. L'un des objectifs d'Hydra est d'optimiser les performances de son architecture, tout en facilitant le développement des composants et la mise en place d'un processus business. Il s'agit surtout de donner toute l'agilité et la souplesse nécessaires à son environnement SOA. L'originalité d'Hydra est de pouvoir offrir un environnement distribué pour mieux tirer parti du matériel et donc obtenir de meilleures performances.

SOCIÉTÉS

La valse des rachats continue

Fin 2005 et début 2006 sont marqués par une recrudescence de rachats :

- Confirmation du rachat de Macromedia par Adobe,
- Mercury achète l'éditeur Systinet pour 105 millions de dollars. Systinet développe des solutions de gouvernance et de gestion de cycle de vie pour architecture SOA. Par cette opération, Mercury souhaite devenir un acteur majeur du marché SOA en pleine explosion.
- Progress Software rachète Neons Systems. NEON fournit des solutions sur l'accès aux données et aux applications stockées sur mainframe. NEON intégrera la filiale Datadirect Technologies de Progress spécialisée dans la connexion logiciels – données.
- Quest Software acquiert AfterMail Ltd, spécialisé dans la gestion des contenus de messageries. Pour Quest, il s'agit d'investir sur un marché

prometteur de la gestion et de l'archivage du contenu de messagerie. Quest étendra donc ses fonctions d'archivage aux outils de type Notes, Groupwise, Exchange...

- Après le rachat de Bowstreet Inc par IBM pour faciliter l'intégration des applications dans des portails, le Grand Bleu annonce son attention d'acquies Micromuse. Micromuse fournit des outils sur la gestion de réseau complexe supportant l'audio et la vidéo ainsi que la voix sur IP.

GESTION

Rational Portfolio Manager 6.2. IBM met à niveau son portefeuille

Pour gérer une multitude de projets, il faut disposer d'une solution dédiée, un portfolio. IBM vient d'annoncer Rational Portfolio Manager 6.2. Une étape importante pour IBM dans sa solution globale de cycle de vie des applications et de gouvernance. Dans cette version, on dispose : d'un outil de modélisation de processus de développement, de nouvelles fonctions de gestion des processus (Advanced Graphical Workflow Designer). Le côté analyse se renforce avec l'intégration de Team Unifying Platform. Il s'agit d'un tableau de bord pour visualiser données et métriques d'un projet. On dispose aussi d'un nouveau toolkit de développement.

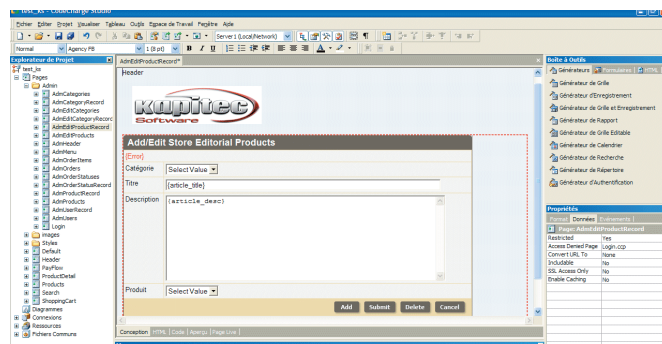
Site : www.ibm.com/rational

WEB

CodeCharge Studio 3.0

L'éditeur YesSoftware, distribué en France par Kapitec Software, a sorti la version 3 de son outil de développement Web : CodeCharge Studio. Il permet de développer des applications Web pour .Net, Java, PHP, ColdFusion. Cette version inclut un module de reporting web. Elle propose aussi de nouvelles fonctions de localisation afin de faciliter la conception d'application multilingue. On dispose aussi d'un nouveau générateur de calendrier web, très utile quand on développe ce type d'application. Pour le développement en équipe, l'environnement utilise les dernières versions de CVS et du Visual SourceSafe. Pour les données, on utilisera un générateur visuel de requêtes SQL amélioré. Pour le déploiement sur plusieurs serveurs, on dispose désormais du module « profils de serveurs » pour centraliser les paramètres serveurs. Version d'évaluation disponible.

Site : www.kapitec.com



SUIVEZ LA VOIE DE L'AUTOMATISATION



Et programmez moins!
Réduisez jusqu'à 70% les délais
et coûts de réalisation de vos applications

Leonardi Logiciel Client Universel

Téléchargez la version gratuite de Leonardi
sur notre site www.lyria.com

Leonardi est un framework JAVA/XML.
Son principe de fonctionnement "model driven"
permet de créer des applications prêtes à l'emploi
à partir d'un simple modèle de données.

Crée les IHM à la volée en fonction de la technologie
d'affichage choisie : HTML, DHTML, SWT, AWT, SWING, SVG.

Support transparent d'AJAX, intégration plate-forme
Eclipse, déploiement via JavaWebStart...

Connexion aux principales bases de données du marché :
Oracle, DB2, MS SQL Server, Sybase, MySQL, PostgreSQL...

S'intègre dans votre système d'information grâce à des
modules de connexion pour JMS, LDAP et serveurs
d'applications.

Gestion de la persistance des données et des transactions.

Solution portable sur toute plate-forme (Windows, Linux,
Unix, Mac OS...) grâce à l'utilisation du langage JAVA.

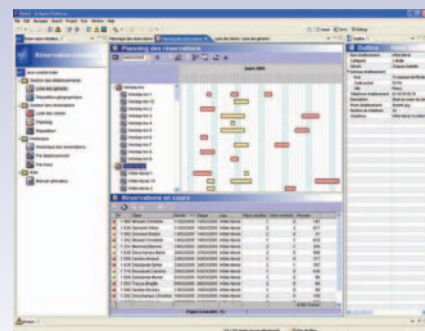
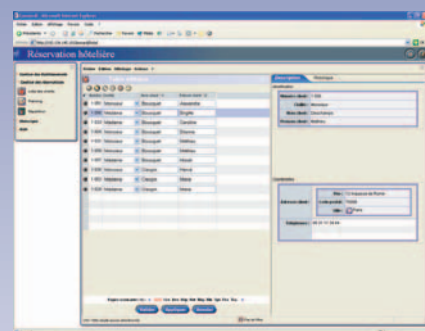
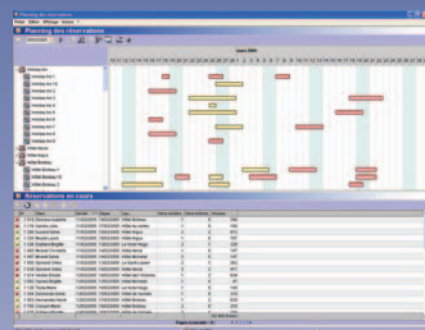
Livré avec Leonardi Studio, environnement de
développement intégré dans Eclipse.

Implémentation du code métier en JAVA.

Import de modèles de données UML à l'aide de
XMI.

Différents types de vues disponibles :

- formulaires de consultation et modification,
- tableaux,
- vues arborescentes,
- diagrammes de Gantt,
- vues synoptiques,
- vues statistiques...



Déployez votre application en WEB,
client lourd ou sous forme de plugin Eclipse
sans aucune modification!



www.lyria.com

Client Software Solutions

Pour en savoir plus sur la solution Leonardi, rendez-vous sur notre site www.lyria.com ou envoyez-nous un courriel à info@lyria.com.

WEB 2

Ajax surprend encore !



Bindows construit un nouvel ensemble de développement autour d'Ajax qui depuis presque un an monte en puissance, au point de devenir incontournable. Bindows repose sur une API proposant tous les objets d'interface que l'on peut trouver sur un système Windows, ou même en Java. Il supporte XML, SOAP, XML-RPC. Il fonctionne sur n'importe quel navigateur moderne. Un SDK est disponible pour les développeurs. Si le projet est intéressant, notamment par sa richesse, en production les lenteurs du concept Ajax peuvent être rédhibitoires.

Site : <http://www.bindows.net/>

IDE

Eclipse dévoile la Web Tools Platform

Depuis quelques semaines, la fondation Eclipse propose la Web Tools Platform 1.0. Elle définit un ensemble de standards et d'outils pour développer des applications web et J2EE. Les premiers à l'avoir adopté sont BEA, Borland, IBM, JBoss et Objectweb. Cette version constitue une première étape qui doit renforcer le poids d'Eclipse dans l'industrie, en définissant des standards pour le développement web et J2EE autour d'Eclipse, que les éditeurs pourront implémenter dans leurs propres solutions. La prochaine étape sera de prendre en compte Java 5 et WSDL 2.0 et de supporter Eclipse 3.2.

Site : <http://www.eclipse.org>

ACCÈS

Le développement multi mode sous Eclipse

Le projet Multimodal Tools Project for Eclipse, issu des développements d'IBM alphaworks, fournit aux développeurs web les éléments pour concevoir des applications web autorisant différentes possibilités d'utilisation : le clavier, l'écran tactile, l'encre numérique, la voix, etc. Il s'agit de définir un langage standard pour cet accès multi mode. Soutenu par Motorola, IBM et Opera Software, le XHTML+Voice doit être ce futur langage. Actuellement, le projet est une extension du Voice Tools Projets d'Eclipse. À l'avenir, il s'agira de laisser aux utilisateurs le choix de la méthode d'accès la plus appropriée.

Site : <http://www.alphaworks.ibm.com/tech/mmtp>

IDE

NetBeans 5.0 arrive !

Avec quelques semaines de retard, NetBeans 5.0 devrait être disponible en version finale, lorsque vous lirez ces lignes. Depuis début janvier, la version RC1 était téléchargeable. Cette version devrait confirmer la bonne santé de NetBeans sur le marché grâce à un nouveau module de conception d'interface, Matisse. On devrait rapidement avoir connaissance des prochaines versions.

Site : www.netbeans.com

Tribune libre

Faites ce que je dis, ne faites pas ce que je fais...



Nous autres, entreprises innovantes françaises devons une fière chandelle aux différents organismes publics qui nous écoutent, nous aident et nous soutiennent. Pour notre cas personnel, je pense que nous n'existerions plus, ou sûrement pas dans la forme actuelle, sans leur aide. Un très grand MERCI donc !

Tout ceci serait parfait, si dans le même temps nous n'avions pas une obstruction systématique quant à la fourniture de nos solutions auprès de différents ministères et autres organismes institutionnels sous prétexte que notre solution n'est pas Open Source.

Ne vous méprenez pas ! Je ne fais pas le procès des logiciels Open Source, ils existent. En tant qu'éditeur de logiciel il faut composer avec et d'ailleurs, il y a parmi ceux-ci beaucoup de bons outils, dont certains peuvent s'avérer meilleurs que leurs concurrents commerciaux. La question n'est pas là.

Seule la politique dogmatique du « Tout Open Source » et le mirage du « Tout Gratuit » me gênent.

A l'heure où nos dirigeants soutiennent l'idée d'un pacte PME, une des meilleures aides à nous accorder serait de nous considérer comme une solution potentielle et de ne pas oublier que pour aboutir à une application, il faut des outils additionnés à du travail et que le coût d'une solution ne s'apprécie qu'en ayant fait le compte global. Messieurs les décideurs publics, au besoin, aidez-nous moins et gardez s'il le faut un peu d'argent pour acheter nos solutions si elles prouvent qu'elles sont compétitives.

S'il m'est donné de formuler un dernier vœu pour cette nouvelle année, ce serait également que l'on se souvienne aussi d'un slogan inventé il y a quelques années pour encourager la consommation citoyenne : « Nos emplettes sont nos emplois » et que l'on nous laisse au moins une chance de les défendre.

Jean-Luc Saouli,
P-DG Lyria

CODE

Novell ouvre AppArmor

L'éditeur Novell annonce l'ouverture du code d'AppArmor. AppArmor avait été acquis par le rachat de la société Immunix en mai 2005. AppArmor est une solution de sécurité système pour Linux. Il protège autant Linux que les applications, des attaques internes et externes. Il permet de définir des politiques de sécurité avec réglages fins sur les accès et les privilèges. L'outil est déjà intégré dans SUSE Linux. Le projet est disponible sur le site OpenSUSE.

Site : <http://www.opensuse.org>

SGBD

4D connecte Oracle 10g

Présenté durant l'Apple Expo de septembre dernier, 4D RAD for Oracle database devait initialement être vendu sous forme packagée. Finalement, cela ne sera certainement pas le cas, du moins pas immédiatement. Il s'agit d'un outil complexe permettant de créer des applications mappant des données de bases Oracle 10g ou de pouvoir utiliser à la fois des bases Oracle et 4D dans une même application. Il s'agit de pouvoir profiter de la souplesse d'utilisation d'un L4G avec un vrai RAD pour créer l'interface et le programme, que ce soit sous Windows ou MacOS X. S'il est possible de créer les requêtes basiques, dès que l'on veut approfondir les manipulations, il faut mettre les mains dans le code... Pour les sociétés intéressées par l'outil, il faut directement contacter 4D. Par ailleurs, l'éditeur annonce la création d'un nouveau centre de formation, qui est une refonte de l'ancien centre existant depuis 1993. Cette nouvelle mouture propose en 2006 pas moins de 34 formations (dont 19 nouvelles), une refonte totale des cours et des supports. À noter, la mise en place d'un site dédié au centre de formation pour plus d'interactivité. Enfin, sur le SGBD 4D, il n'y aura sans doute pas de versions majeures en 2006. Cependant, d'importantes annonces sont prévues durant l'université 4D se déroulant en mai prochain.

Site : <http://formation.4d.fr>

GESTION

Perforce et la configuration

L'éditeur Perforce annonce la disponibilité de la nouvelle version de Perforce, dédiée à la gestion de configuration. Parmi les nouveautés,



l'amélioration de l'interface et surtout une meilleure expérience avec le plug-in WebSphere Studio. D'autre part, l'éditeur annonce le support de Windows Studio 2005 avec le nouveau plug-in Perforce SCC 2005.2. On notera aussi une nouvelle administration simplifiée avec l'utilisation de triggers d'authentification. Et il est maintenant possible d'utiliser dans Perforce un annuaire LDAP ou similaire.

Site : www.netbeans.com

Perforce.

Le système de GCL rapide.

Pour les développeurs qui n'aiment pas attendre.



[Rapide] [Évolutif] [Distribué]

Vous en avez assez d'utiliser un système de gestion de configuration logicielle qui vous empêche de soumettre vos modifications ? Le système de GCL Perforce est très différent : aussi rapide que puissant, aussi convivial qu'efficace. Perforce travaille à votre rythme.

Les performances de Perforce reposent sur trois piliers de conception : une base de données relationnelle indexée avec soin qui vous garantit un temps de réponse rapide pour les petites opérations et un haut débit lorsque les requêtes sont importantes, portant sur plusieurs millions de fichiers ; un protocole de transmission réseau continue des plus efficaces, qui minimise les temps de réponse et exploite au maximum les avantages de la bande passante ; enfin, un modèle de données intelligent centré serveur qui garantit l'incroyable vitesse de fonctionnement de la base de données et du réseau.

C'est à vous de voir. Voulez-vous travailler ou attendre ?

PERFORCE
SOFTWARE

Téléchargez sans conditions une copie gratuite de Perforce sur www.perforce.com. Vous bénéficiez d'une assistance technique gratuite pendant votre période d'évaluation.

Toutes les marques commerciales citées dans le présent document sont des marques commerciales, déposées ou non, de leurs dépositaires respectifs.

Faible WMF : 12 jours d'attente

La découverte d'une vulnérabilité dans le traitement des images au format WMF s'est révélée être la saga sécurité du début d'année. Retour sur les événements majeurs de ce feuilleton sécuritaire.

Lorsque le 27 décembre 2005 une vulnérabilité est annoncée dans l'interprétation des images au format Windows Metafile (WMF), beaucoup d'observateurs pensent à une erreur : le mois précédent, Microsoft avait justement corrigé une vulnérabilité critique qui frappait déjà WMF, et il s'agit donc probablement d'une confusion.

l'application tente de convertir l'image WMF en un format imprimable (EMF), la vulnérabilité est déclenchée.

Les autres vecteurs d'infection (un autre navigateur, le client email, etc...) nécessitent, eux, l'appel direct d'une image au format WMF.

L'occasion était trop belle et les auteurs de codes malicieux n'en sont pas restés là. Après la publication d'un code de démonstration, notamment par le FrSIRT, de nouvelles versions de l'exécutable d'infection étaient découvertes, plus efficaces si l'on en croit l'Internet Storm Center, l'organisme qui a probablement le plus contribué à informer la communauté durant cette affaire.

Très vite, de nouveaux vecteurs d'exploitation sont apparus : une carte de vœux, tout d'abord, qui embarque une image WMF piégée renommée en .JPG pour faire bonne mesure. Puis un ver informatique se propageant via le réseau MSN : il suffit de cliquer sur un lien affiché par le parasite pour lancer Internet Explorer et le diriger vers une page web piégée.

Mais que fait Microsoft ?

Aucun correctif n'étant disponible plus d'une semaine après la découverte de l'alerte, et la communication officielle de Microsoft se limitant au strict minimum (à l'exclusion du blog de ses techniciens), la communauté a dû s'organiser : en relayant massivement le conseil de désactiver une librairie indispensable à l'application Windows Picture & Fax Viewer (shimgvw.dll). Cela aide face à l'infection automatique via Internet Explorer, sans éliminer totalement le risque, car les autres vecteurs d'infection demeurent.



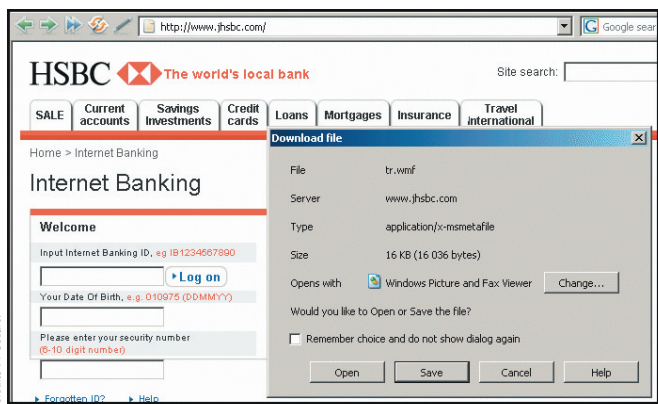
Ilfak Guilfanov, l'auteur d'IDA Pro, a développé le premier correctif efficace contre la faille WMF.

Ilfak Guilfanov, auteur bien connu du désassembleur IDA Pro, développe un correctif destiné à patcher la fonction vulnérable en mémoire. Ilfak injecte un code de son cru dans tous les processus faisant appel à la librairie user32.dll. Celle-ci, en retour, modifie la librairie vulnérable gdi32.dll afin d'interdire l'appel à la fonction SetAbortProc.

Le correctif s'offre en outre le luxe de ne rien toucher au système et d'être facilement désinstallable. C'est un succès immédiat, et le blog d'Ilfak s'effondre sous le poids des téléchargements.

Microsoft finira toutefois par produire un correctif, le 5 janvier. L'éditeur expliquera son retard en invoquant la nécessité de localiser le correctif pour 23 langues et de le tester abondamment. Quelques jours plus tard cependant, deux nouvelles vulnérabilités (non critiques) étaient à nouveau découvertes dans l'interprétation des images WMF.

■ Jérôme SAIZ



Plusieurs jours après la publication du correctif par Microsoft, la faille est toujours exploitée. Un site de phishing imitant la banque HSBC télécharge une image WMF piégée à l'arrivée sur la fausse page d'accueil.

Il s'agit non seulement d'une nouvelle vulnérabilité, mais elle est en outre particulièrement sérieuse. La faille concerne à nouveau le Graphic Rendering Engine de Windows et son interprétation des images au format maison WMF. Concrètement, ces dernières peuvent embarquer à la fois des données vectorielles et bitmap, mais aussi, déclencher l'exécution de code dans le GDI en appelant une séquence d'échappement. Et c'est l'une des fonctions disponibles, SetAbortProc, qui est mise en cause dans cette alerte. Une image au format WMF peut provoquer l'appel de la fonction SetAbortProc, puis exploiter un dépassement de mémoire à l'aide d'une charge offensive qu'elle embarquerait.

Vingt-quatre heures à peine après l'annonce de la découverte, les éditeurs Sunbelt (antispyware) et F-Secure (antivirus) affirment avoir repéré des tentatives d'exploitation de la vulnérabilité. Des sites web malicieux distribueraient des images WMF piégées. Leur interprétation par Windows provoquerait alors l'exploitation de la vulnérabilité WMF et l'installation d'un code malicieux (logiciels publicitaires ou botnet dans un premier temps).

Pour la majorité des utilisateurs de Windows, l'infection est automatique, à cause d'Internet Explorer. Car, bien que ce dernier refuse d'obéir aux appels de fonction vers le GDI lorsqu'il traite une image WMF (il s'agit d'une mesure de sécurité spécifique), la vulnérabilité reste exploitable automatiquement par le biais de l'application Windows Picture & Fax Viewer. La plupart des exploits recensés utilisent ainsi la technique d'une IFRAME afin de passer l'image WMF au Windows Picture & Fax Viewer, plutôt que d'attendre son décodage par le navigateur. Et lorsque

DONNÉES

Des données encore plus réelles

L'éditeur Progress Software vient de dévoiler la v6.3 de son outil ObjectStore. Cette édition inclut le SGBD léger ObjectStore PSE Pro. ObjectStore permet de créer des bases robustes en C++ ou en Java, notamment dans les bases transactionnelles. Les principales améliorations concernent les performances Java, le support de Linux 64-bit en C++ et le compilateur VC 7.1.

Site : www.progress.com

SÉCURITÉ

Une étude troublante ?

Comment mesurer réellement le nombre de failles d'un système ? Faut-il y inclure les applications ? La dernière étude de l'US-Cert montre les limites et le problème des chiffres bruts. Pour 2005, le CERT recense quelques 5 198 vulnérabilités, 812 sous Windows et 2 328 sous Unix et Linux, et 2 058 sur de multiples systèmes. De quoi relancer le débat : quel est le meilleur, question sécurité, Windows ou Linux ? Le problème principal de l'étude est l'aspect fourre-tout, il n'y a pas de classement précis, juste une longue liste. Bien que polémique, cette liste a le mérite de fournir les types de vulnérabilités possibles.

Site : www.us-cert.gov

COACHING

Redynamiser les développeurs !

Vos développeurs sont fatigués ? Ils ont besoin de se remotiver ? De ressembler un groupe ? Pourquoi ne pas faire un « stage » en plein désert ? Loin de tout ou presque, vous pourrez faire de la cohésion d'équipe, du coaching ou encore du management. JMS Médiations propose de tels services, sur mesure, dans différents déserts du monde. Il est toujours bien de reformer les troupes de temps en temps !

Site : www.jmsmediations.com

agenda

FEVRIER 2006

Du 31 janvier au 02 février 2006

Paris La Défense, CNIT

Solutions GNU/Linux 2006

Rendez-vous européen annuel

avec débats, conférences,

séminaires et tutoriels,

pour se plonger au cœur

de l'actualité

des logiciels libres et rencontrer

ses principaux acteurs.

www.solutions.linux.fr

Du 31 janvier au 02 février 2006

Paris La Défense, CNIT

ObjectWeb Con' 06

Cinquième édition de la conférence

mondiale du consortium ObjectWeb.

Nombreuses conférences et sessions

techniques prévues.

<http://objectwebcon06.objectweb.org>

MARS 2006

Du 15 au 16 mars 2006

Paris La Défense, CNIT

Documation 2006

Le rendez-vous annuel de la gestion

de contenu et de document

www.documation.fr

étranger

Du 25 au 26 février 2006

Bruxelles, Free University

FOSDEM 2006

6^e conférence des développeurs européens

des logiciels libres et open source. Événement

majeur du monde libre en Europe !

www.fosdem.org/2006/index

Du 20 au 23 mars 2006

États Unis, Santa Clara

EclipseCon 2006

Conférence entièrement dédiée à Eclipse.

www.eclipsecon.org

3.248 heures pour développer le code...

184 pour trouver ce sacré bug!

142 heures de réunions...

108 pizzas!

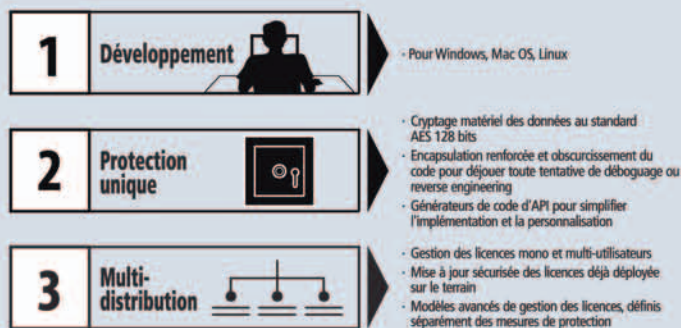
14 week-ends à l'eau!

11 nuits blanches...

Protégez tout d'un seul appel!

HASP
SOFTWARE DRM

En choisissant les solutions de protection de nouvelle génération **HASP**, vous offrez – en une seule opération – une protection ultime à vos créations logicielles tout en bénéficiant d'un environnement simple et transparent pour gérer les options de vos licences.



Demandez votre kit Développeur, ainsi que son guide technique et une clé de démonstration sur www.aladdin.com/hasp

Aladdin
SECURING THE GLOBAL VILLAGE

Amérique du Nord: 1-800-562-2543 International: +972-3-636-2222 UK: +44-1753-622266
 Allemagne: +49-89-894-221-0 Benelux: +31-30-688-0800 France: +33-1-41-37-70-30 Espagne: +34-91-375-99-00
 Israël: +972-3-636-2222 Asie-Pacifique: +852-21-66-8605 Japon: +81-426-60-7191

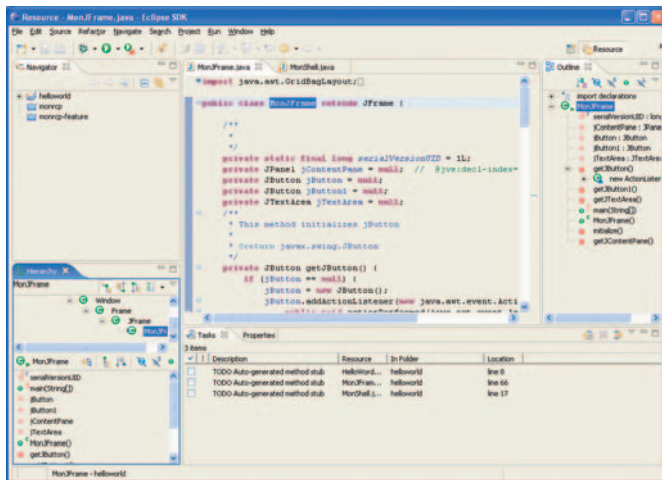
Eclipse vs Netbeans : match nul entre deux champions

Tous deux Open Source et de plus en plus populaires, Eclipse et Netbeans sont des poids lourds parmi les outils de développement Java et largement utilisés. Comparons les.

Comparer Eclipse et Netbeans ? Vaste programme... Disons-le d'entrée de jeu : ils sont formidables tous les deux. Qui, il y a quelques années, aurait pensé (ou même rêvé) disposer d'outils gratuits d'une telle qualité ? Les développeurs de la communauté Open Source aiment la compétition et Eclipse et Netbeans en bénéficient. Par cette comparaison, nous souhaitons simplement montrer en quoi l'un et l'autre se démarquent particulièrement. Nous nous basons sur Eclipse 3.1 et Netbeans 5.0 bêta 2, mais dont la version définitive aura sans doute vu le jour au moment où vous lirez ces lignes.

Premier contact

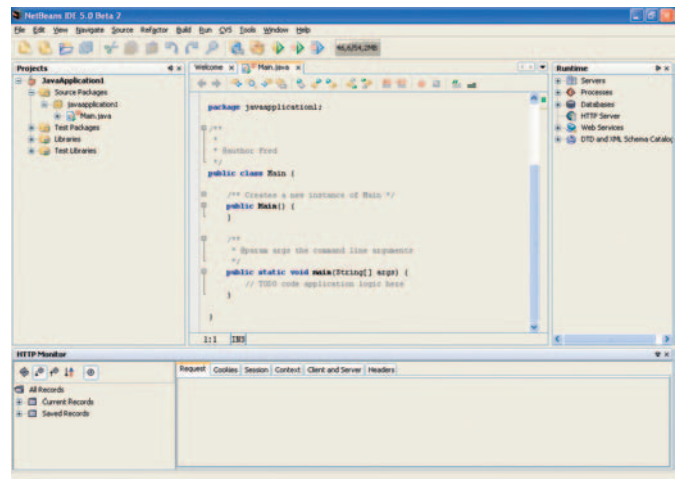
D'aspect, Eclipse est le plus beau. Il est aussi le plus réactif aux sollicitations de l'utilisateur, cela, en vertu de SWT, la librairie graphique native développée spécialement pour lui. L'interface utilisateur est organisée



Vue d'ensemble d'Eclipse.

en perspectives. Une perspective est un arrangement d'un ensemble de vues (ou fenêtres) dont le point central est un éditeur. Les autres vues de la perspective reflètent différents aspects de ce qui est édité dans l'éditeur central. L'utilisateur bascule d'une perspective vers une autre, selon les besoins : ressources, débogueur, etc.

Netbeans est de facture et d'organisation plus classique. Pas de notion de perspective, mais des fenêtres arrangées à la convenance de l'utilisateur. Netbeans est écrit avec Swing, ce qui selon nous nuit à son esthétique, surtout sous Linux. Toutefois, ce point devrait s'améliorer automatiquement avec Java 6.0, alias Mustang. Netbeans est aussi un petit peu plus lent à réagir aux actions de l'utilisateur, mais rien de bien



Vue d'ensemble de Netbeans.

gênant. En raison de sa facture plus classique, l'interface de Netbeans est peut-être plus intuitive que celle d'Eclipse, du moins durant les premiers temps d'utilisation.

Sous le capot

Avant d'être un outil de développement Java, Eclipse se veut un outil universel et extensible par des plug-ins. C'est à la fois son avantage et son inconvénient. Son concept de points d'extension fait toute sa valeur à ce niveau. En revanche, pour utiliser pleinement Eclipse en production il faut se mettre en quête de plug-ins, libres ou commerciaux. Bien évidemment, la SDK Java, un plug-in d'Eclipse, permet déjà de travailler, mais on est vite limité, surtout pour un projet d'entreprise.

De facture plus classique, Netbeans est lui aussi extensible au moyen de plug-ins, mais ne se présente pas comme universel, il est centré sur le développement Java et offre dès le départ une solution et une palette d'outils plus complète. En revanche, lorsqu'il s'agit d'étendre les fonctionnalités de l'IDE par plug-ins, Netbeans se montre moins docile.

La gestion des projets

Eclipse utilise la notion d'espaces de travail (Workspace) dans lesquels sont rangés les projets. Projets, ouverts ou fermés, d'un même espace de travail sont visualisés simultanément. Eclipse gère les métadonnées du projet de façon spécifique, ce qui implique qu'il n'est pas vraiment prévu que du code écrit sous Eclipse soit compilé en dehors d'Eclipse. Eclipse peut utiliser ANT pour construire un projet, mais ce n'est pas son

Innovations par InterSystems



Développement Rapide sur base d'Objets Robustes



Vitesse foudroyante grâce à un moteur multidimensionnel



Administration facile



Scalabilité massive sur matériel minimum

Plus de Mapping Objet/Relationnel.

Caché est le premier SGBD multidimensionnel qui gère avec la même efficacité le traitement transactionnel et le décisionnel en temps réel. Sa technologie post-relationnelle combine la fiabilité des technologies Objet et SQL, et élimine ainsi tout mapping Objet/Relationnel. Il offre une scalabilité maximum sur un matériel minimum, requiert peu d'administration, et inclut en outre un environnement de développement rapide d'applications.

Ces innovations se traduisent par une introduction plus rapide de vos applications sur le marché, des coûts opérationnels plus réduits et des applications plus performantes. Parce que nous avons une pleine confiance dans nos affirmations, nous vous offrons une garantie de remboursement: *Achetez Caché pour un nouveau développement et si endéans l'année, vous n'êtes pas satisfaits, pour quelle que raison que ce soit, nous vous rembourserons votre licence.** Caché est disponible sur UNIX, Linux, Windows, Mac OS X, et OpenVMS – et est déployé sur plus de 100.000 systèmes supportant de 2 à 50.000 utilisateurs. Nous sommes InterSystems, une société technologique globale reconnue pour ses innovations depuis plus de 25 ans.



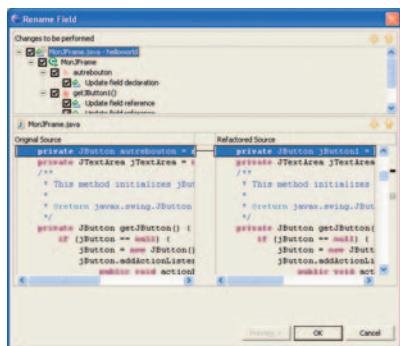
Testez gratuitement un SGBD innovateur : téléchargez une version fonctionnellement complète et illimitée dans le temps de Caché ou demandez votre CD sur : www.InterSystems.fr/Mapping

comportement par défaut. En outre nous trouvons l'importation de code ou de fichiers pré-existants assez pénible.

Netbeans ne connaît pas la notion d'espace de travail, mais il sait ouvrir plusieurs projets simultanément. L'importation de sources pré-existantes est plus aisée. Il y a même des assistants pour créer un des projets autour de l'existant. Tout ce qui concerne les projets, y compris les méta-données, est rangé dans des scripts ANT, ce qui est un point fort et même très fort, car cela signifie qu'il est possible de compiler en dehors de Netbeans exactement de la même façon qu'en dedans.

Écrire du code

Développer en Java signifie, a priori, écrire aussi avec d'autres langages que Java. HTML et XML notamment. En ce qui concerne ce dernier, Eclipse est assez pauvre, même pas de coloration syntaxique pour XML, tant



Eclipse permet de visualiser les modifications avant un refactoring.



Netbeans affiche la Javadoc dans la fonctionnalité d'achèvement de code.

l'option sélectionnée dans la liste de possibilités par une fenêtre affichant la Javadoc avec des liens actifs.

Sur les concepteurs visuels, car on peut désormais considérer qu'Eclipse en est (enfin) pourvu, même s'il faut télécharger le plug-in séparément sur le site officiel. Le concepteur d'Eclipse permet de travailler avec SWT ou Swing et dérive de celui de Visual Age pour Java. Il se présente comme une fenêtre splittée, la partie du haut montre le concepteur visuel proprement dit, tandis que la partie du bas montre le code. Toute modification dans l'une est répercutée dans l'autre. Ce concepteur est solide, mais à l'usage, nous ne le trouvons pas très ergonomique. La customisation du GridBagLayout devrait être améliorée. Enfin, le composant en

cours de conception apparaît sur la barre des tâches du système, ce qui est parfois agaçant.

De son côté, le nouveau concepteur visuel de Netbeans, Matisse, est un modèle du genre. Pas de fenêtre splittée ici, mais deux éditeurs parallèles. On bascule de l'un à l'autre avec la souris ou un raccourci clavier. Selon nous, ce concepteur visuel, la customisation du GridBagLayout en particulier, est tout simplement le meilleur du marché.

Plug-in et plate-forme Client Riche

Lorsqu'il s'agit d'écrire un plug-in, c'est Eclipse qui se révèle le plus intéressant, pour plusieurs raisons. D'abord son mécanisme de points d'extensions. Ensuite, une documentation intégrée claire. Deux choses qui rendent (relativement) facile la tâche de bien comprendre ce qu'il faut faire d'abord et de passer à la réalisation ensuite. Du côté de Netbeans, c'est moins immédiat, car l'outil n'intègre pas, ou peut-être n'intègre pas encore, une documentation riche et claire sur le sujet. Et il est difficile de s'y retrouver dans une API gigantesque, comme l'est celle d'Eclipse d'ailleurs.

Une application client riche est une application qui s'appuie sur le runtime d'Eclipse ou de Netbeans, amputée des plug-ins inutiles et éventuellement augmentée d'autres plug-ins. Fondamentalement, une application client riche est avant tout un plug-in. Pour cette raison, on comprend que l'écriture de clients riches est plus aisée sous Eclipse. Par ailleurs, SWT permet de fournir un look natif. Cerise sur le gâteau, le SWT d'Eclipse 3.2 intègre des fonctionnalités de dessin avancées, basées sur le moteur Cairo Vector de GTK et Motif, GDI+ de Windows. Petit bémol, SWT oblige le programmeur Java à gérer manuellement les ressources. Si faire du client riche sous Netbeans est moins agréable, il y a toutefois un petit dièse qui est l'emploi de Swing, ce qui dispense le programmeur Java de s'inquiéter de la gestion des ressources.

Côté entreprise

Tant qu'il n'est pas enrichi par des plug-ins, Eclipse ne permet pas de développer pour l'entreprise. D'un autre côté, il est vrai que les plug-ins disponibles sont nombreux. Netbeans permet de développer directement pour l'entreprise : JSP, Web Services, JSF, Struts, EJB. Netbeans embarque Tomcat et connaît les serveurs d'applications de Sun, avec JBoss et BEA Weblogic. Cela signifie qu'il est possible d'y déployer un EJB en un clic de souris. Nous apprécions aussi beaucoup le moniteur HTML et la fenêtre Runtime qui visualise de façon centralisée, serveurs, pilotes de base de données, DTD, etc.

En guise de conclusion

Choisir entre ces deux outils est difficile. Si l'on a besoin d'un environnement adaptable à un développement particulier, ou si l'on souhaite écrire des applications client riche, le choix d'Eclipse semble s'imposer. Si l'on préfère un outil complet et surtout, tout prêt pour développer des applications d'entreprise, ou encore si l'on pense avoir besoin de compiler en dehors de l'EDI, on se tournera vers Netbeans. Mais si les besoins sont plus nuancés, alors le choix est plus difficile entre deux outils vraiment excellents. Peut-être alors choisira-t-on simplement celui avec lequel on se sent le plus en affinité ?

■ Frédéric Mazué fmazue@programmez.com

MACHINE

Apple dévoile ses armes

Durant la keynote de la dernière MacWorld, Steve Jobs, le patron d'Apple, a présenté les premiers modèles de Macintosh utilisant les processeurs Intel. Le premier est l'iMac. Il utilise un processeur double cœur, tout comme le futur MacBook Pro (ex. gamme PowerBook). Les machines semblent prometteuses, même si ce n'est qu'un début. Steve Jobs a par ailleurs annoncé le support total de MacOS X (la v10.4.4) des Mac Intel. De plus, une nouvelle génération d'applications apparaît : les applications universelles. Elles sont capables de fonctionner sur processeurs Intel et PowerPC. De nombreuses applications sont en cours de portage. Les applications professionnelles d'Apple le seront dans quelques semaines. Pour les applications non universelles, elles fonctionneront, via la technologie Rosetta, permettant de traduire à la volée les instructions PowerPC en instructions Intel. Par ailleurs, Intel proposera rapidement des outils adaptés aux nouvelles machines Apple. Côté logiciel, la sensation est venue de l'outil iWeb. Il s'agit d'un créateur de site internet, très simple d'utilisation, dans le lequel le glisser-déposer est roi. Il s'intègre aux autres outils de la gamme iLife tels que iMovie ou iPhoto. Il s'agit de pouvoir mettre en ligne du contenu multimédia très rapidement. Pour l'interface, on dispose de plusieurs modèles. La puissance fonctionnelle est impressionnante, notamment dans les fonctions graphiques. Il dispose d'un moteur de blog pour créer des blogs rapidement. Il est aussi possible de créer facilement des podcast et de mettre en place des flux RSS. L'outil utilise Ajax pour l'interface des sites (ex. : le slideshow). Seul

regret, il ne fonctionne que sur MacOS X ! Par contre, rien sur le prochain MacOS X prévu fin 2006 ou courant 2007. Il faudra sans doute attendre la prochaine conférence des développeurs Apple, courant juin.

Site : apple.com

SÉCURITÉ

Aladdin et les virus

L'éditeur Aladdin vient de rejoindre le Microsoft Virus Information Alliance. Il s'agit d'un consortium regroupant différents éditeurs de la sécurité et qui partage

les informations sur le domaine de l'antivirus et les dernières menaces. L'objectif étant que Microsoft alerte rapidement ses clients sur les risques et les méthodes de lutte. « Aladdin est fier de collaborer avec Microsoft en tant que membre de l'alliance VIA. Notre équipe de surveillance de contenus CSRT (Content Security Response Team) sera activement impliquée dans le partage d'informations relatives à la propagation et à l'impact des nouveaux virus, » commente Shimon Gruper, Vice-Président des technologies de l'unité Aladdin eSafe.



**Nous
développons
des langages,
pas des
discours !**

**Aux longs développements
dans les médias, nous préférons
le développement efficace
pour nos clients.**

**Développons de nouvelles relations.
Rencontrons-nous !**



Nat System
Stratégiquement vôtre

Nat System - Immeuble Le Péricentre
16 av. Louison Bobet - 94120 Fontenay sous Bois
Tél. : 01 45 14 73 73 - www.natsystem.fr

Réussir avec L'OPEN SOURCE

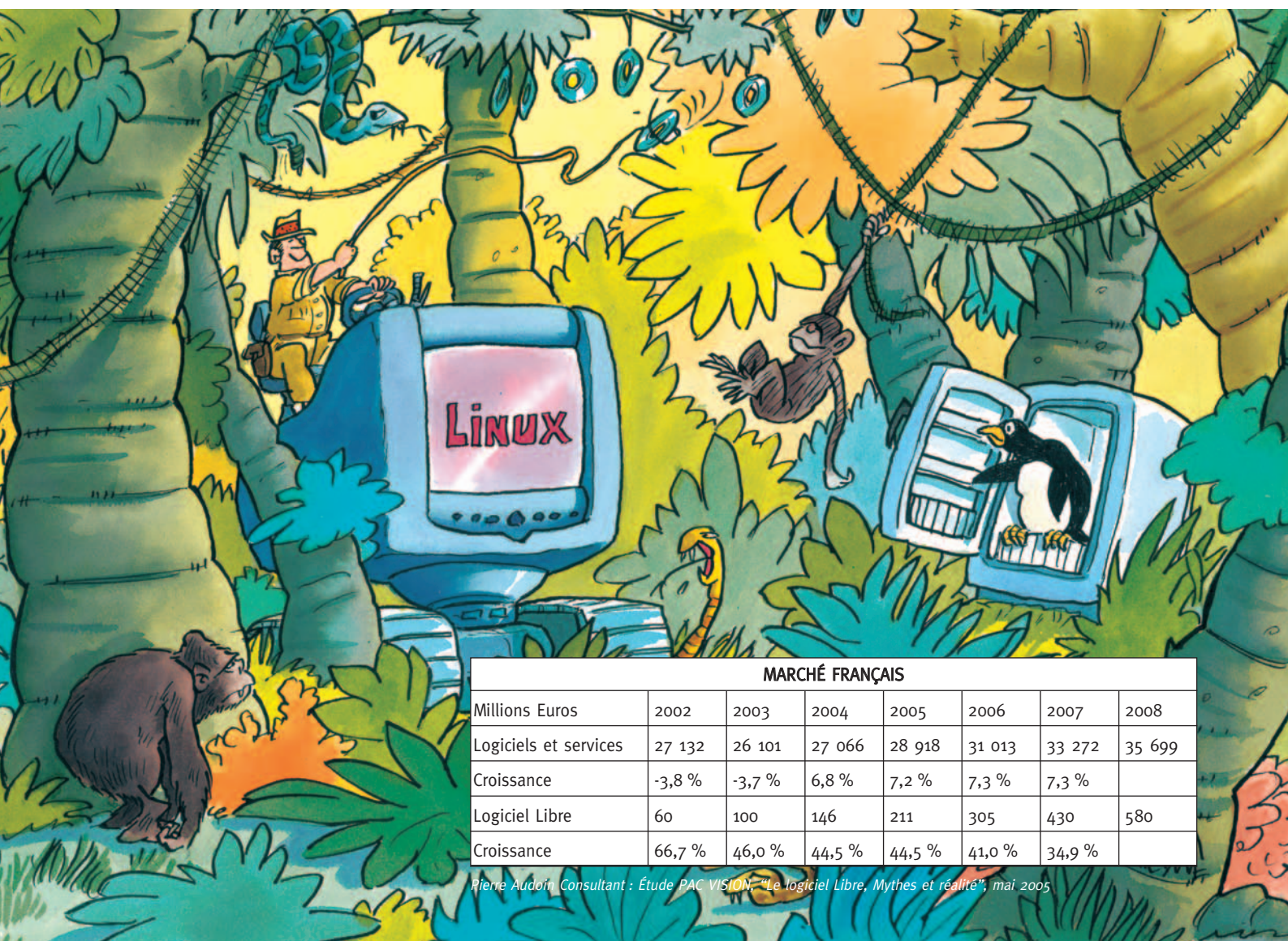
Fini le cloisonnement logiciels ouverts / propriétaires. Désormais, le marché se partage très clairement entre les deux types de logiciels. Il y a à peine 2 ans, les partisans des logiciels ouverts fustigeaient le monde propriétaire / commercial, et inversement. En moins de 24 mois, la situation a radicalement changé. Cette dualité se voit chez les utilisateurs et les éditeurs. Les éditeurs propriétaires n'hésitent plus à utiliser de l'Open Source et les éditeurs libres passent accord sur accord avec les acteurs du monde

propriétaire. La question n'est plus de savoir si je dois utiliser de l'Open Source, mais comment le faire fonctionner avec son existant !

Si le marché se clarifie, l'Open Source demeure en partie une jungle : multiplication des licences, foisonnement des projets ouverts, manque de briques communes entre les différentes distributions Linux, etc. La licence constitue un élément crucial pour une entreprise, un éditeur qui souhaite vendre une solution utilisant de l'Open Source. Que risque-t-on ?

Quelles sont les clauses de la licence utilisée ? Les licences sont-elles compatibles entre elles ? Faut-il avoir peur de la double licence ? Les questions demeurent nombreuses. Dans ce grand dossier, nous allons essayer de faire le point sur le marché et la vision des éditeurs et spécialistes de l'Open Source. Nous vous donnerons les clés pour choisir votre distribution Linux. Vous saurez s'il faut ou non se former aux outils open source pour trouver un emploi.

■ François Tonic



MARCHÉ FRANÇAIS							
Millions Euros	2002	2003	2004	2005	2006	2007	2008
Logiciels et services	27 132	26 101	27 066	28 918	31 013	33 272	35 699
Croissance	-3,8 %	-3,7 %	6,8 %	7,2 %	7,3 %	7,3 %	
Logiciel Libre	60	100	146	211	305	430	580
Croissance	66,7 %	46,0 %	44,5 %	44,5 %	41,0 %	34,9 %	

Pierre Audoin Consultant : Étude PAC VISION, "Le logiciel Libre, Mythes et réalité", mai 2005

Le marché open source / Linux : dynamique et pragmatique

Désormais, Open Source, Logiciel Libre, sont rentrés dans les mœurs de nombreux développeurs et entreprises. Dans une moindre mesure, le système Linux connaît le même impact même si, dans ce cas précis, il se cantonne encore à quelques sections particulières, comme le serveur ou le poste utilisateur en entreprise. La tendance du mixage open source / libre et logiciel commercial s'impose sur de nombreux ordinateurs.

Le logiciel libre, ou open source n'est pas synonyme d'installation de Linux. Les deux sont indépendants dans l'approche de leur utilisation. Un DSI peut réduire la charge licencing en restant sous Windows, mais en utilisant des outils libres / ouverts. La moitié des utilisateurs des solutions Zend ou JBoss utilise Windows.

De plus, une des clés du succès des logiciels ouverts est justement d'être disponibles sur de nombreuses plates-formes. Firefox ou OpenOffice n'auraient, sans doute, jamais eu un succès aussi rapide et aussi fulgurant sans un portage sous Windows. Pour certains puristes, le fait de porter des logiciels libres sur Windows conforte le système Microsoft et le pérennise, ce qui n'était pas l'objectif au départ. Dans tous les cas l'un des intérêts du logiciel libre / ouvert est de laisser le choix aux utilisateurs.

75 % du web avec de l'Open Source

Fin novembre dernier, le cabinet Gartner publiait une étude sur les logiciels open source et le marché logiciel traditionnel. À l'horizon 2010, l'analyste prédit 80 % de logiciels open source dans les investissements d'infrastructure et 25 % dans les outils business. 75 % des applications web (incluant système, middleware, outils de développement) seront réalisées avec des briques ouvertes. Enfin, les éditeurs n'ayant pas implémenté des outils, composants open source, prennent le risque de devenir non concurrentiels. En 2006, Gartner pense que 75 % des outils open source seront déployés dans le cadre de nouveaux projets. Cela signifie que tout le legacy, souvent créé sur des outils / technologies propriétaires demeurera quasiment inchangé pour des raisons de coût et de temps de migration.

Gartner définit un logiciel open source ainsi : tout logiciel sous licence reconnue par l' Open Source Initiative. Ce logiciel peut être modifié et redistribué par l'utilisateur sous la licence originale. De nombreuses licences répondent aux critères de l'Open Source Initiative. Un logiciel open source offre donc plus que le simple fait de disposer du code source. Et il est notable que les revenus générés par l'open source pour les éditeurs / prestataires viennent majoritairement des services et de la maintenance.

Selon le Gartner, l'open source va modifier le logiciel traditionnel. Il distingue trois axes :



Sacha Labourey, JBoss

“La moitié des développeurs sous Windows”

En quelques années, JBoss est devenu un redoutable concurrent dans les serveurs d'application et maintenant dans le middleware. L'outil libre a en face de lui des poids lourds comme BEA et IBM. Même sur le marché libre, JBoss doit faire face à une multiplication de projets : Red Hat, ObjectWeb, Apache.

Aujourd'hui, le pragmatisme selon Sacha Labourey de JBoss est devenu le credo premier. « Il était stupide de dire : « je n'utilise que des outils libres ou je n'utilise que des outils commerciaux ». Les utilisateurs ne pouvaient en faire une religion. Il faut accepter le monde hétérogène. Nous avons passé des accords avec HP, Sun et maintenant Microsoft, cela correspond à la réalité du marché. Par exemple, la moitié des développeurs JBoss utilisent le système Windows. C'est tout l'intérêt pour un développeur de travailler avec des outils et des langages reposant sur une machine virtuelle. Historiquement, Java sur Windows était plus simple que sur Linux. »

L'accord entre JBoss et Microsoft a de quoi surprendre, mais finalement, cela rejoint le pragmatisme du marché. Le marché se partage entre outils libres et commerciaux. Ce serait, en quelque sorte, l'environnement optimal de pouvoir utiliser au moins les deux mondes. « Nous déplaçons la valeur de la licence, pour les logiciels de masse, vers les souscriptions et les services ». Ce modèle de services et de souscriptions semble être devenu une règle chez les éditeurs d'outils libres / open source. Red Hat constitue un parfait exemple en ce domaine. ■ F.T.

Eric Mahé, Sun "Le logiciel libre, un axe central"

2005 fut une grande année pour Sun et le logiciel: nous avons fêté les 10 ans de Java, l'arrivée des versions StarOffice 8.0 et de OpenOffice 2.0, le démarrage des projets OpenSolaris et GlassFish, la validation par l'OASIS du format Open Document et l'annonce de l'accès gratuit à l'offre Solaris Enterprise Systems. Quel est le point commun à tous ces projets? L'accès au code source, même si tous ne sont pas dans l'Open Source, qui est une constante de la stratégie de Sun depuis plus de 20 ans. Et pourquoi une telle actualité en 2005? Car nous sommes persuadés que l'industrie du logiciel sera désormais basée sur trois piliers: la transparence de l'accès au code, l'importance fondamentale des communautés et la disparition du prix à la licence. Le logiciel libre est donc devenu, en 2005, l'axe central de notre stratégie, en nous engageant – dans les limites de ce que les contraintes techniques ou légales nous permettent – à mettre à terme la totalité de notre offre dans l'Open Source. Il s'agit d'une tendance qui concerne l'ensemble des économies que la mondialisation tend à rendre de plus en plus globales: Sun n'aborde pas, par exemple, un marché comme la Chine, avec les mêmes méthodes que celles en vigueur dans d'autres pays. L'Open Source est à ce titre le plus fort symbole de la transformation que l'industrie de l'informatique est en train de négocier et qui était en gestation depuis des années: le passage d'une économie de licence à une économie de services. 2006 devrait donc voir comment ces nouveaux modèles vont être mis en œuvre, à la fois par les éditeurs de logiciel et leurs clients. A l'image de ce que Sun a déjà réalisé avec Java, nous pensons que les marchés les plus prometteurs sont ceux que chaque acteur est capable de créer (convergence IP, Web 2.0, mobilité, sécurité, gestion des identités), et l'Open Source fait déjà partie intégrante de la croissance de ces nouveaux secteurs. ■



offrir une alternative peu chère ou totalement gratuite et l'open source a le potentiel pour éroder les revenus des éditeurs provenant de la vente de licences.

les logiciels open source vont créer de nouveaux modèles économiques liés à la souscription, à la maintenance, aux services, aux supports, à l'administration, etc.

les logiciels open source vont accroître la dépense globale de logiciels et donc accroître le marché, y compris sur les marchés non ciblés par les éditeurs commerciaux.

De nouveaux marchés vont donc s'ouvrir aux logiciels libres, tels que les outils de CRM. Les PME / PMI seront aussi des marchés de choix, surtout si leurs prestataires se forment et fournissent des prestations autour des logiciels open source. Au niveau des éditeurs, l'open source fournit une telle pression que les éditeurs ne pouvant pas innover, évoluer et four-

nir des solutions de qualité et de la valeur ajoutée, auront du mal à survivre. Chaque éditeur doit donc appréhender l'open source sur son marché et en évaluer l'impact, les risques et pourquoi pas, comment l'implémenter à son offre. Ainsi, Gartner se risque à une estimation sur le nerf de la guerre: les revenus du marché des logiciels. Actuellement, 95 % des revenus sont générés par les éditeurs commerciaux, et 3 % par l'open source. En 2009, l'open source passe à 11 % et les commerciaux descendent à 80 %.

Très clairement, dans ses recommandations, Gartner préconise aux entreprises d'établir un plan d'implémentation de composants open source dans les développements, pour réduire les coûts et être en phase avec le « time to market », ne pas hésiter à tester les logiciels ouverts et, le cas échéant, à investir les services professionnels pour le support des composants ouverts. Pour les éditeurs et SSII, les cabinets conseillent de créer des services autour de l'open source, investir dans des partenariats avec des spécialistes de l'open source, notamment dans les marchés émergents.

Bref, pas de révolution, pas de grand chambardement, mais une confirmation que l'on va vers un marché hétérogène où les outils ouverts et commerciaux vont cohabiter et vivre, de gré ou de force. Les éditeurs vont devoir, soit intégrer des composants ouverts, soit prouver leur capacité à innover et à apporter une réelle valeur ajoutée par rapport aux concurrents ouverts.

■ François Tonic

Gilles Lunzenfichter, Directeur Marketing Europe du Sud, Novell

Le 'Mixed Source': Nouveau Paradigme Logiciel

Pour Novell, le futur du logiciel est à la fois ouvert et propriétaire, et les éditeurs les plus performants seront ceux qui seront partisans et supporters des deux camps. Les clients accordent une attention particulière à la valeur ajoutée, aux coûts, aux délais de mise en œuvre, à l'innovation et aux standards ouverts, quel que soit le fournisseur. Les éditeurs qui s'engagent dans une voie de solutions et de produits issus d'un fournisseur unique ne seront plus écoutés par les clients. Beaucoup d'entreprises songent ainsi à une

migration d'Unix vers Linux afin de réduire leurs coûts jusqu'à 75% sur ces machines. Les logiciels open source rattrapent de manière rapide les offres propriétaires au niveau des systèmes d'exploitation, des serveurs d'applications, des bases de données entre autres. Mais au lieu de vouloir faire table rase des solutions propriétaires, l'avancée des solutions open source crée de nouvelles opportunités aussi bien pour les éditeurs de logiciels propriétaires que pour la communauté open source. Ce contexte favorise les éditeurs ayant un ■ ■ ■

Pour les initiés.

Vous désirez améliorer les performances de votre application? Vous voulez migrer plus rapidement vers le 64 bits ou passer au multi-cœur avant vos concurrents? Si seulement vous connaissiez quelqu'un d'initié à ces nouveaux matériels. Quelqu'un qui les connaisse vraiment. Il s'avère que OUI, vous connaissez un tel quelqu'un.

Outils de Développement Intel®. Des outils logiciels par les fabricants du cœur du système.

“Le nouveau compilateur Intel® version 9.0 a démontré dans tous les tests Polyhedron Benchmarks, sans avoir modifié les programmes, des améliorations importantes de vitesse d'exécution. La version 64 bits a tourné d'une façon très satisfaisante, même sur un processeur non Intel.”

John Appleyard, Polyhedron*

Performances

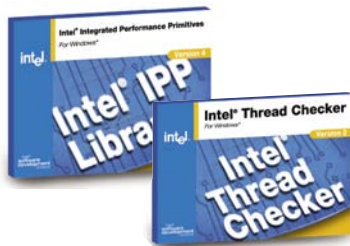
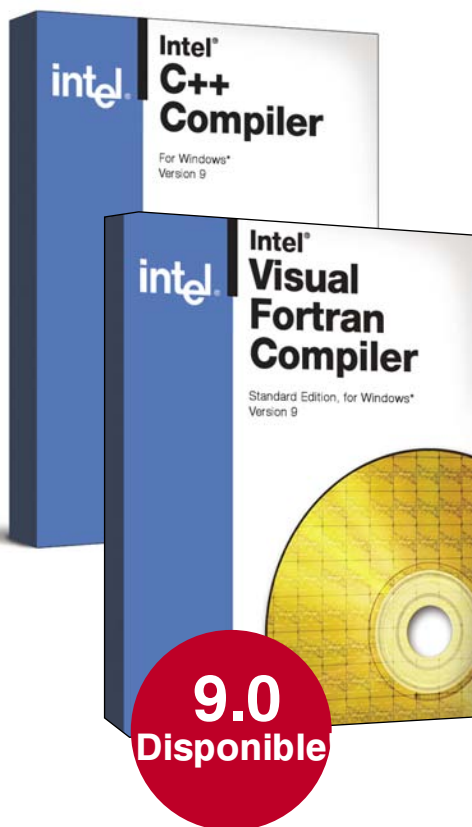
Les outils de développement Intel® dotent les plus grands développeurs d'une connaissance d'initié aux plates-formes informatiques hautes performances. Leur quête vers des niveaux de performances exceptionnelles est simplifiée grâce à des caractéristiques comme l'auto-parallélisation, OpenMP, des mises en disponibilité débogables, et autres.

Compatibilité

- Compatible avec les outils de développement que la plupart des grands programmeurs d'applications Windows* utilise. Compatible source et binaire avec Microsoft Visual C++ 6.0 et .NET.
- Compatible source et objet à partir de la version GCC 3.2 et postérieure, et autres utilitaires largement utilisés dans les chaînes de développement Linux. Intel C++ Compiler est en plus également conforme à la norme C++ ABI.
- Un ensemble puissant qui réunit le frontal de Compaq Visual Fortran (CVF) avec le moteur Intel® Fortran, et qui est doté d'une compatibilité de code source non négligeable.

Support

Le support Intel® Premier Support et les mises à jour sont inclus pendant un an.



Intel® Thread Checker 2.1

Détecte des bogues multi-filaires, des conditions d'échappement, des blocages par verrouillages et autres.

Intel® Integrated Performance Primitives Library 4.1

Code très optimisé pour des applications graphiques, le multi-média, les maths et le traitement de signal.

Pour les compilateurs et bibliothèques Intel® un seul numéro: 01 30 82 04 54

Prix, disponibilité, service chez **microsigma**, demandez Valérie ou allez sur www.microsigma.fr/intel

© 2005 Intel Corporation Intel, le logo Intel, Pentium, Itanium, Intel Xeon et VTune sont des marques de fabrique ou enregistrées de Intel Corporation ou de ses filiales aux Etats Unis d'Amérique ou autres pays. *D'autres noms ou marques appartiennent à leurs propriétaires respectifs.

Alexandre Zapolsky, PDG de Linagora

"400 entreprises et 5 000 emplois créés en moins de 3 ans ! Qui dit mieux ?"



Le marché de LINUX et des Logiciels Libres est certainement celui qui connaît la plus grande vitalité. Avant 2000, le marché professionnel de l'informatique Libre n'existait pas. A ce moment là se créent les premières Sociétés de Services en Logiciels Libres ou SS2L. Et puis ce sera, avec l'éclosion de la bulle Internet, quelques années noires pour ces pionnières de l'Informatique Libre. Début 2003, seule une poignée d'acteurs historiques est encore présente. Depuis c'est l'engouement ! Nous totalisons en moins de 3 ans la création de plus de 400 SS2L en France. Ce phénomène est certainement aussi important que celui que nous avons connu avec le développement d'Internet. Nous estimons que le marché en France du Logiciel Libre pèse plus de 400 M €

en 2005. L'année 2005 marque le début des grands projets d'industrialisation de solutions libres. Ainsi, une grande Administration française a mis en place un marché de près de 40 M € pour le support, la maintenance et les prestations associées (expertise, aide à la migration, maintenance évolutive, etc.) sur plus de 100 Logiciels Libres. Ce marché inaugure les grands projets de migration. L'autre tendance lourde du marché du Logiciel Libre concerne le rattrapage du secteur privé par rapport au secteur public. Si le public était majoritaire au niveau de l'investissement en informatique libre jusqu'à présent, 2005 marque le dépassement du public par le privé.

Enfin, d'un point de vue des technologies utilisées, nous avons pu assister en 2005 à la mise en place des grandes migrations sur le Poste de Travail (1/6 des postes de l'Administration Centrale tourne avec Open office), la mise en œuvre des premières solutions de Groupware à grande échelle (55 000 utilisateurs), le développement de nouveaux outils d'ECM (Enterprise Content Management) comme Alfresco. 2006 devrait être l'année des consolidations sur le marché des SS2L. Nous verrons alors apparaître les grandes Sociétés de Services en Logiciels Libres de demain, celles qui occuperont la place de leader Français... et pourquoi pas celle de leader Européen ! ■

■ ■ ■ pied dans les deux mondes : des fournisseurs bien ancrés au sein de la communauté open source, mais cependant capables de fournir et supporter des solutions propriétaires. Ce nouveau paradigme reflète parfaitement le positionnement de Novell puisque la société propose à ses clients le meilleur des solutions propriétaires et de l'infrastructure open source et fournit le support et les services adaptés aux deux modèles. Les distributions Linux concurrentes poursuivent avec un glissement une approche 100 % open source et



les éditeurs de logiciels propriétaires s'évertuent de manière vaine à combattre les avancées de la vague open source. Ces deux modèles échoueront individuellement s'ils n'intègrent pas l'approche de l'autre à leur propre modèle. Enfin, Novell est impliqué dans le support et le développement de projets open source, en donnant aux clients une meilleure perspective des avancées possibles à réaliser sur le court ou le long terme avec l'open source ou avec les solutions propriétaires. ■

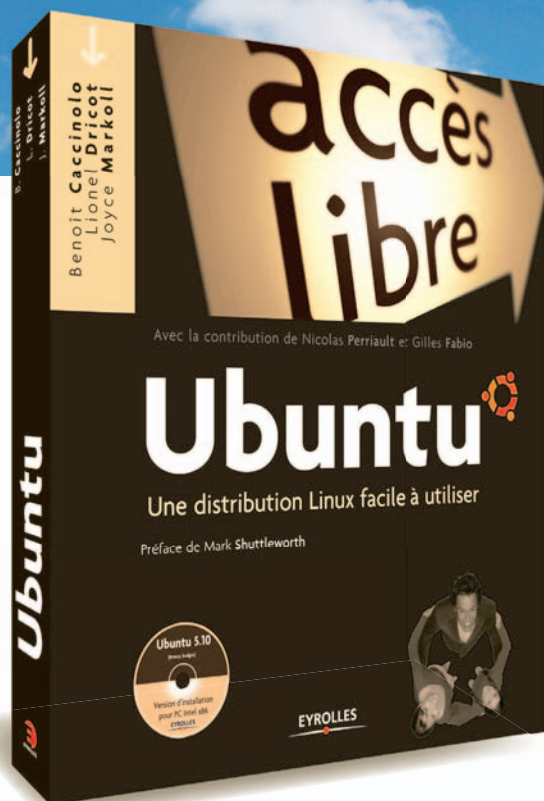
Cyril Pierre de Geyer, Anaska

"Un marché européen soutenu"



En Europe, le marché du logiciel libre se porte bien. Cette forme est soutenue par les clients (en France principalement les administrations) et par les éditeurs/acteurs traditionnels (Intel, IBM, SAP, ...) qui investissent dans les acteurs du libre (Zend, MySQL, Mandriva, ...).

Le logiciel libre est une bonne opportunité pour l'Europe qui ne dispose que de très peu de gros éditeurs. Et dans ce cas, l'exemple de Mandriva est intéressant, car nombre de ses clients sont des clients qui ne veulent pas travailler avec un éditeur américain. Les tendances sont à la hausse. Nous sommes encore dans un marché de niche qui est en phase de forte croissance. Si l'administration continue à pousser le libre (ce qui est le cas, si j'en crois les gens de l'ex ADAE), 2006 sera un premier pas vers la démocratisation du logiciel libre. Mais je pense que la phase d'industrialisation n'aura pas lieu avant 2008-2010. Ce qui me marque actuellement, c'est le portage du modèle classique vers le modèle libre. Par exemple, l'intérêt croissant des certifications. D'autre part, il y a de nombreux outils commerciaux qui permettent de travailler avec des outils libres. Ces outils ne sont pas indispensables, mais peuvent apporter des gains de performances. Le logiciel libre n'est pas complètement opposé au modèle propriétaire, il convient de chercher et de prendre ce qui est bon chez chacun. ■



300 pages - 15 €
CD-Rom avec Ubuntu 5.10 Breezy Badger

Eyrolles

Le "libre" à la portée de tous

- **Un guide complet pour installer et utiliser rapidement Ubuntu, une distribution Linux fiable et conviviale.**

“ Ubuntu est un projet libre dont l'ambition est d'offrir une distribution Linux complète, disponible partout, pour tous et librement. Le mot Ubuntu exprime que toute aide apportée à autrui rejaille nécessairement de façon bénéfique sur tous. C'est l'exacte description du cercle vertueux de l'Open Source – nous partageons et utilisons du code, et les progrès que nous y apportons profitent à tous.

Préface de Mark Shuttleworth
Fondateur d'Ubuntu

Nouveau dans la collection Accès libre



OpenOffice.org 2 efficace

Sophie Gautier et les chefs de file du projet fr.OpenOffice.org vous aideront ici à maîtriser la suite OpenOffice.org 2.

400 pages - 29 € - **CD-Rom avec OpenOffice.org 2.0**

Débuter sous Linux avec Mandriva

Le guide idéal pour passer à Linux sans difficulté, de la découverte du système aux outils d'administration Linux.

500 pages - 32 € - **CD-Rom avec Mandriva Free**

Gimp 2 efficace

Découvrez et utilisez Gimp 2 ! Cet ouvrage de référence détaille l'ensemble du logiciel : retouche photo, dessin, impression, publication web...

360 pages - 29 € - **CD-Rom avec Gimp 2.2.7**

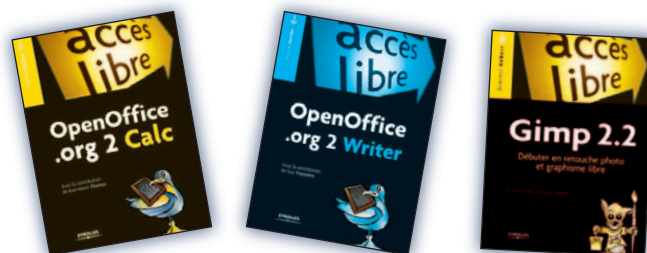
Les Poches Accès libre

Des guides de prise en main pour bien débuter et gagner rapidement en efficacité.

OpenOffice.org 2 Calc - 202 pages - 12 €

OpenOffice.org 2 Writer - 232 pages - 12 €

Gimp 2.2 - 298 pages - 12 €



Alexandre Vasseur, Senior Technologist, BEA System "L'avenir est dans le mélange des genres, l'approche « blended » !"

L'année 2005 a été riche en événements pour l'Open Source. Plusieurs auteurs de projets clés ont fondé leur société sur un modèle de service (support, formation, expertise) : Spring, Axis, Maven. De nouveaux modèles économiques ont émergé pour évaluer la pérennité des projets Open Source et fournir le support et la certification des applications dont les développeurs ont besoin. SpikeSource, Virtus ou Covalent en sont des exemples. L'Open Source est ainsi de plus en plus fédérateur de marchés de niche et de strates applicatives variées. On a pu vérifier ce phénomène de consolidation dans le monde Linux, il y a 2 ans. Sur 1000 contributeurs à la communauté Linux, 100 seulement contribuaient à 95% des changements. Cette frange active de la communauté rassemble d'ailleurs le plus souvent des développeurs employés par des éditeurs de distribution Open Source.

En outre, les frameworks applicatifs Open Source, standards de facto tels que Struts, Spring et Hibernate, sont pour les développeurs d'excellentes solutions afin d'échanger et de capitaliser sur leurs travaux et – parfois

de manière plus active que sur la base d'une spécification décidée de jure. Cette mixité s'exprime dans les projets d'aujourd'hui.

Le marché de l'Open Source se doit donc de tenir compte d'une réalité en 2006: les applications sont blended, une combinaison de solutions Open Source et d'éditeurs. Le choix d'un serveur applicatif ne doit pas



contraindre à faire des choix de framework. L'usage de l'Open Source ne doit pas être incompatible ni avec l'usage d'outils de développements avancés, ni avec des garanties de support et de certification et ce, quelle que soit la composition du mélange.

L'Open Source est un investissement et non une mode ou une croyance. Adopter – et au-delà supporter – la mixité liée aux technologies Open Source permet d'enrichir l'offre produit et de la démarquer là où elle est pertinente, sur des facteurs quantifiables. Le développement et déploiement blended de solutions Open Source sur la plateforme BEA offre ainsi le meilleur des deux mondes: souplesse et agilité des développements et simplicité d'administration, performance et robustesse en production. ■

Michel Lara, IBM "Windows est un marché"



Depuis 2001, IBM investit énormément dans l'open source et les standards ouverts. Aujourd'hui, les outils de développement reposent principalement sur un socle commun, Eclipse, projet phare du monde ouvert, initié par IBM. « L'open source est une véritable stratégie chez IBM, un modèle économique. Eclipse est open source et nous y proposons les plugins Rational. Il y a quelques années, on aurait pu penser tout développer et concevoir nos propres standards. Il faut apporter de la valeur ajoutée », commente Michel Lara.

La multiplication des licences pose aussi un problème de compréhension, un point de plus en plus sensible, visiblement. « J'aborde souvent la question des licences. Le développeur s'en moque, par contre,



Sherley Brothier, IdealX "Un marché pragmatique"

IdealX, avec d'autres SSLL françaises, constitue le fer de lance de l'année du logiciel libre dans les projets d'entreprise, marché en pleine croissance. Le mixage logiciels libres / ouverts et propriétaires ? « On le voit au quotidien. Il faut pouvoir s'interfacer avec les outils déjà en place. On prend le meilleur des deux mondes. Sur certains domaines, comme les environnements d'ERP, le libre est encore un peu en retard », confie Sherley Brothier.

« Dans les analyses de performances, les outils libres / ouverts sont performants et parfois on modifie du code et outils propriétaires ! Il n'existe pas de véritable virage tout libre, tout propriétaire. Les éditeurs reconnaissent les composants ouverts, notamment dans le développement. »

Pour le futur immédiat, 2006-2007, il est plus difficile de se faire une idée précise des évo-

à prendre !”

quand on est en production, on s'interroge. Un de nos clients intégrait Cloudscape dans sa solution développée, il fallait que l'on regarde le cadre juridique, Cloudscape intégrant la licence Derby. On peut rapidement aller aux conflits, aux procès. Quand on colle du code open source, il faut vérifier que l'on ne viole aucune licence ou brevets... Si on souhaite vendre un logiciel utilisant du code ouvert, cela devient vite un casse-tête juridique. Il existe des licences contaminantes et non contaminantes. Parfois, on réécrit tout simplement le code pour éviter tout problème ! Aucun projet fédérateur n'existe », poursuit Michel Lara. Et quand on évoque le problème d'une fondation commune à toutes les distributions, les mêmes difficultés surgissent. « Des éditeurs, Mandriva et Red Hat, ont des initiatives. On voit Linux diverger, et c'est une réelle inquiétude. »

Si aujourd'hui le débat n'est plus quel système choisir, Windows représente une opportunité. IBM travaille beaucoup avec Zend ou Apache et les récents accords entre Zend et Oracle ou Microsoft et JBoss confirment l'attrait de Windows. « Le marché devient pragmatique. Certains projets privilégient Windows. La version Linux d'Open Office a des bugs que l'on ne retrouve pas sous Windows. Pour eux, Windows est un marché à prendre. »

■ F.T.

lutions. « Difficile de répondre à une telle question. On s'oriente vers une mixité logiciels ouverts et propriétaire. Le choix devient (enfin) pragmatique. Cependant, cela va aussi dépendre de l'environnement cible », poursuit Sherley Brothier. Sur les licences, le problème semble désormais mieux maîtrisé par les entreprises. « Nos clients sont remarquablement au courant des licences. Ils savent de quoi on parle, malgré la relative complexité de la question. Même sur les licences propriétaires, ils connaissent la question. La licence

Franz Meyer, Red Hat “400 développeurs pour la communauté”



On ne présente plus Red Hat. Il fut le premier éditeur à packager et à distribuer une distribution « clé en main » aux utilisateurs. Au départ, « nous vendions des boîtes. Puis nous avons proposé du support, du service. Ce n'est pas un business visant le grand public. Mandriva, par exemple, tente un modèle grand public » raconte Franz Meyer, directeur Europe du sud, Moyen Orient, Afrique.

Acteur majeur de l'Open Source -avec une croissance annuelle de 40 % à 50 %, la stratégie est clairement là : « le modèle de développement Open Source est un gros avantage. Nous participons tous les jours à la communauté. On investit beaucoup en R&D, avec plus de 400 développeurs ! La contribution à l'Open Source est notre premier rôle. On partage beaucoup, notamment

avec Fedora. Red Hat est le second contributeur du noyau Linux, en nombre de lignes. Mais Red Hat n'a qu'une contribution, une contribution minoritaire, même s'il faut avoir un certain leadership sans devenir majoritaire. Il faut préserver la communauté et son modèle de développement. Beaucoup de projets ne sont pas pérennes. Il faut se poser des questions sur la fiabilité du projet et sur la pérennité de sa communauté. Qui paye les développements ? Ce sont les éditeurs. Chacun fait son business », dit Franz Meyer.

Si à première vue, le modèle de développement peut sembler anarchique, il ne l'est pas du tout, bien au contraire. Ce n'est pas pour autant qu'il faut renier le modèle propriétaire. « Au premier abord, le client ne voit qu'une seule tête, cela peut rassurer de n'avoir qu'un unique interlocuteur. Dans l'Open Source, le fournisseur n'a qu'une responsabilité partielle. C'est pour cela que l'on propose une assurance Open Source », poursuit M. Meyer.

On reproche souvent à Linux et à l'Open Source de manquer d'unité. Un défi que Red Hat ne sous-estime pas. Reste à le relever. « Il faut masquer la complexité à l'utilisateur. Il ne doit pas s'en apercevoir ou se poser la question de savoir si son matériel fonctionnera ou quelle version de logiciel utiliser. Nous devons améliorer la transparence d'utilisation, simplifier encore l'installation et faire en sorte que l'on puisse faire fonctionner n'importe quelle application, sans souci de compatibilité », argumente Franz Meyer. Outre cette amélioration bienvenue, quels sont les grands chantiers de Red Hat pour les futures versions ? « La virtualisation constitue une fonction phare. Nous aurons une solution complète sur cette technologie pour la fin 2006. Ensuite, il s'agit de banaliser le poste client, c'est-à-dire pouvoir retrouver son environnement de travail, ses applications, ses données où que l'on soit, sur n'importe quel ordinateur » conclut Franz Meyer.

■

est rentrée dans les mœurs » indique Sherley Brothier.

Sur la migration vers des outils libres, la situation va dépendre de l'environnement et du type d'application. Pour toutes les applications de type OpenOffice ou Firefox, la migration peut se réaliser, que l'on reste ou non sur Windows ou MacOS X. « Sur le développement, cela dépend des cas et ce ne sera pas aussi évident » conclut Sherley Brothier.

■ F.T.

Microsoft mise sur les licences et la communauté

L'annonce, il y a maintenant quelques semaines, de nouvelles licences logicielles n'a pas manqué de réveiller les différentes communautés. L'initiative Shared Source, lors de son lancement, n'avait pas enthousiasmé grand monde ; aujourd'hui, c'est l'inverse. L'un des plus célèbres projets ouverts, en shared source, est le projet Rotor, la mise à disposition du CLI du framework .Net sur BSD et MacOS X. Quelle est aujourd'hui la position de Microsoft face à l'open source et aux logiciels libres ?

Microsoft a pris conscience de l'ampleur du phénomène Linux et Open Source sur le marché. La dernière réponse en date fut l'annonce de trois nouvelles licences :

- Permissive Licence : on peut voir, redistribuer, modifier le code source sur des projets commerciaux ou non. Possibilité d'appliquer des royalties sur le code modifié. Ressemble à la licence BSD.
- Community Licence : tout code développé sous cette licence doit être redistribué sans réciprocité. Toute modification d'un code sous cette licence doit être diffusée. Ressemble à la licence Mozilla et proche de la GPL.
- Référence Licence : modification et redistribution du code interdites.

Des variantes, liées à la plate-forme Windows, existent.

Dans son premier commentaire, la Free Software Foundation Europe avait bien accueilli les nouvelles licences. « Ces licences tiennent en une page et Microsoft les utilise pour ses codes » précise Stéphane Kimmerlin (Responsable Stratégie pour la Division Plate-forme d'entreprise, Microsoft France). Si les anciens codes, déjà proposés dans le programme Shared Source, ne verront pas modifiée leur licence, tous les nouveaux codes utiliseront ces licences. C'est déjà chose faite pour les starter kit de Visual Studio 2005.

On pourra se demander, pourquoi Microsoft n'utilise-t-il pas une licence déjà existante

comme GPL, LGPL, etc. ? On entend et lit régulièrement qu'un code ne peut être ouvert/libre s'il n'est pas en licence GPL. Les éditeurs créent leur propre licence ouverte par commodité (Sun, Apple notamment) et pour leurs propres usages.

« Le processus de développement des projets open source s'est industrialisé. Tous les gros projets reposent sur un petit cœur, avec une "zone" de développeurs bénévoles, mais le gros de ces projets n'est plus sur Sourceforge. Ils sont financés par les grands groupes », poursuit Stéphane Kimmerlin. « On peut comprendre qu'on ne veuille pas dépendre des briques Microsoft. Il faut alors construire des briques alternatives. Les éditeurs se mettent d'accord sur les éléments où ils ne seront pas en concurrence directe, là où il n'y a pas de business à faire. »

Pour le moment, on ne sait pas si Microsoft proposera des licences blanches. C'est-à-dire, laisser la possibilité à des éditeurs tiers ou à des développeurs d'utiliser ces nouvelles licences pour leurs propres codes. Si tel était le cas, une véritable communauté pourrait alors se créer autour de ces licences. Aujourd'hui, la communauté se disperse entre différentes licences.

Optimiser JBoss sous Windows

Les relations entre Microsoft et certains acteurs du monde libre se sont normalisées. Il y a quelques mois, JBoss et Microsoft s'enten-



daient pour mieux optimiser JBoss sous Windows. La moitié des utilisateurs du middleware JBoss utilise Windows ! Il s'agit d'améliorer l'intégration de JBoss sur le serveur Microsoft, d'en optimiser le fonctionnement et les performances. Les travaux doivent débuter début 2006 avec des développeurs des deux sociétés. On sait déjà qu'il y aura une interopérabilité, support de développement sur ActiveDirectory, SQL Server. Pour JBoss, cela permet d'espérer une plate-forme mieux intégrée et pour Microsoft, de garder des clients de ses serveurs et pourquoi pas en espérer de nouveaux... Il ne faut pas se leurrer, une partie des projets open source connaît le succès grâce à sa disponibilité sur Windows. Il ne fait aucun doute que Microsoft saura être pragmatique et peut être tisser de nouveaux partenariats avec des éditeurs libres, notamment dans le middleware et les outils serveurs.

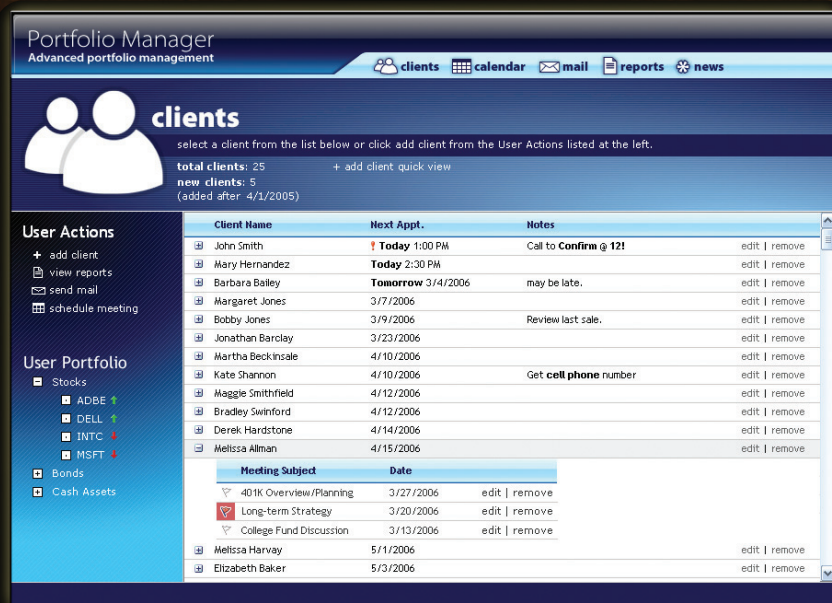
Office contre OpenOffice ?

Microsoft s'agit aussi sur le format natif XML dans sa suite bureautique. Après OpenDocument Format, intégré par défaut dans OpenOffice (et dans quelques autres suites), et standardisé depuis, Microsoft a du réagir rapidement en lançant son propre format ouvert. Mais celui-ci ne sera pas totalement libre, il restera lié à Microsoft, pour l'instant. L'éditeur espère une standardisation ECMA avant la sortie d'Office 12, fin 2006. Microsoft met en avant les limites du format OpenDocument, notamment le fait qu'il n'est pas possible de lier un schéma métier au document, ce qui sera possible avec le format Office 12. Cela permet une possibilité entreprise et plus complexe du format XML.

Sur les licences :

<http://www.microsoft.com/resources/sharedsource/licensingbasics/sharedsourcelicenses.msp>

■ François Tonic



2006 Vol. I Caractéristiques de performance du WebGrid et du WebTree

- utilisez la technologie AJAX afin de charger dynamiquement les données et autant que vous le désirez sans aucun postback, améliorant considérablement la réaction de vos applications web
- améliorez la performance globale de vos applications en prenant avantage d'un nouveau light weight renderer, réduisant la quantité de données envoyée au navigateur
- améliorez l'utilisation de vos applications avec de nouvelles techniques de databinding de haute performance.



WebGrid & WebTree

Lightweight WebGrid rendering et support étendu d'AJAX pour le WebGrid et le WebTree vous aident à délivrer des applications ASP.NET évolutives et de haute performance

NetAdvantage™ 2006 Vol. 1

L'ultime boîte à outils pour créer rapidement l'interface utilisateur de vos applications Windows Forms, ASP.NET et Tablet PC.

NetAdvantage vous offre les outils vous permettant de vous exprimer avec:

- plus de 100 contrôles pour le développement d'interface utilisateur pour Win Forms et Web Forms
- designers intégrés dans Visual Studio et Smart Tags afin de programmer visuellement, composition des données et rendu de vos interfaces utilisateur
- Métaphores UI pour Office, XP, Vista, Microsoft CRM, Outlook, Portails, Docking, Dashboards et bien plus en appliquant simplement une de nos présélections en mode design
- Haute performance AJAX pour les contrôles WEB
- Construit sur un framework garantissant consistance et promouvant
 - La réutilisation du code
 - Temps réduit d'apprentissage pour le développeur
 - Maintenance facile de vos applications
 - Une expérience productive des utilisateurs de vos applications
- 3 nouvelles versions par an - vous informant des dernières technologies
- Code source fournit pour tous les contrôles et designers*
- Support mondial grâce à nos équipes de support technique aux développeurs sur trois continents (Amérique du Nord, Europe et Asie)

Essayez dès aujourd'hui NetAdvantage et donnez facilement à vos applications une interface utilisateur riche et irrésistible.

www.infragistics.com/download

TestAdvantage™

Le premier outil prêt-à-utiliser automatisant le test des interfaces utilisateur en supplément du QuickTest Professional de Mercury afin de supporter les interfaces utilisateur Windows Forms développées avec NetAdvantage.

www.infragistics.com/testadvantage

Training

Apprenez comment implanter efficacement la technologie Infragistics pour interfaces utilisateur selon votre domaine d'expertise pour le développement de vos applications.

www.infragistics.com/training

Infragistics®
Powering The Presentation Layer

NetAdvantage avec Souscription et Support Prioritaire (875 €)* NetAdvantage avec Souscription (699€)*

*la souscription inclut tous les éléments ainsi que toutes les mises à jour; code source C# complet pour tous les éléments .NET et designers pendant un an, renouvelable. Prix hors taxes et susceptibles d'être modifiés.

N°Vert 0800 667 307 • sales-europe@infragistics.com

Développeurs Open Source : un marché porteur pour l'emploi

Editeurs, SS2L, SSII, entreprise utilisatrice, administration centrale ou locale : les employeurs potentiels de compétence Open Source se diversifient.

Les grands intégrateurs sont en train de créer un appel d'air sur les compétences Open Source », analyse Olivier Guilbert, P-DG d'IdealX ; Il est vrai que la plupart d'entre elles ont développé des pôles de compétences sur l'Open source et ont, assez naturellement, été puiser dans le vivier de certaines SS2L. Signalons Capgemini et sa filiale Sogeti, Atos Origin, EDS Answare, Bull. Bull a en effet décidé de capitaliser sur les compétences Open Source acquises sur le projet Novascale, pour proposer une offre de services complète autour des technologies Open Source.

Selon Jean-Pierre Barbéris, directeur général des activités Services et Solution de Bull, les compétences Open source sont assez faciles à trouver, dans la mesure où la plupart des universités et Ecole d'ingénieurs les utilisent. Le pôle Open Source de Bull étant notamment basé à Grenoble, Bull estime que les formations locales répondent assez bien aux compétences recherchées. Une formation initiale satisfaisante également aux yeux d'Olivier Guilbert, P-DG d'IdealX « qui va contribuer au développement de l'Open source dans les entreprises ». IdealX indique recruter une majorité de diplômés de grandes écoles : Supelec, Normale Sup, Centrale. Avis très différent d'Alexandre Zapolsky, P-DG de Linagora, qui souligne le manque de cursus spécialisés dans les logiciels libres

Pénurie sur certains profils

A peu près toutes les compétences sont recherchées : Java, encore et toujours, PHP, Xul, Python, Perl pour les langages, Ajax, Zope, Sambaé, Lamp... tout ce qui touche à Linux embarqué dans le secteur de l'industrie etc. Les grandes SSII se plaignent depuis près d'un an de pénurie sur Java et J2EE, mais ce sont particulièrement les SS2L, comme Linagora ou Open Wide, qui soulignent avoir du mal à recruter sur certaines technologies : Alexandre



Alexandre Zapolsky.



Patrick Bénichou.

Zapolsky indique avoir du mal à trouver sur Zope et Python. « Sur quinze à 20 CV reçus chaque jour par Linagora, très peu mentionnent des compétences Zope ou Python, ce qui freine d'ailleurs le développement de ce marché », remarque-t-il.

De son côté, Patrick Bénichou, P-DG d'Open Wide, signale avoir des difficultés à recruter sur certains langages et frameworks récents, comme Xul et Ajax. Open Wide et Linagora signalent des pénuries importantes sur la partie Linux embarqué.

Egalement dans les compétences manquantes, relevées par Linagora, Open LDAP ou les Infrastructures de clés publiques (PKI). Autant de sujets qui, selon A. Zapolsky manquent dans les formations d'ingénieurs actuelles. « Au-delà de ces technologies, c'est une partie des métiers associés à ces technologies qui manquent : les PKI ou les annuaires ne sont pas des sujets très techniques, mais il est difficile de trouver des ingénieurs capables de mettre en œuvre une infrastructure de clés publiques dans son aspect organisationnel. » Au-delà de technologies précises, Alexandre Zapolsky note un manque d'usage des logiciels libre en environnement professionnel de la part des candidats, qui sont souvent grands utilisateurs des Logiciels Libres, souvent contributeurs, mais pas forcément au niveau du développement. A contrario, il relève que sont reconnues, au niveau national comme international, des personnes qui, bien que pas particulièrement diplômées en informatique font preuve d'une compétence unique, et rentrent dans la catégorie Expert.

Où les trouve-t-on ?

Sur les sites des SS2L ou éditeurs précités : Linagora annonce entre 80 et 100 recrutements en 2006, Open Wide au moins une vingtaine, IdealX entre 20 et 30 personnes. Pour des postes de développeurs débutants, une des portes d'entrée principale reste le stage. Olivier Guilbert chez IdealX indique embaucher la moitié de ses stagiaires. Les deux sites suivants fédèrent beaucoup d'offres d'emploi dans le domaine du libre : <http://fr.lolix.org> ou <http://fr.joinux.com>. La plupart des offres concernent des postes de développeurs ou d'administrateurs réseaux. La clé du recrutement dans ce domaine va être – bien sûr – serait-on tenté de dire – la contribution à des projets Open Source et la notoriété acquise par le candidat au sein de la communauté – et pas seulement en participant à des forums.... « Nous recherchons des gens passionnés, impliqués dans la communauté, qui ne sont pas attirés par les grandes SSII », indique Patrick Bénichou. C'est également un point mis en avant par les recruteurs qui sont des pure players de l'Open source pour attirer les candidats : vous continuerez à être impliqués dans ce mouvement.

Et ça paye bien ?

Pas forcément beaucoup plus qu'ailleurs sur des postes de développement pur : entre 28 000 et 31 000 euros annuels pour un débutant, des prix qui remontent depuis quelques mois, davantage à Paris qu'en régions. Ça peut aussi être beaucoup moins dans la fonction publique, grille de salaire oblige : 23 KE...

Ça peut, en revanche, augmenter assez vite avec quelques années d'expérience : 35 à 40 KE avec deux à trois ans d'expérience, 42 KE à 50 KE pour des profils encore plus senior. A ce niveau, les rémunérations dépendent aussi d'autres compétences que technologiques, des compétences métier par exemple, ou la capacité de piloter un projet. Un chef de projet peut espérer entre 38 KE et 48 KE annuels, un directeur de projet plus de 45 KE et jusqu'à 60 KE annuels. Pour les profils expert, ça monte très haut, 80 à 100 KE mais ils sont TRES rares...

■ Carole Pitras

Ce n'est pas un type très sérieux.
Il va juste acheter **votre nom de domaine en .eu** avant vous.



Ne perdez pas de temps,
achetez votre nom de domaine européen.



**Pack
Web
Nom**

Protégez dès maintenant vos noms sur Internet

- Votre nom de domaine .eu, .com, .net...
- Service DNS
- Redirection web
- Emails illimités (redirection unique)
- Anti-virus/Anti-spam
- Hébergement 2 Mo (ext. à 50 Mo)
- Outil de création de site
- Trafic illimité

12 € HT/AN* (à partir de 12 € HT/AN soit 14,35 € TTC/AN)

**Pack
Web
Pro**

Pour les webmasters les plus exigeants

- Votre nom de domaine .eu, .com, .net...
- Service DNS
- Comptes POP/IMAP et FTP privés
- Anti-virus/Anti-spam
- Hébergement dynamique 50 Mo ext. à 300 Mo
- PHP4, Perl 5.0, ASP.NET, 2 bases MySQL
- Statistiques détaillées
- Trafic illimité

70,80 € HT/AN (à partir de 70,80 € HT/AN soit 84,68 € TTC/AN)

Avec plus de 200 000 noms de domaine gérés, AMEN est l'un des principaux fournisseurs européens de présence Internet.

AMEN est Registrar Officiel auprès de l'EURid
(European Registry of Internet Domain Names).



Compatibles

0892 55 66 77

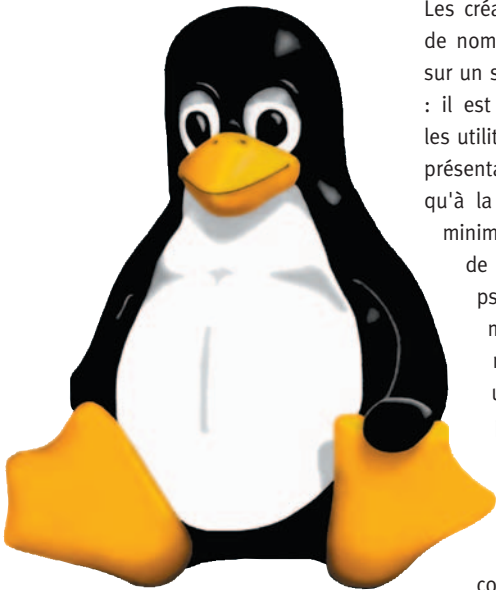
(0,34 € TTC/mn depuis la France 9h - 19h)

www.amen.fr

amen
IN WEB WE TRUST

Rootkits & Linux : la menace fantôme

Avant de déferler sur Windows, les rootkits ont longtemps menacé, essentiellement les systèmes Unix, puis Linux. Anatomie d'une menace discrète.



Le nom même cache difficilement les origines Unix des rootkits : "kit pour devenir root", on ne peut faire plus clair. Bien avant Windows, les systèmes Unix et leurs cousins (Linux et *BSD notamment), connaissaient les rootkits. Leur rôle est de permettre à un pirate de conserver le contrôle du système après l'avoir compromis : le rootkit s'installe donc après le piratage (c'est d'ailleurs souvent le premier réflexe du pirate), mais il ne le provoque pas. "Avant d'installer un rootkit, il faut pouvoir exploiter une faille. Si le système est correctement protégé, il n'y a donc rien à craindre des rootkits", rassure Jérémie Jourdain, consultant sécurité chez Intrinsec.

Être furtif

Si ce n'est pas le cas, en revanche, les choses peuvent alors devenir beaucoup plus complexes. Car si le rôle premier du rootkit est de permettre au pirate de revenir à sa guise sur le système qu'il vient de compromettre, il ne peut atteindre son objectif qu'en demeurant invisible aux yeux de l'administrateur. C'est donc ici la caractéristique essentielle des rootkits : la furtivité.

Les créateurs de ces parasites ont développé de nombreuses techniques afin de les cacher sur un système Unix. Les plus simples d'abord : il est particulièrement simple de remplacer les utilitaires d'audit système par des versions présentant la même taille, la même date et jusqu'à la même somme de contrôle CRC. "Au minimum, un rootkit recompilera lui-même de nouvelles versions des utilitaires tels ps ou top, afin qu'elles disposent des mêmes caractéristiques que les originaux", poursuit Jérémie Jourdain. Sur un système compromis, ces outils remplaceront les versions légitimes et ne montreront pas les fichiers ou les processus du rootkit durant l'audit.

Voilà qui prêche en la faveur d'un conseil de sécurité bien connu : ne jamais laisser de compilateur sur un système en production (non pas qu'il soit impossible au pirate d'installer le sien, mais autant ne pas lui faciliter la vie).

Par ailleurs, même les rootkits les plus simples sont capables d'éditer les journaux du système (s'ils sont accessibles...), afin de retirer leurs propres traces. Là aussi, la bonne pratique de sécurité consistant à exporter ses journaux en temps réel vers une imprimante, ou un support non réinscriptible, a du bon.

Des rootkits variés

Pour le reste, ces rootkits primitifs se contentent de dissimuler leurs fichiers dans des recoins du système, en espérant que personne n'ira les y chercher (par exemple /dev). "Ce sont la majorité des rootkits que nous rencontrons. Ils ne sont pas du tout sophistiqués, et ils se contentent en général de modifier la configuration d'inetd afin d'écouter à un port particulier (pour y attacher un shell, ndlr). Ils ne font pas vraiment d'effort non plus pour cacher leurs fichiers", relativise Jérémie Jourdain.

Très bon exemple d'un tel rootkit en mode utilisateur particulièrement répandu : Tron. Ce rootkit embarque ses propres versions des utilitaires find, du, ifconfig, login, ls, netstat, ps, top, sz et une version piégée du démon syslog

(afin de ne plus journaliser son activité). Il installe ensuite également un sniffer réseau pour faire bonne mesure.

Ces rootkits là sont particulièrement simples à détecter à l'aide d'outils spécialisés, tels chkrootkit (disponible sur <http://www.chkrootkit.org/>). Et parfois même sans, si l'administrateur sait où regarder et surtout, avec quels outils : Torn, par exemple, "oublie" de livrer une version piégée de lsof, et il peut donc facilement être détecté de la sorte. De la même manière, il suffit d'utiliser des copies "fraîches" des outils modifiés par le rootkit pour découvrir la supercherie.

Mais leurs créateurs n'en sont pas restés là : puisqu'il est si simple de détecter la présence d'un parasite vivant en userland (le mode utilisateur), ils se sont alors rapidement tournés vers le noyau, et dès 1997 le magazine Phrack publiait un article décrivant l'infection du noyau de Linux. Désormais, les rootkits Linux les plus efficaces modifient ainsi le noyau du système, afin d'en prendre le contrôle et être en mesure de rester cachés.

Pour cela, plusieurs techniques ont été mises au point. La plus évidente consiste bien sûr à créer un module noyau (LKM). Le rootkit est alors un module, au même titre que n'importe quel pilote du système. À ceci près que son rôle est d'intercepter certains appels systèmes (syscalls) afin de renvoyer des valeurs truquées (notamment lors de l'affichage du contenu d'un répertoire ou de la liste des processus actifs en mémoire).

L'utilisation d'un noyau compilé statiquement réduit alors l'exposition à ce type de rootkits. Autre technique, très similaire : modifier un module du noyau existant. Plutôt que d'écrire son propre LKM, l'auteur du rootkit peut en effet choisir "d'infecter" un module déjà installé afin d'obtenir les mêmes fonctionnalités que s'il avait créé son propre LKM. Il "ajoute" alors une fonction d'interception des appels systèmes à un module censé fournir un tout autre service (un pilote de périphérique par exemple).

Dernière technique, enfin, probablement la plus séduisante du point de vue de la programmation : l'écriture directe dans le device `/dev/kmem`. Ce dernier offre en effet une "fenêtre" sur l'espace mémoire occupé par le noyau en fonctionnement. Avec un peu d'astuce (et beaucoup d'essais infructueux !) il est alors possible de modifier le noyau à la volée. Se protéger contre une telle attaque exige alors de corriger le noyau, à l'aide d'une rustine spécifique (et pas entièrement efficace).

Linux, une terre promise

Mais il ne suffit pas de pouvoir écrire, d'une manière ou d'une autre, dans le noyau du système pour faire un rootkit. Il est surtout nécessaire de savoir contrôler les appels systèmes afin de pouvoir intercepter ceux susceptibles de renvoyer la vision d'un système infecté.

Pour cela, les rootkits noyau utilisent une technique guère éloignée des virus furtifs sous DOS à la fin des années 80 et au début 1990 : ils détournent les interruptions systèmes et y substituent leur propre code.

Sous Linux, la table des interruptions s'appelle Interrupt Descriptor Table (IDT), mais le prin-

cipe est le même. Un rootkit peut ainsi, au choix, modifier la table elle-même afin de fournir des pointeurs vers ses propres routines plutôt que celles du système, ou laisser la table originale et ne changer que les adresses des routines qu'il souhaite modifier (généralement celles qui permettent l'exploration du disque ou des processus).

Dans tous les cas, le résultat est le même : quel que soit l'utilitaire employé par l'administrateur pour contrôler son système (y compris des versions propres), il ne verra rien. Les appels systèmes seront dirigés vers des routines fournies par le rootkit et ce dernier présentera la vue rassurante d'un système non infecté.

SuckIT est probablement le plus célèbre de ces rootkits noyau. En 2003, un pirate parvenait à compromettre quatre machines du projet Debian et à y installer SuckIT. Sa détection s'était montrée alors particulièrement difficile malgré l'appui d'experts et la présence sur les serveurs, au moment de l'installation, du package de détection d'intrusion aide (qui, toutefois, a été le premier à donner l'alerte en indiquant une modification de `/sbin/init`, le processus maître sous Linux).

Mais il y en a d'autres : "J'ai connu un rootkit qui virtualisait totalement le système après en avoir pris une image avant l'infection. Il agissait alors comme une couche d'abstraction qui permettait d'avoir une vision du système infecté absolument identique à celle du système propre, jusque dans les sommes de contrôle MD5", se souvient Benoît Picaud, d'IdéalX. Il pourrait s'agir d'Adore-ng, connu pour exploiter la couche VFS (*Virtual File System*) du noyau Linux et avoir ainsi le contrôle complet de ce qu'il présente à l'utilisateur.

La détection de tels rootkits noyau est, bien entendu, beaucoup plus difficile, notamment lorsque le système est en fonctionnement. La meilleure technique consiste alors à monter le disque de démarrage sur un autre système, afin de l'explorer. Et si cela n'est pas possible, des outils spécifiques permettent de contrôler la table des interruptions, ou les appels systèmes à la recherche de changements non autorisés. C'est le cas par exemple de `kern_check`, `CheckIDT`, `check-ps` (détection de processus cachés) ou `Kstat` (exploration du noyau via `/dev/kmem`).

■ Jérôme Saiz

Nat System. Des outils de pointe pour optimiser votre productivité.

Allez au bout de vos projets en choisissant vos outils de développement dans notre gamme ; comme Nateo -par exemple- qui vous propose de développer vos applications de gestion en J2EE.

Après avoir construit son expertise sur l'édition d'outils de développement d'architectures client/serveur (NS-DK, NatStar), Nat System s'est attaqué avec brio aux nouvelles technologies.

Profitez-en !

Comme plus de 500 grands comptes, simplifiez l'ouverture de votre patrimoine applicatif en choisissant Nat System.



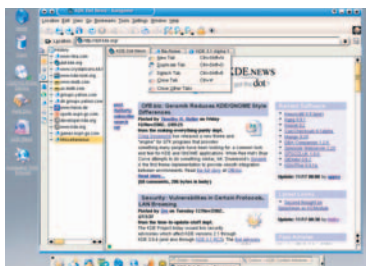
Nat System
Stratégiquement vôtre

Les distributions Linux

Depuis le premier comparatif des distributions Linux, paru dans Programmez n°68, de nouvelles distributions ont pu apparaître ou convaincre le marché de leur utilité, les plus marquantes étant Ubuntu, OpenSuse, Knoppix. Sun se lance aussi dans la bataille avec OpenSolaris. Comment choisir ? Voilà toute la question.

Les distributions se ressemblent autant qu'elles se différencient. Si de nombreux packages et applications demeurent identiques quel que soit l'éditeur, de nombreux détails font la différence : sur l'installation, la stabilité, les dernières versions des applications, le support / assistance, la documentation, etc. Les premières questions à se poser concernent :

- l'utilisation que je souhaite en faire
- me faut-il un support / assistance
- les outils que je veux utiliser fonctionnent-ils ou sont-ils présents par défaut ?
- mes périphériques fonctionnent-ils ?
- la migration sera-t-elle simple ?



KDE ou Gnome, vous avez le choix de l'interface.

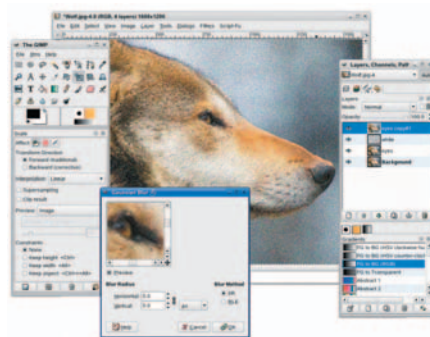
Ce sont ici des questions élémentaires. Ensuite, on peut se poser des questions sur l'accessibilité aux fonctions, à l'administration, la facilité de mise à jour, la compatibilité avec le matériel (éventuellement la présence de certification), le nombre de packages proposés, la fréquence de mise à jour de la distribution, les fonctions de sécurité, etc.

En fonction de l'utilisation

Toute distribution ne convient pas à toutes les utilisations. Vu la segmentation du marché des distributions Linux et chez un même éditeur, mieux vaut savoir ce que l'on veut faire. Vous aurez alors une grande différence d'une édition Linux à une autre, sur telle ou telle utilisation. Par exemple, la Mandriva Linux Discovery n'est pas adaptée au poste de travail professionnel et une SUSE Linux Enterprise Server ne

DISTRIBUTIONS ET USAGES				
	Poste personnel	Portable	Poste de travail	Développement
Mandriva Linux 2006 Discovery	*	*		
Mandriva Linux 2006 PowerPack		*	*	*
SUSE Linux 10	*	*	*	*
OpenSUSE	*	*		*
Red Hat Enterprise Linux WS			*	*
Fedora	*	*	*	*
Ubuntu	*	*		

conviendra pas à une utilisation personnelle ! Si vous êtes habitué au système, voire à Linux, une version librement disponible (Ubuntu, OpenSUSE, Fedora...) suffira à de nombreux usages. Mais dès que l'on souhaite faire de la production, du développement professionnel, excepté si vous êtes expert Linux, n'hésitez pas à jouer la sécurité en choisissant une distribution payante incluant support et assistance.



Gimp, « le Photoshop libre ».

ce, quitte à souscrire un support complémentaire si besoin.

Cela sera d'autant plus nécessaire en entreprise (grande ou petite) où dans certains domaines, il faudra déployer des dizaines de postes ou faire des serveurs critiques à haute

disponibilité. De toute manière, rien ne vaut des tests pour évaluer une distribution. En production ou en utilisation professionnelle, privilégiez les distributions connues et reconnues, évitez les packages exotiques.

Durée de vie des versions

Dans les critères de choix, outre l'adéquation distribution – usage et le support / assistance (disponible auprès de l'ensemble des éditeurs) la durée de vie des versions peut constituer un élément sensible. En fonction

de cela, votre plate-forme peut être plus ou moins pérenne et nécessiter des mises à niveaux plus fréquentes.



Si on prend Red Enterprise Linux WS Standard et l'édition Desktop, fondamentalement, il n'y a pas de différence, le socle est le même, hormis une différence sur la mémoire vive acceptée et que le Desktop est mono processeur. Un détail, très utile en entreprise, en production et pour le développeur voulant une plate-forme pérenne, Red Hat assure une durée de vie pour chaque version de 7 ans à compter du jour de sa sortie sur le marché. Fedora couvre une durée de 12-18 mois et ne bénéficie pas de certifications. Alors que les versions officielles en bénéficient (logiciel + matériel). À cela s'ajoute l'assurance open source, par défaut dans la souscription. Cela signifie que vous bénéficiez de la garantie d'éviter tout conflit juridique vis à vis des composants open source utilisés par l'éditeur.

Mandriva ne propose pas d'assurance open source comparable à celle de Red Hat. Sur la durée de vie des versions, elle est en général de 18 mois pour les distributions de type PowerPack, 24 mois pour les solutions d'entreprise. Pour les versions grand public

39 € HT/MOIS

soit 46,64 € TTC/MOIS pour un engagement semestriel

Pour une fois, votre Baby ne va pas vous coûter cher.

Livré en 2 heures !*



Baby d'AMEN, le serveur dédié le moins cher du marché.

Redhat/Fedora/Debian/Windows Server 2003 • Interface d'administration Webmin ou TSE
AMD Sempron 2200 ou Duron 1600 • IDE 80 Go/RAM 256 Mo • 1 adresse IP • Trafic 1 To



- Athlon XP 2400 ou Sempron 2400
- IDE 80 Go / RAM 512 Mo
- 2 adresses IP
- Amen Data Backup 2 Go
- Amen Reboot instantané
- Interface Plesk 7
- Trafic illimité¹

59 € HT/MOIS
soit 70,56 € TTC/MOIS pour un engagement semestriel



- Duron ou Sempron 2800
- IDE 120 Go / RAM 768 Mo
- 4 adresses IP
- Amen Data Backup 4 Go
- Amen Reboot instantané
- Interface Plesk 7
- Trafic illimité¹

89 € HT/MOIS
soit 106,44 € TTC/MOIS pour un engagement semestriel



- Athlon XP 3000
- RAID 2x160 Go / RAM 2 Go
- 4 adresses IP
- Amen Data Backup 6 Go
- Amen Reboot instantané
- Interface Plesk 7
- Trafic illimité¹

139 € HT/MOIS
soit 166,24 € TTC/MOIS pour un engagement semestriel

Avec plus de 3000 clients sur serveurs dédiés et plus de 200 000 noms de domaine gérés, AMEN est l'un des principaux fournisseurs européens de présence Internet.

- 'Satisfait ou remboursé' sous 10 jours
- Assistance technique par mail et par téléphone au 0899 70 9001 (1,34 € TTC l'appel puis 0,34 € TTC/mn)
- Offres de serveurs dédiés également disponibles sans engagement de durée



0892 55 66 77

(0,34 € TTC/mn depuis la France 9H - 19H)

www.amen.fr

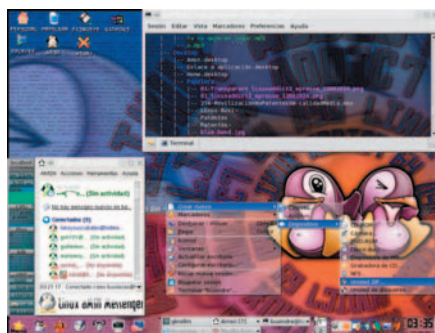
amen
IN WEB WE TRUST

(Discovery), le cycle est de 12 à 18 mois selon les composants, cela n'inclut pas les versions librement téléchargeables ou distribuées dans la presse. Concernant SUSE, Novell assure le support (maintenance gratuite limitée aux patches) de 2 ans pour SUSE Linux 10 (cycle de vie habituel chez cet éditeur).

Tarification et support

Là aussi, c'est en quelque sorte une jungle dans laquelle, on perd parfois le fil. Si pour les distributions librement téléchargeables des éditeurs (OpenSUSE, Fedora, Mandriva Linux...), il n'y a aucun support, aucune assistance, les distributions payantes (en boîte ou par souscription), proposent toujours un minimum de support.

Le niveau de prestation est très variable. Ainsi pour une souscription d'un an à une Red Hat, le support est valable un an (web, téléphone), avec les mises à jour. Pour une SUSE ou une Mandriva, l'assistance standard est limitée dans



L'interface se personnalise très finement si on sait y faire.

le temps (entre 30 et 90 jours), le support téléphonique est parfois optionnel. À vous de voir, ensuite, le niveau de services que vous souhaitez. À noter qu'un support Ubuntu (payant) est possible auprès de « l'éditeur ». Au-delà de la distribution, les éditeurs peuvent étendre les services de support à des solutions tierces. Par exemple, SUSE propose du support JBoss ou MySQL.

Vous pouvez aussi vous certifier, ou vous former, sur telle ou telle distribution. SUSE, Red Hat, Mandriva (en interne ou via des partenaires) propose une gamme complète de formations et certifications, avec des tarifs très variables (ex. : 750 euros pour un examen Red Hat, et jusqu'à 2 500 euros pour une semaine de cours).

Certifications matériels / logiciels :

SUSE Linux : <http://cdb.novell.com/>

Red Hat : <http://bugzilla.redhat.com/hwcert/>

Mandriva : <http://wwwnew.mandriva.com/fr/hardware>

	Mandriva Linux 2006 Discovery	Mandriva Linux 2006 PowerPack	SUSE Linux 10.x	OpenSUSE 10.x	Red hat linux desktop	Red hat Enterprise Linux WS Standard v4	Federa Core 4	Ubuntu 5.1.x
64-bit	oui	oui	oui	oui	oui	oui	oui	oui
Support Web/durée	oui / 30 jours	oui / 60 jours	oui / 90 jours (sur installation)	non	oui 24/7, l ou 3 ans	oui 24/7, l ou 3 ans	non	oui
Support téléphonique	option	option	oui / 90 jours (sur installation)	non	oui jours ouvrés l ou 3 ans	oui jours ouvrés l ou 3 ans	non	non
Mise à jour	1 mois gratuit offert	1 mois gratuit offert	oui	n/a	selon durée souscription	selon durée souscription	n/a	Tous les 6 mois
Support payant complémentaire	oui	oui	oui	non	oui	oui	non	oui
Support Français Web/Téléphone	oui/oui	oui/oui	oui/oui	n/a	oui/oui	oui/oui	n/a	oui/nc
Prix distribution	44,90 €	79,90 €	à partir de 64 €	gratuite	56 € par an par pack de 50	143 € basic 239 € standard	gratuite	gratuite

Attention : les prix sont donnés à titre indicatif. Tarifs éditeurs.

Le coin développeur

Novell intègre son environnement ouvert compatible .Net, Mono et embarque Eclipse 3.x pour le développement Java. Cette distribution est très bien fournie en SGBD avec Derby (version Apache de Cloudscape), MySQL et PostgreSQL. On notera aussi la présence de la solution XEN pour la virtualisation (qui se généralise dans les distributions). Red Hat / Fedora possède aussi une excellente offre, très proche d'une SUSE ou Mandriva. Nouvelle importante pour les développeurs voulant utiliser C# / ASP.net sans les outils Microsoft, la future Fedora Core 5.0 intégrera en standard Mono ! Une annonce importante qui permet après SUSE d'étendre le support de Mono dans le monde Linux. De plus en plus d'ou-

tils commerciaux fonctionnent sous Linux et les solutions ouvertes dans le test, la modélisation évoluent rapidement. Le seul point faible demeure la conception web malgré la présence de NVU, il n'existe toujours pas de véritable dreamweaver ou golive like... À noter que Red Hat propose Red Hat Developer Suite, compatible avec Red Hat Enterprise Linux. Il s'appuie sur Eclipse. Idéal pour développer des applications Linux ou multi-plates-formes.



La distribution Mandriva.

LES OUTILS DE DÉVELOPPEMENT

	Mandriva Linux 2006	SUSE Linux	Red Hat Enterprise	Fedora Core 4
Principaux outils	Gcc, xhtml-cole, gnome-ypton, perl-Gilb, j2re, NVU, strace, Qt, Ruby, emacs, vim, SGBD, etc. En plus dans PowerPack : postgresql, sqlite, MySQL, unixODBC, outils Intel...	Mono, Eclipse, Kdevelop, Derby, MySQL, Postgres, Java, gcc...	Kdevelop, GCC, Qt, SGBD, Java, Eclipse (en téléchargement)...	GCC, GDB, junit, Kdevelop, Qt, ODBC, outils SGBD
Principaux langages	Perl, Python, C, C++, Java, latex	Perl, Python, C, C++, Java, C#/.NET, Tcl	Java, perl, python, C, C++	Java, perl, python, C, C++

Listes des packages

SUSE Linux :

http://www.novell.com/products/linux/packages/professional/index_group.html

Mandriva Linux 2006 :

<http://wwwnew.mandriva.com/fr/individuals/products/2006powerpack/packages32>

Les distributions

Ubuntu 5.1.0 : ce Linux dérive de la distribution Debian. Plus rustique que les autres, du moins pour l'installation, elle profite de la solidité de sa base Debian. Evolution rapide. Cependant, des problèmes de compatibilités



peuvent apparaître. Peu adaptée à une utilisation professionnelle.

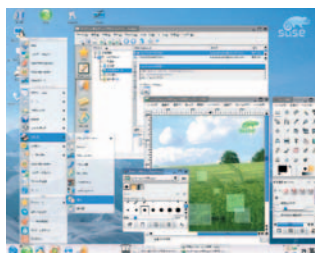
Les +

Stabilité, performance, mise à jour tous les 6 mois. Sa popularité.

Les -

Pas de compte root, installation rustique et peu rapide. Nécessite l'ajout d'outils pour le développement. Pas de compatibilité totale avec Debian Sarge.

OpenSUSE 10 : version gratuite et ouverte de SUSE, mise en place récemment par Novell pour contrer Red Hat – Fedora. Distribution identique aux versions commerciales. Système très polyvalent pour un usage standard.



Les +

Version librement téléchargeable, identique à la version commerciale. Bonne stabilité.

Les -

Pas de support, pas de téléchargement DVD (sur le site officiel). Manque d'outils de développement.

SUSE Linux 10.0 : une des meilleures distributions pour tout usage. Stable et performant sur portable aussi bien que sur desktop.

Les +

Installation, stabilité, les outils disponibles.

Les -

Documentation parfois en Anglais

Fedora Code 4 : une distribution homogène et très polyvalente pouvant servir aussi bien à la maison que sur le poste de travail ou comme plate-forme de développement.



La distribution Red Hat Fedora.

Comme pour les autres distributions du même type, le manque de support officiel ne favorise pas une utilisation en production.

Les +

Stabilité, installation, panoplie des outils

Les -

Support, pas de CD officiel

Mandriva Linux 2006 : si la Discovery permet de démarrer en douceur dans l'univers Linux, il est certain que la version PowerPack



sera bien plus intéressante pour des utilisateurs plus exigeants souhaitant exploiter Linux sans à avoir à rajouter de nombreux logiciels. Si vous maîtrisez votre système actuel, la PowerPack est le choix idéal.

Les +

Bonne tenue générale, installation, liste des outils en PowerPack, prix, Xen

Les -

Discovery assez pauvre fonctionnellement

Red Hat : elle est basée

sur la Fedora (tout comme l'ensemble de la gamme Red Hat). Bien finie, elle convient parfaitement au poste de travail. Elle est plutôt orientée entreprise par son prix et les prestations proposées. Convient aussi à un usage plus personnel. Cependant, si vous utilisez un portable, ce n'est peut-être pas le meilleur choix à cause du manque d'outils spécifiques.



Les +

Basée sur Fedora, stabilité – installation, support, richesse fonctionnelle

Les -

Gamme étendue de services et prestations parfois peu claire. ■

Open source : double licence et modèle économique

Les logiciels libres provoquent il y a encore peu au sein des DSI, deux grandes familles d'interrogations. Comment peut-on produire des logiciels de qualité dans un mode « anarchique » ? Comment peut-on s'assurer de la pérennité de logiciels développés par des bénévoles ?

La première famille de questions pointe en fait le modèle communautaire de production des logiciels libres. Si ce modèle reste encore globalement assez mal connu, il est clair que la fiabilité et la qualité des logiciels produits sont de moins en moins contestées. Linux, Apache, MySQL, Firefox, Thunderbird, Openoffice, pour n'en citer que quelques uns, sont passés par là. Au final, les faits chassant les doutes, les questions que se posent aujourd'hui les entreprises portent plus sur les domaines couverts par des solutions Open Source arrivées à maturité que sur la façon dont celles-ci sont produites. Les DSI intègrent désormais à leur stratégie les avantages qu'ils attendent de l'Open Source : respect des standards et des normes gages d'interopérabilité, indépendance (ou tout au moins effet de levier) vis-à-vis de fournisseurs devenus trop incontournables, économies budgétaires pour l'acquisition et la maintenance de composants logiciels devenant inéluctablement des « commodités logicielles ».

Vers une industrie de l'Open Source ?

La question, en revanche, de savoir par qui sont produits les logiciels Open Source, selon quelles motivations ou selon quels modèles économiques, est clairement une question d'actualité, loin d'être mineure quant à l'avenir même des logiciels libres.

Sur ces deux dernières années, plusieurs tendances de fond semblent se dégager et s'accroître. Tout d'abord, les grands acteurs mondiaux de l'informatique sont de plus en plus impliqués dans la production ou la « libération » de logiciels en Open Source. On peut citer le récent exemple d'IBM, qui, fort satisfait du succès remporté par la mise en Open Source de son IDE Eclipse, renouvelle cette fois l'opération avec un serveur d'application J2EE et une solution de gestion de stock. On peut éga-



lement mentionner les apports de Sun ou de Novell qui ont chacun libéré des logiciels emblématiques tels que Solaris, hier encore jalousement détenus en Copyright. On l'aura deviné, ces dons n'ont rien de philanthropique, ils s'inscrivent dans des stratégies marketing planétaires visant à diffuser des standards, des frameworks ou des environnements logiciels, qui, de facto, permettront de mieux occuper un marché et la diffusion d'une gamme connexe de produits ou de services. Ces grands acteurs prennent place dès à présent au sein du mouvement Open Source, par le niveau de leurs contributions, comme ils l'ont fait hier au sein des organismes de normalisation.

Parallèlement, apparaissent de nouveaux acteurs dans le sillage du succès d'une technologie Open Source particulière. On peut citer, parmi les plus connus, MySQL AB, Trolltech et sa bibliothèque graphique multi

plate-forme, QT ou encore, le JBoss Group et son serveur d'application J2EE. Ces nouveaux entrants réinventent actuellement le modèle d'éditeur de logiciels en essayant de concilier les attentes traditionnelles des clients (la fourniture de support et de correctifs, des informations sur les roadmap, des documentations techniques ou marketing...), les contraintes de licences des logiciels libres et un verrouillage de leur business modèle. La mise en place par ces nouveaux éditeurs d'une politique de double licence semble notamment en passe de devenir la règle. Il s'agit, dans les grandes lignes, de libérer un logiciel en Open Source et de lui adjoindre en parallèle une version en licence propriétaire, qui va conférer aux clients des droits et avantages particuliers. Les versions propriétaires ouvrent généralement droit à des services de support, à des informations privilégiées réservées aux clients et souvent à des fonctionnalités avancées qui ne sont pas

DÉVELOPPEZ VOS COMPÉTENCES AVEC O'REILLY !



www.oreilly.fr



► Une méthode révolutionnaire et incroyablement efficace pour bien maîtriser les design patterns et les bons principes de conception.



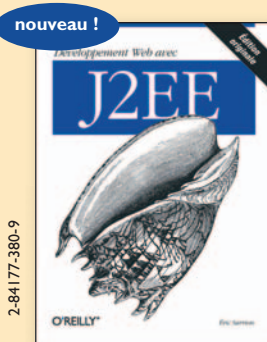
► Un ouvrage pour enfin apprendre Java d'une manière différente et incroyablement efficace.



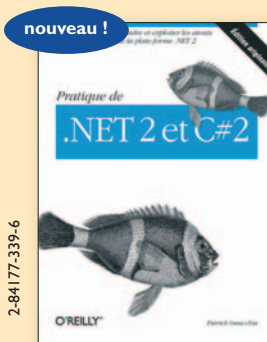
► Un ouvrage qui vous livrera toutes les clés pour écrire du code Java plus léger, plus rapide, plus propre et plus facile à maintenir.



► Le must des manuels de référence sur Java augmenté et mis à jour pour Java 5.0.



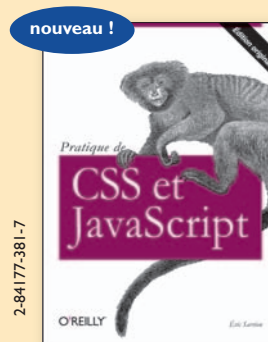
► Un ouvrage complet et sans bavardage pour les développeurs souhaitant maîtriser rapidement J2EE.



► Enfin « le Smacchia » dans sa version .NET 2. La bible du développeur .NET.



► 167 solutions efficaces aux problèmes les plus souvent rencontrés par les programmeurs C.



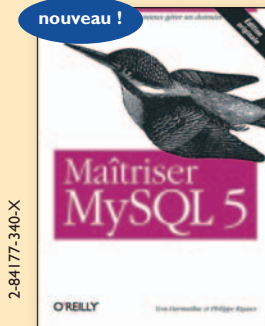
► Un guide pratique et compact pour que le Web 2.0 ne se fasse pas sans vous.



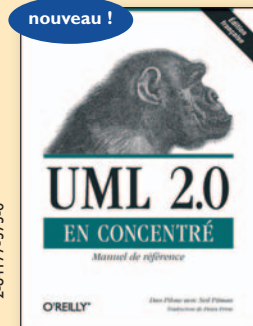
► Référence complète sur XML et les technologies génériques l'utilisant. En bref, la bible de tous développeurs XML.



► Référence complète sur SQL et ses différentes déclinaisons.



► Des outils et des méthodes de spécialistes pour optimiser sa gestion de données.



► Référence complète sur UML 2.0.

L'INFORMATIQUE À LA SOURCE

O'REILLY®

Pour recevoir notre catalogue, envoyez un email à info@editions-oreilly.fr en indiquant le nom de ce magazine.

Ouvrages en vente dans toutes les librairies !



disponibles dans la version totalement libre. Parfait me direz-vous si cela correspond à une professionnalisation et à une industrialisation du mouvement Open Source qui répond aux attentes des entreprises. En quoi cela peut-il nuire au développement des logiciels libres ?

Droits d'auteurs et double licence

Pour essayer de répondre à cette question, il faut revenir aux fondamentaux des logiciels libres. En simplifiant grossièrement, les logiciels libres sont soumis à des licences qui octroient un grand nombre de droits (usage illimité, accès aux sources, modification, diffusion, duplication...), mais qui les protègent de toute tentative de récupération commerciale de type « copyright ». Contrairement à une idée reçue, le logiciel libre s'inscrit à l'inverse d'une démarche d'abandon du droit d'auteur. Les auteurs d'un logiciel restent en effet seuls détenteurs du droit de faire évoluer ses conditions d'utilisation, à savoir, la licence qui le régit. Or, par nature, un logiciel libre est le résultat de multiples contributeurs, chacun pouvant se prévaloir de ses droits d'auteurs. On comprend aisément la difficulté à mettre en œuvre un système de double licence au sein d'une communauté réellement ouverte de contributeurs, chacun d'eux pouvant s'opposer à une décision concernant ses propres contributions logicielles. La gestion de la double licence, au plan juridique, peut très vite deve-

nir inextricable, et le seul moyen pour un éditeur de se préserver de ces difficultés (et de rassurer ses clients à ce sujet) est de restreindre l'accès aux développements proprement dits. Les effets de bords peuvent être assez immédiats : risque d'appauvrissement de la communauté de développeurs, risques de dérivations de projets (fork), ainsi que les risques de retour des vieux démons de l'édition, comme les

marketing ou les sorties de versions prématurées.

D'un modèle de licence vers un modèle de communauté ?

La licence GPL et ses très nombreuses licences dérivées, ont permis, grâce à ses protections, un véritable essor du développement logiciel communautaire depuis plus de 20 ans maintenant. Cependant, les licences de logiciels libres précisent bien peu de choses quant aux modalités de gouvernance des communautés humaines qui les produisent et les font vivre. À titre d'exemple, les différentes licences libres n'imposent pour la plupart aucune contrainte sur les modalités de diffusion du logiciel sur Internet, ni sur les règles de participation d'un contributeur à leur développement. Il en résulte une évolution de différents modèles de communautés drastiquement différentes. D'un côté, des « fondations » telles que les fondations Apache, Mozilla ou en Europe, la communauté Object Web, mettent en œuvre des gouvernances fortement ancrées sur la technologie et sur une légitimité de compétences et de contributions. De l'autre, des communautés fortement pilotées par des éditeurs, dont l'accès à la contribution est intimement lié à une politique marketing et commerciale.

Du côté des services, les choses s'accroissent également. On assiste notamment à une rapide

montée en compétences des SSII dans le sillage des pionniers spécialisés (SSLL) qui offrent des services clé en main d'intégration et de support. Parallèlement, le métier d'éditeur de distribution, dont la valeur ajoutée repose sur l'assemblage et le « packaging » de nombreux composants Open Source (ce qui permet aux entreprises d'accéder à ces composants de façon industrielle et supportée) semble plébiscité par le marché, si l'on en croit le récent développement de Red Hat au niveau mondial.

On le voit, l'arrivée à maturité de nombreuses solutions Open Source, véritables alternatives aux logiciels propriétaires, induit de nouvelles règles du jeu au niveau des acteurs économiques mondiaux qui apprennent à gagner de l'argent autour des logiciels libres. Quels que soient les modèles économiques expérimentés, il ne faudra pas perdre de vue que le logiciel libre puise sa force et sa pérennité dans son ouverture et sa diversité. Le marketing commercial appliqué trop brutalement au modèle du « Bazar », pour séduisant qu'il puisse paraître aux entreprises à court terme, pourrait bien porter les germes de son propre appauvrissement. Gageons que les passionnés de l'Open Source sauront trouver les conditions de l'harmonie entre le développement communautaire et le développement économique.



■ Patrick Bénichou,
Président fondateur d'Open Wide

ABONNEZ-VOUS

et recevez **GRATUITEMENT**

DEMO CHARGE 2005



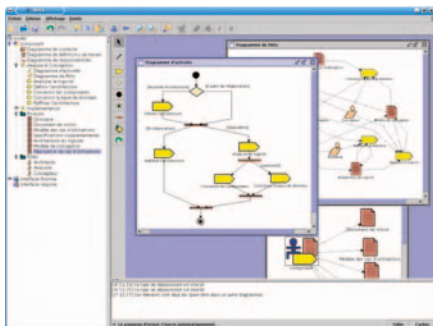
Pour obtenir leur licence gratuite, les nouveaux abonnés devront se connecter à l'URL suivante : <http://www.kapitec.com/Offres/Programmez/DC2006.html>

Les meilleurs outils Open Source

Apes

Apes (*Apes is Process Engineering Software*) s'appuie sur les spécifications du SPEM (Software Process Engineering Metamodel) rédigées par l'OMG. Ce logiciel placé sous licence GPL, qui a vu le jour en 2003, est capable de modéliser de manière entièrement graphique des processus métier (activité, définition de travail), en utilisant des diagrammes UML étendus. Il vous permettra de mettre en évidence l'enchaînement des activités dans le temps, des dépendances existantes entre définitions de travail et leurs responsables, et de pointer les responsables par produit de travail.

APES: http://developer.berlios.de/project/show-files.php?group_id=1394



Cocoon

Cocoon est un des projets de la fondation Apache. Prenons un document XML et sa feuille de style XSL : pour l'afficher à l'aide d'un navigateur, le document devra être formaté. Et c'est justement là où intervient Cocoon du côté serveur (Cocoon fonctionne comme une servlet, et il faut donc lui créer sur le serveur tout un environnement comprenant la plomberie java). Il est aussi capable de générer en output du PDF ou du WML. Au-delà de cette application pratique, Cocoon est considéré comme une plate-forme de développement d'applications web à part entière. Il implémente la notion de "pipe-lines de composants", qui lui permet de construire une application Web de manière modulaire.

Apache Cocoon : <http://cocoon.apache.org/>

Dev-C++ / Codeblocks

Si vous développez en C++ et que vous êtes issus du monde Windows, vous chercherez probablement un équivalent à Visual C++ de Microsoft ou au Builder C++ de Borland. Sous Linux, il existe Dev-C++ de Bloodshed, un IDE destiné à la programmation en C/C++. Il est complet, performant, gratuit, et utilise GCC (*GNU Compiler Collection*) comme compilateur (il peut également être utilisé conjointement avec Cygwin sous Windows). Il n'est pas livré avec une librairie graphique, mais vous pouvez facilement créer votre interface avec wxWidget.

Seulement, Dev-C++ semble abandonné depuis février 2005. Mais heureusement, il existe un autre IDE qui a déjà pris le relais : Code::Blocks. Si vous partez de Dev-C++ vous ne serez pas très désorientés. Code::Blocks est un environnement de développement gratuit et multi plate-forme qui a été conçu autour d'une architecture de plug-ins. Il est par conséquent très facilement extensible. L'une de ses principales caractéristiques est qu'il est capable de s'interfacer avec un grand nombre de compilateurs (Visual C++ 2003, MinGW, Borland C++ 5.5, Digital Mars, Open Watcom), et peut aussi importer des projets VC++ et Dev-C++. Ses autres caractéristiques sont, dans le désordre : la compilation directe, ou via un makefile, le support des templates, la gestion des espaces de travail, le débogage avec GDB, la coloration syntaxique, l'auto complétation de code, l'exploration des classes, l'indentation automatique, et la gestion des listes de "TODO".

codeblocks : <http://www.codeblocks.org/>

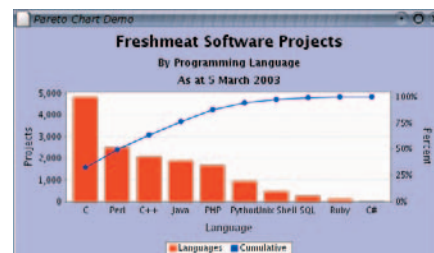
Dev-C++ : <http://www.bloodshed.net/devcpp.html>

Développement Java

Le développement sous Linux ne passe pas nécessairement par Java, mais ce dernier langage fédère une grande partie des développeurs professionnels officiant sous la plate-forme Linux. Citons dans le désordre quelques logiciels renommés de cette catégorie : Ant, l'outil de gestion de projets (<http://ant.apache.org/>), Jikes, le compilateur java d'IBM (<http://www124.ibm.com/develo-perworks/oss/jikes/>), Xdoclet, un outil de généra-

tion de code (<http://xdoclet.sourceforge.net/xdoclet/index.html>), Cactus, du projet Jakarta qui est un framework de test pour applications Web (<http://jakarta.apache.org/cactus/index.html>), Jalopy, un outil de mise en forme de code java (<http://jalopy.sourceforge.net/>), Avalon, un outil de développement de composants par design patterns (<http://avalon.apache.org/index.html>), Junit, un framework de test unitaire automatisé (<http://www.junit.org/index.htm>), JSwat, un débogueur graphique (<http://www.bluemarsh.com/java/jswat/>), SableCC, un générateur de compilateurs orientés-objets (<http://sablecc.org/>) et enfin GCJ le compilateur GNU pour Java (<http://gcc.gnu.org/java/>).

Dans cette catégorie, nous pouvons aussi placer d'innombrables bibliothèques, dont une des plus célèbres est JfreeChart, une librairie qui permet de créer des graphiques en Java. L'export est possible en PNG, JPEG mais aussi PDF ou SVG (<http://www.jfree.org/jfreechart/index.php>).



On n'oubliera pas les IDE phares : Eclipse et NetBeans.

Doxygen

Doxygen [contraction de docs (documents) et de gen (generator)] a été développé par Dimitri van Heesch, dans le but de générer de la documentation à partir d'un code source écrit dans un des langages suivants : C, C++, Java, Objective C, Python, IDL, PHP, C# et D). Pour y parvenir, il analyse le code ainsi que les commentaires qui s'y trouvent (et qui ont été rédigés avec une grammaire spécifique telle que l'ouverture des commentaires par /** ou encore l'indication des paramètres avec @param). La documentation qui en résulte est disponible dans le format HTML, mais aussi PDF ou RTF. Doxygen est devenu rapidement fort populaire au sein de la communauté

des développeurs, car rien n'est plus ennuyeux pour un développeur que de rédiger de la documentation. Or, si les commentaires ont été judicieusement écrits, tous les ingrédients nécessaires à une bonne documentation se retrouvent déjà dans le code. En outre, il est aussi possible d'extraire un minimum de documentation à partir d'un code source non documenté. En effet, Doxygen structurera les informations au sujet des prototypes et des fonctions, des variables, de la liste des fichiers inclus, de la structure des données. Il générera un diagramme des classes, un graphe d'appels, etc. KDE utilise Doxygen pour générer la documentation de son API. Doxygen est publié sous licence GPL.

Doxygen : <http://www.stack.nl/~dimitri/doxygen/>

Gambas

Gambas est un interpréteur pour Linux du langage Basic, avec des extensions orientées objet, et proche de Visual Basic. Gambas signifie « *Gambas Almost Means Basic* » (Gambas veut presque dire basic). Son auteur, Benoît Minisini, vit à Paris et est passionné par l'informatique depuis son adolescence. Il a découvert le langage BASIC sur CPC Amstrad 464 et puis plus tard sur Atari 520 STE. Il a créé ce projet dans le but, non pas de créer un clone de Visual Basic, mais de concevoir un environnement de développement graphique plus simple dans sa syntaxe, plus stable, plus cohérent et plus léger (que celui de VB). Gambas est distribué sous les termes de la licence GPL.

Avant tout, il s'agit d'un compilateur BASIC avec extension objet. Chaque programme créé avec Gambas est composé d'une série de fichiers, et ceux-ci décrivent des classes en terme de programmation objet. Les classes sont compilées, puis exécutées par un interpréteur. Cela fait un peu penser au comportement de Java ou encore de VB.NET : il y a « compilation », puis « interprétation » du code compilé. Cerise sur le gâteau, l'environnement de développement de Gambas est écrit en... Gambas... Un projet, composé de x fichiers, est stocké dans un répertoire unique. À la compilation, ces fichiers seront rassemblés en un seul exécutable unique. Seules les classes sont compilées, les références externes sont résolues dynamiquement lors de l'exécution. Gambas possède une architecture ouverte : n'importe qui peut écrire une extension au langage sous la forme d'une bibliothèque de fonctions partagées. L'interface graphique se présente sous

la forme d'un composant, et est en principe indépendante du kit graphique. Gambas utilise pour l'instant le toolkit graphique QT et permet de créer rapidement une application KDE, utilisant si nécessaire une base de données telle que MySQL. Il est très facile de déboguer en exécutant pas à pas chaque ligne. Vous pouvez à tout moment visualiser le contenu des variables locales, de la pile des appels, ouvrir une console, évaluer une expression. Le BASIC de Gambas est moderne avec des structures comme CATCH...FINALLY pour gérer les erreurs.

Gambas : <http://gambas.sourceforge.net/>

Lazarus

Si nous effectuons encore une fois un parallèle avec ce qui existe sous Windows, il existe en France une grande base de développeurs utilisant Delphi de Borland. Lazarus sous Linux (mais aussi sous Windows) se base sur le langage de programmation Free Pascal et est capable de reprendre carrément des projets Delphi/Kylix ! Pour être plus exact Lazarus est une application libre dont les projets sont à 99 % compatibles avec Delphi. Lazarus est bien ce que l'on appelle un outil de développement rapide, car non seulement il comporte un IDE, mais aussi un concepteur d'interface graphique (par ajout de composants et de formes que l'on glisse/dépose sur un formulaire). Chaque composant de l'interface est ensuite modifiable (visualisation/édition des propriétés, resizing, association d'événements, etc.). Vous pouvez passer de Linux à Windows par simple recompilation du projet.

Lazarus : <http://www.lazarus.freepascal.org/>

MTASC

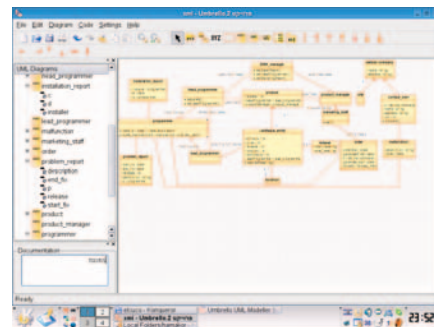
MTASC (Motion Twin ActionScript Compiler) est un compilateur libre ActionScript2, placé sous licence GPL, léger et très rapide (plus rapide que celui de Macromedia). Il permet de générer un fichier SWF à partir de classes ActionScript2, ou même de recompiler un SWF déjà généré par MM Flash (le compilateur MTASC étant plus rigoureux et plus rapide). Grâce à MTASC il est désormais possible de compiler du Flash sous Linux.

Umbrello et BOUML

Umbrello, un modèleur UML sous licence GPL, intégré à l'environnement de bureau KDE. Il vous

aidera à créer des diagrammes UML tels que les diagrammes de classes, de séquences, de collaborations, de cas d'utilisation, d'états, d'activités, de composants, de déploiements et enfin de relations entre entités. Il est possible d'exporter le tout dans un format image ou encore de générer automatiquement des diagrammes à partir du code source. Mais la caractéristique la plus intéressante de ce logiciel reste la génération du code source dans de nombreux langages cibles comme le C++, Java, PERL, Ruby, Python ou PHP.

BOUML est un logiciel modèleur plus récent qu'Umbrello qui est rapide et peu gourmand en mémoire. Il est possible d'étendre ses fonctionnalités par l'ajout d'une série de modules plug-out externes écrits en C++ ou Java. Vous pouvez générer du C++, du Java et de l'IDL, mais aussi effectuer du reverse C++ et Java ! Enfin, cerise sur le gâteau, il vous est loisible d'importer des projets Rose (modèleur commercial).



Umbrello : <http://uml.sourceforge.net/index.php>

BOUML : <http://bouml.free.fr/>

WxWidgets

Dans sa boîte à outils, le développeur Linux aura besoin d'un toolkit graphique de programmation d'interfaces. WxWidgets, créé en 1992 par Julian Smart, est un de ceux-ci. À l'origine, son nom était wxWindows : W pour Windows et x pour X, mais plus tard, x fut abandonné au profit de GTK+ et en 2004 le nom fut changé pour éviter un conflit juridique avec Microsoft. Il s'agit d'un logiciel libre multi plate-forme qui possède comme principale caractéristique de conserver l'apparence native de chacun des sous-systèmes sous-jacents utilisés. Autrement dit, WxWidgets ne tentera pas d'obtenir le même rendu sous Linux ou sous Macintosh, ou encore sous Windows CE, lorsqu'il doit, par exemple, y afficher une fenêtre. L'utilisateur se sentira chez lui, car WxWidgets n'imposera pas une interface utilisateur identique. Cette bibliothèque est écrite en C++, mais il existe de nom-



breux wrappers vers de nombreux autres langages tels que le Python (wxPython), le C#.NET (wx.NET), le Perl (wxPerl), le Java (wxJava), etc. De très nombreux logiciels s'appuient sur ce toolkit comme Codeblocks. wxWidgets est très riche fonctionnellement, et comme il existe depuis plus d'une décennie, il possède une très grande quantité de documentations et tutoriaux.

wxwidgets : <http://www.wxwidgets.org/>

Les langages fonctionnels Open Source

La programmation fonctionnelle est assez déconcertante pour un programmeur habitué à la programmation impérative. En effet, un langage fonctionnel n'utilisera pas le mécanisme classique de l'assignation, mais s'attachera à n'appeler qu'une seule fonction qui contiendra elle-même au besoin d'autres fonctions imbriquées. L'ancêtre du langage fonctionnel est le Lisp (« list processing ») qui date de la fin des années 50 ! Dans les années 80, il est devenu le langage de référence dans le domaine de l'intelligence artificielle. Il existe des variantes plus modernes comme le Scheme (qui date des années 70 et dont une implémentation sous Linux est stklos de l'université de Nice), et le Common Lisp (né dans les années 80). Sous Linux, l'implémentation la plus répandue est le CMU Common Lisp (CMUCL) de l'université de Carnegie Mellon, ainsi que le GNU Common Lisp (GCL), le compilateur Lisp du projet GNU.

Erlang est un autre langage fonctionnel qui a été placé sous licence Open Source en 1998. Il est fort utilisé dans le domaine des télécommunications car il possède un atout de taille : des interfaces avec d'autres langages comme le Java ou le C++.

Le langage Objective Caml (Ocaml) est un langage de programmation avancé qui convient aussi bien à la programmation fonctionnelle qu'impérative. Dans la même veine, il existe Oz, qui a été développé par trois universités européennes, et qui permet de programmer aussi bien de manière impérative que fonctionnelle. Notez qu'il existe un mode Oz conçu pour Emacs. Enfin, il existe d'autres fonctionnels tels que ML ou Haskell qui peuvent également tourner sous Linux.

CMUCL : <http://www.cons.org/cmucl/>

GCL : <http://www.gnu.org/software/gcl/gcl.html>

STKLOS (Scheme) : <http://www.stklos.org/>

ERLANG : <http://www.erlang.org/>

OCAML : <http://caml.inria.fr/download.fr.html>

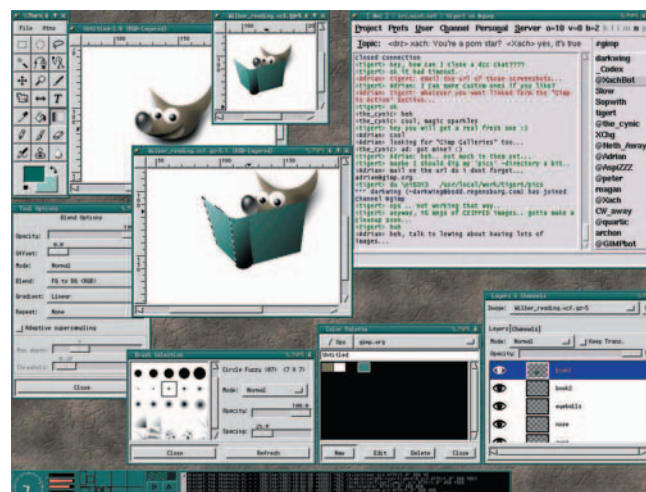
Oz : <http://www.mozart-oz.org/>

Compilateur mlton (ML) : <http://mlton.org/>

Hugs (Haskell) : <http://www.haskell.org/hugs/>

Toolkits Open Source

GTK+ (*The GIMP Tool Kit*) est un ensemble de trois bibliothèques de fonctions C qui ont été développées, à l'origine, pour les besoins du seul logiciel de traitement d'images «The GIMP». GTK+ est libre et permet de construire facilement des interfaces utilisateurs fenêtrées. Il existe de très nombreuses boîtes à outils de ce genre sous Linux, dont les plus populaires sont sans doute GTK+, Qt et wxWindow.



Le point fort de GTK+ est de permettre au programmeur de simuler une apparence similaire à d'autres bibliothèques graphiques, telles que Qt ou NeXTSTEP (il suffit de configurer un moteur d'aspect différent). Les trois bibliothèques en question sont glib (contenant des fonctions de manipulation de la gestion

mémoire, de chaînes de caractères ou de création de threads), gdk (qui communique avec le serveur X) et enfin gtk (qui s'appuie sur les deux précédentes pour proposer une série de widgets, dans le but de créer une interface).

Un autre toolkit OpenSource intéressant est FLTK (Fast Light Toolkit). « Fulltick » a été conçu pour être lié de manière statique. Le code non commun aux diverses plates-formes est de l'ordre de 10 %, assurant un comportement similaire d'une grande compatibilité, quel que soit l'OS cible. FLTK fait appel directement à Xlib (Unix), ou aux API WIN32 (Windows), ou encore à Carbon (Mac OS X). En résumé, vous n'avez pas besoin de vous soucier des dépendances lors du déploiement de votre application.

GTK : <http://www.gtk.org/>

FLTK : <http://www.fltk.org/>

Pear

PEAR, projet majeur de la communauté PHP, signifie « PHP Extension And application Repository », et se rapproche du système CPAN de Perl. PEAR se base sur un standard de programmation assez strict, pour permettre de partager rapidement ce que fait un code donné. Stricto sensu, il s'agit d'une collection pour PHP de classes en Open Source (gestion de base de données, e-commerce, Web Services, etc.) mises à disposition sous la forme d'un système de distribution pour composants réutilisables. PEAR s'installe en pointant son navigateur à l'adresse go-pear.org. Sous Linux cela donne : `#lynx -source http://go-pear.org/php-q`

Après l'installation, vous vous retrouvez avec la commande « pear ». Celle-ci permet d'installer une nouvelle extension, de lister les extensions déjà installées, de lister l'ensemble des packages existants que l'on peut installer, etc.

PEAR : <http://pear.php.net/>

■ **Xavier Leclercq**

Xavier.Leclercq@programmez.com

Quelques autres outils

Nom	Adresse Internet	Description
Abbot	http://abbot.sourceforge.net/doc/overview.shtml	Test java en environnement graphique
Anjuta	http://anjuta.sourceforge.net/	L'environnement intégré de développement destiné aux développeurs GNOME
ArgoUML	http://argouml.tigris.org/	Modéleur au même titre qu'Umbrello et BOUML
Bugzilla	http://www.mozilla-europe.org/fr/products/bugzilla/	Logiciel implémentant le suivi de bogues (ou de « tickets ») via une interface web. Il est utilisé par de nombreuses organisations (GNOME, KDE, RedHat, Mandriva) pour suivre le développement de nombreux logiciels, tant propriétaires qu'Open Source
CVS	http://www.cvshome.org/	Gestionnaire de suivi de code source (mûr et stable)
DejaGNU	http://www.gnu.org/software/dejagnu/	Framework qui permet d'automatiser le test de programmes C et C++
Emacs	http://www.gnu.org/software/emacs/emacs.html	La référence en matière d'éditeur
GCC (GNU Compiler Collection)	http://gcc.gnu.org/	Collection de logiciels libres intégrés capables de compiler divers langages de programmation, dont C (gcc), C++ (G++), Java (GCJ), Ada (GNAT) et Fortran (Gfortran)
Gcov	http://gcc.gnu.org/onlinedocs/gcc-3.0/gcc_8.html	Logiciel de tests de taux de couverture
gdb	http://www.gnu.org/software/gdb/gdb.html	Le GNU Debugger est le débogueur standard du projet GNU. Il est portable et fonctionne pour plusieurs langages comme le C, le C++ et le Fortran
Gengetopt	http://www.gnu.org/software/gengetopt/gengetopt.html	Gestionnaire des arguments passés en ligne de commande (parser C/C++)
gettext	http://www.gnu.org/software/gettext/	Gestionnaire de la localisation
gforge	http://gforge.org/	Environnement de développement collaboratif Web style sourceforge
GNU Autoconf	http://www.gnu.org/software/autoconf/	Logiciel servant à produire des scripts shell qui configurent automatiquement le code source d'un logiciel pour l'adapter à divers systèmes
GNU Automake	http://sources.redhat.com/automake/	Logiciel générant des makefiles portables qui peuvent être utilisés par make pour compiler des programmes
Guile	http://www.gnu.org/software/guile/guile.html	Interpréteur pour le langage Scheme
JBoss IDE	http://jboss.sourceforge.net/jbosside/updates	Plug-in pour Eclipse, facilitant le développement pour JBoss (EJB, Servlet, WebServices)
JUnit	http://junit.sourceforge.net/	Tests unitaires java
Kcachegrind	http://kcachegrind.sourceforge.net/cgi-bin/show.cgi	Outil de profilage C/C++
KDevelop	http://www.kdevelop.org/	L'environnement intégré de développement destiné aux développeurs sous KDE
Libtool	http://www.gnu.org/software/libtool/	Logiciel servant à créer des bibliothèques portables par encapsulation des dépendances par rapport à chaque plate-forme
Maven	http://maven.apache.org	Outil open source, issu d'Apache Jakarta, qui permet à l'instar d'ANT d'automatiser la gestion et la construction d'un projet (java ou autre)
OpenMDX	http://www.openmdx.org/	Gestionnaire CRM
Planner	http://planner.imendio.org/	Gestionnaire de suivis de projets. Anciennement Mr. Project
SQLite	http://www.sqlite.org/	Bibliothèque écrite en C proposant un moteur de base de données SQL au standard SQL92
Struts	http://struts.apache.org/	Environnement de développement Java combinant JSP et Servlets, et respectant l'implémentation MVC (Modèle-Vue-Contrôleur)
Subversion	http://subversion.tigris.org/	Gestionnaire de suivi de code source
Valgrind	http://www.valgrind.org/	Outil de débogage C/C++



2006

Tous les outils et toutes
les applications
GNU/Linux, Open Source
et logiciels libres pour
l'entreprise :

EMBARQUÉ
STOCKAGE
E-COMMERCE
INTERNET
INTRANET
SERVEURS
SÉCURITÉ
SERVICES
DÉVELOPPEMENT
GRILLES
GESTION
BUREAUTIQUE



CNIT - PARIS LA DÉFENSE
31 JANVIER, 1&2 FÉVRIER 2006

Avec



170 EXPOSANTS, 27 CONFÉRENCES, TABLES RONDES ET SÉMINAIRES

Pour exposer : linux@infopromotions.fr

Pour visiter et s'inscrire aux tutoriels et séminaires : www.solutionslinux.fr

Programmation de D-BUS sous Linux

KDE 3.4, qui est sorti en mars 2005, a été la première version à utiliser D-BUS selon les recommandations de Freedesktop. Freedesktop vise la collaboration entre différents projets issus du libre, dont principalement GNOME et KDE. Le but étant d'obtenir une meilleure interopérabilité entre les deux mondes en harmonisant leur infrastructure. D-BUS a été initié en 2002 par Havoc Pennington et Alex Larsson de Red Hat.



Havoc Pennington, est un des pères fondateurs de D-BUS.

D-BUS est au même titre qu'UDEV ou HAL (*Hardware Abstraction Layer*), une technologie récente (Udev est un système de fichier remplaçant Devfs sur les noyaux Linux de la série 2.6 qui gère les périphériques dans /dev. La couche d'abstraction matériel (HAL) est une technologie qui permet de dresser une liste des périphériques d'un système).

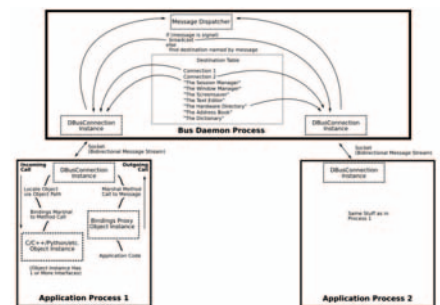
D-BUS est un système de communication inter-processus (IPC) fournissant un mécanisme simple, mais puissant, permettant à des applications (processus concurrents ou distants) de dialoguer, c'est-à-dire, de transmettre des informations entre elles et de faire appel à des services. D-BUS a été conçu de zéro (il ne se base pas sur le code de DCOP), pour répondre aux besoins d'un système d'exploitation moderne comme Linux. Le but initial est de remplacer CORBA et DCOP, les systèmes utilisés respectivement dans GNOME et KDE. Ces concepteurs espèrent que D-BUS débouchera sur un mécanisme unifié qui sera employé intensivement par les deux bureaux.

D-BUS est à la fois chargé d'échanger des données dans le but de réaliser l'intégration d'applications, mais aussi d'envoyer des signaux dans un but de synchronisation. L'unité de base de l'IPC dans D-BUS est le message qui se compose d'en-têtes (des métadonnées) et des données proprement dites. Le format de ce message est binaire, typé, entièrement aligné et simple, et non un jet aléatoire de bytes,

comme c'est le cas avec d'autres systèmes IPC (COM, SAVON, XML-RPC, XPCOM). Il est fourni avec un daemon qui route les messages entre les processus sur un bus spécifique. Avec ce système de bus, un processus est capable de dialoguer avec une ou plusieurs applications en même temps. Les applications peuvent envoyer, où se mettre à l'écoute de divers événements sur un bus. Il existe deux types de bus : un bus système et un bus de session. Le bus système est global et tous les utilisateurs du système peuvent communiquer au-dessus de celui-ci, avec les permissions appropriées, permettant de réagir aux événements propres au système. Le bus de session est par contre seulement employé par un utilisateur particulier. Les messages sont envoyés aux objets en utilisant des noms de chemin tels que /org/cups/printers/queue.

D-BUS supporte des types de messages multiples (signaux, appels de méthodes, messages de retour et messages d'erreur). Ces signaux simples et asynchrones avertissent qu'un événement spécifique s'est produit. Les messages de type appel de méthode permettent à une application d'invoquer une méthode sur un objet distant. Un message de retour d'une méthode fournit la valeur de retour de l'invocation d'une méthode, tandis que les messages d'erreur fournissent des exceptions en réponse à une invocation de méthode. D-BUS est entièrement typé : les types valides incluent le byte, le booléen, l'entier sur 32-bit (signé et non signé), le nombre entier sur 64-bit (signé et non signé), la virgule flottante à double précision et enfin, la chaîne de caractères. D-BUS est sécurisé dans le sens où il implémente, selon un protocole basé sur SASL (*Simple Authentication and Security Layer*), l'authentification d'une connexion point à point (ce que l'on appelle « de pair-à-pair »,

dans le cas d'applications qui veulent communiquer directement de l'une vers l'autre). Ce qui signifie que la lecture et l'écriture des messages d'une interface spécifique vers les bus sont régies par un système de sécurité dont un administrateur peut commander l'accès. Le daemon D-BUS a été pensé en tenant compte



de ce problème de sécurité.
Schéma de principe de fonctionnement de D-BUS.

Pourquoi D-BUS et non DCOP ?

DCOP (*Desktop Communication Protocol*) est une technologie KDE qui est plus qu'un mécanisme IPC, car cette technologie peut servir à contrôler des applications en utilisant la ligne de commande. DCOP possède toute une série d'outils annexes, comme dcoptest qui permet le démarrage d'une application KDE le renvoi de son ID, dcofind qui sera chargé de trouver une application KDE en fonctionnement, dco-pref qui crée une référence DCOP, dcoobject qui obtient une ID d'un client selon une référence, et enfin dcoobject qui extrait des objets selon une référence. Ajoutons que DCOP est stable et utilisé depuis des années tandis que DBUS n'est pas encore arrivé à maturité (version < 1.0).

L'inconvénient majeur de DCOP est qu'il ne définit pas lui-même les types de base, mais réutilise celui du langage utilisé par les applications. Avec D-BUS ce n'est pas le cas.

Autrement dit, l'utilisation de Dcop dans le cadre de KDE implique l'emploi du C++ avec QT, ce qui entraîne obligatoirement l'emploi des types de QT. D'où une communication difficile avec des applications non QT. Si Dbus n'est utilisé par gnome que pour les notifications système, à terme il devrait s'intégrer à toutes les applications Gnome.

Le bus système de D-BUS est simple, partagé par le système entier et tient compte de la propagation des événements. Il améliorera grandement l'intégration des logiciels avec le kernel. Les événements sont envoyés de manière asynchrone pour communiquer du noyau vers l'espace utilisateur (disque plein, file d'attente d'impression vide, etc.). Le kernel est capable d'envoyer des signaux D-BUS par l'intermédiaire de sysfs (les chemins de sysfs sont facilement traduisibles en chemins D-BUS). D-BUS et le développement du kernel ont été fusionnés à partir du noyau 2.6.10-rc1.

L'API D-BUS (écrite en C) est plutôt de bas niveau mais une grande variété de langages peuvent s'y connecter via des wrappers, comme Python, Perl, Ruby, Java et C# Mono. En fait, des types complexes comme les tableaux, des dictionnaires, peuvent être codés dans un message. Ceci permet la communication entre applications écrites dans un éventail de langages très grand. En programmation, un objet D-BUS peut être considéré comme n'importe quel autre objet, excepté qu'il est basé, non pas sur des adresses mémoire, mais sur des chemins d'objet. Les chemins d'objet prennent la forme d'une chaîne de caractères semblable aux chemins de fichiers tels qu'on les rencontre sous Unix (comme /org/freedesktop/DBus). Les interfaces sont également prévues par D-BUS et elles permettent au même nom de méthode d'être employé plus d'une fois. Seulement, pour acheminer un message vers son destinataire, les objets et les interfaces ne sont pas les seuls éléments à tenir en ligne de compte. Nous devons encore comprendre le concept de service. Un service est un endroit unique sur le bus que possède une application (et qui a été enregistré par l'application et qu'elle seule peut libérer). Si un autre processus tente d'utiliser un service ayant déjà été enregistré comme tel par une application (et non libéré), il recevra un message d'erreur déclarant que le service n'est pas disponible. Concrètement, par exemple, au lieu de déter-

miner l'existence d'un périphérique en testant un fichier vous pouvez interroger à la place un service d-bus.

Bindings Mono

Nous avons rencontré pas mal de problèmes en essayant de faire tourner ruby ou python avec la version installée par défaut sous SuSE 10. Comme D-BUS est en continuel développement, mieux vaut ne se fier qu'à la dernière version officielle. Nous vous recommandons vivement de compiler dbus en partant des sources. Nous avons récupéré la version 0.60 datant du 30 novembre 2005. Après avoir décompressé l'archive vous devez indiquer, lors de la configuration, quel langage vous allez utiliser :

```
/home/xl/dbus-0.60 # ./configure --enable-mono
checking for a BSD-compatible install...
/usr/bin/install -c
...
```

Entre parenthèses, notre version de mono est la 1.1.8.0

```
#mcs -version
Mono C# compiler version 1.1.8.0
```

et le noyau est un 2.6.13-15 :

```
#uname -a
Linux backup 2.6.13-15-default #1 Tue Sep 13
14:56:15 UTC 2005 i686 i686 i386
GNU/Linux
```

Vous devez commencer par installer les paquets nécessaires, si ce n'est déjà fait : au minimum, dbus-1, dbus-1-devel, dbus-1-glib puis dbus-1-gtk, dbus-1-java, dbus-1-mono, dbus-1-python ou dbus-1-qt, selon le langage.

Les bindings mono consistent en un wrapper (dbus-sharp) autour de l'API C de D-BUS. Comment le lien est-il établi ? Le wrapper crée un proxy pour le client qui désire accéder aux objets d-bus en utilisant la Reflection, dans le but de créer à la volée un assemblage qui contiendra des sous-classes qui seront réellement exploitées (mécanisme de marshaling/démarshaling).

Concrètement, nous tentons de réaliser une connexion avec le bus de sessions (Connection connection = Bus.GetSessionBus();) puis nous

créons un service (service service = new Service(connection, "Programmez.com");) qui sera à l'écoute des requêtes en provenance de programmez.com. Puis nous créons un objet (monObjet) et l'enregistrons de telle manière que dbus-sharp sache l'exporter (grâce aux définitions de l'interface et de sa méthode virtuelle). Nous avons inclus cette définition dans notre code serveur.cs, mais en fait, comme il est redondant avec le client, il est préférable de le placer dans un fichier source du style Interface.cs.

serveur.cs

```
namespace Programmez
{
    using System;
    using DBus;
    using Gtk;

    [Interface("Programmez.com.NumeroEnCours")]
    public class MonObjet
    {
        [Method]
        public virtual string Echo(string message)
        {
            return "Réponse: " + message;
        }

        public class MonServeur
        {
            public static int Main(string [] args)
            {
                Application.Init();

                Connection connection = Bus.GetSessionBus();
                Service service = new Service(connection, "Programmez.com");
                MonObjet monObjet = new MonObjet();

                service.RegisterObject(monObjet, "/Programmez/com/NumeroEnCours");

                Application.Run();

                return 0;
            }
        }
    }
}
```

Du côté client, nous commençons par établir une connexion au bus session, mais au lieu de créer un service, nous en recherchons un qui existe déjà (Service service = Service.Get(connection, "Programmez.com");). Puis nous demandons au service de nous retourner l'objet qui a été créé (MonObjet monObjet = (MonObjet)). À partir de ce moment, nous le traitons comme n'importe quel autre objet en appelant sa méthode Echo.

client.cs

```
namespace Programmez
{
    using System;
    using DBus;

    [Interface("Programmez.com.NumeroEnCours")]
    public class MonObjet
    {
        [Method]
        public virtual string Echo(string message)
        {
            return "Réponse: " + message;
        }
    }

    public class MonCient
    {
        public static int Main(string [] args)
        {
            Connection connection = Bus.SessionBus();
            Service service = Service.Get(connection, "Programmez.com");
            MonObjet monObjet = (MonObjet) service.GetObject(typeof(MonObjet), "/Programmez/com/NumeroEnCours");

            System.Console.WriteLine(monObjet.Echo("N° 83 Février 2006"));

            return 0;
        }
    }
}
```

Nous avons aussi créé un petit exécutable, inspiré de la documentation d-bus de Mono, qui se met à l'écoute du bus et ainsi, qui permet

de détecter la création du service et l'échange des messages :

EcouteDuBus.cs

```
namespace Programmez
{
    using System;
    using DBus;
    using Gtk;

    public class EcouteBusBus
    {
        static void
        ChangementNomProprietaire (string Nom,
        string AncienProprietaire,
        string NouveauProprietaire)
        {
            if
            (AncienProprietaire == "")
            Console.WriteLine ("{0} Créé par {1}",
            Nom, NouveauProprietaire);
            else if
            (NouveauProprietaire == "")
            Console.WriteLine ("{0} Libéré par {1}",
            Nom, AncienProprietaire);
            else
            Console.WriteLine ("{0} Transfert de {1} à {2}",
            Nom, AncienProprietaire,
            NouveauProprietaire);
        }

        public static int Main
        (string [] args)
        {
            Application.Init
            ();

            Connection
            connection =
            Bus.SessionBus ();

            BusDriver driver
            = BusDriver.New (connection);

            driver.NameOwnerChanged += new
```

```
NameOwnerChangedHandler
(ChangementNomProprietaire);

Console.WriteLine ("A l'écoute d'un change
ment de propriétaire...");

Application.Run ();

return 0;
}
```

Voyons un peu comment il réagit :

```
xl@Tpol:~/mono> mono EcouteDuBus.exe
```

À l'écoute d'un changement de propriétaire...

D'abord nous le lançons. Puis nous exécutons le serveur qui créera le service :

```
:1.11 Créé par :1.11
Programmez.com Créé par :1.11
```

Enfin nous exécutons le client :

```
mono client.exe
Réponse: N°83 Février 2006
```

qui provoque l'affichage supplémentaire suivant sur la console contrôlant l'écoute du bus :

```
:1.12 Créé par :1.12
:1.12 Libéré par :1.12
```

Trois points très importants sont ici à souligner :

- vous ne devez en aucun cas exécuter le serveur ou le client en mode super-utilisateur (sous peine de plantage) ;
- la compilation est réalisée comme ceci :

```
mcs serveurur.cs -r:/usr/lib/mono/dbus-sharp/dbus-sharp.dll -r:/usr/lib/mono/gtk-sharp/gtk-sharp.dll ;
```

- l'exécution passe par mono : mono serveurur.exe.

Bindings Ruby et Python

Pour ruby, vous devez télécharger le paquetage sur rubyforge : #wget <http://rubyforge.org/frs/download.php/2754/dbus-ruby-0.1.10.tar.gz>.



bitdefender®

SECURE YOUR EVERY bit

www.bitdefender.fr



ÉDITIONS
PROFIL

Toujours un coup d'avance !



0 seconde !

C'est le temps nécessaire à la technologie proactive d'analyse comportementale HiVE (Heuristic in Virtual Environment) pour vous protéger contre la propagation d'un code malveillant encore inconnu. Son efficacité a été démontrée face, notamment, aux différentes versions du vers Zotob ou encore contre les trojans ayant exploité le rootkit Sony DRM.

L'antivirus le plus réactif du marché

BitDefender a été reconnu comme l'éditeur proposant les mises à jour les plus rapides à ses utilisateurs face aux nouvelles menaces, par AV-Test, Virus Bulletin et PC Welt.*

Une gamme complète pour les environnements Windows, Linux et FreeBSD

Au niveau des stations de travail comme des serveurs, BitDefender propose une protection complète pour Windows, Linux et FreeBSD. Des modules dédiés s'intègrent directement dans MS ISA Server, MS Sharepoint Portal Server, MS Exchange, Lotus Notes, SendMail, QMail, PostFix, Exim, Insight Server, etc. Une console de déploiement et d'administration, Enterprise Manager, ainsi qu'un module Webmin permettent de gérer l'ensemble de façon centralisée.

(*) Temps de réponse moyen face à l'apparition d'un nouveau code malveillant. 2004.

Des services complets qui vous assurent une totale tranquillité

Mises à jour de la base virale toutes les heures, mises à jour en temps réel par PUSH UPDATE vers vos serveurs, mises à jour majeures incluses, accès aux laboratoires BitDefender pour la soumission de fichiers suspects, support technique par téléphone et par email en français, ainsi que par chat online 7j/7 – 24h/24, sont autant de garanties de tranquillité face aux nouvelles menaces.

Une technologie innovante et récompensée :



Pour plus d'informations ou pour tester nos solutions, visitez www.bitdefender.fr

Contactez-nous par e-mail : bitdefender@editions-profil.fr ou par Téléphone au +33 (0)1 47 35 72 73.

Édité par Éditions Profil

49 rue de la Vanne - 92120 Montrouge

Voici un code client d'exemple très proche de Python :

```
#cat example-client.rb

#!/usr/bin/env ruby
require 'dbus'

bus = DBus::SessionBus.new
remote_service = bus.get_service("org.design
fu.SampleService")
remote_object =
remote_service.get_object("/SomeObject",
"org.design
fu.SampleInterface")

hello_reply_list =
remote_object.HelloWorld("Hello from
example-client.rb!")
p hello_reply_list
```

Nous avons déjà évoqué D-BUS et Python dans un article à propos de Skype. Voici le code serveur :

```
serveur.py

#!/usr/bin/env python
import dbus
import gtk

def ListInterfaces(message):
    return ['eth0: Belkin Gigabit Desktop
Card', 'eth1: RealTek RTL8139eth0']

bus = dbus.SystemBus()
service =
dbus.Service("com.Liste.InterfacesReseaux",
bus=bus)
object = dbus.Object("/", service,
[ListInterfaces])
gtk.main()
```

La définition du service réalisé dans le fichier de configuration idoine sous /etc/dbus-1/system.d/.

/etc/dbus-1/system.d/service.conf

```
<busconfig>
<policy user="xl">
<allow
own="com.Liste.InterfacesReseaux"/>
</policy>
<policy context="default">
<allow
receive_interface="com.Liste.Interfaces
Reseaux"/>
</policy>
</busconfig>
```

L'utilisateur xl est autorisé à fournir ce service et n'importe qui peut se connecter à cette interface. Relancez le service dbus

(/etc/init.d/dbus restart) et exécutez le serveur :

client.py

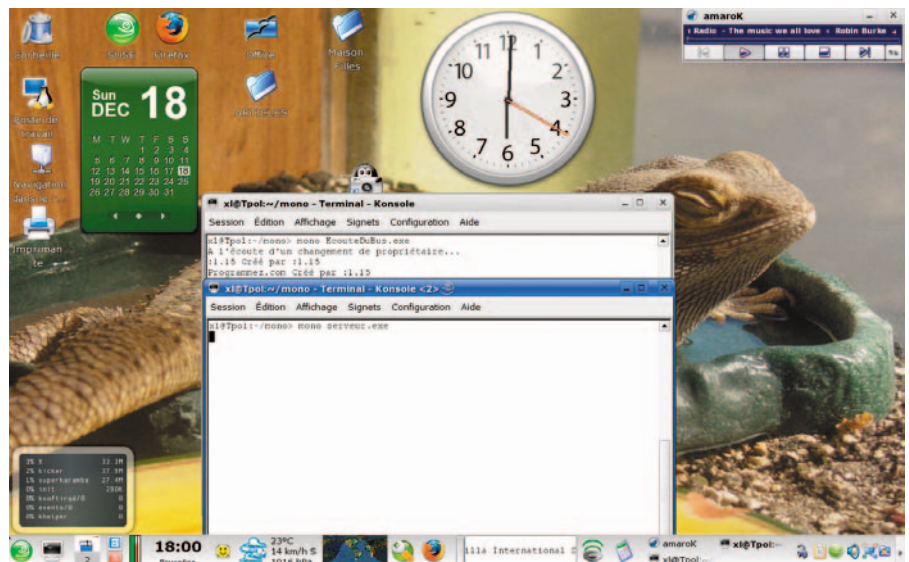
```
import dbus
bus = dbus.SystemBus()
remote_service =
bus.get_service('com.Liste.InterfacesReseaux')
remote_object = remote_service.get_object('/',
'com.Liste.InterfacesReseaux')
print remote_object.ListInterfaces()
```

En exécutant python client.py nous obtenons bien :

```
#python client.py
['eth0: Belkin Gigabit Desktop Card', 'eth1:
RealTek RTL8139eth0']
```

■ Xavier Leclercq

Xavier.Leclercq@programmez.com



Screenshot de mon bureau KDE 3.5 avec exécution en tâche de fond des exécutables Mono EcouteDuBus.exe et de serveur.exe.

Offre 2006

Abonnez-vous et recevez ce magnifique
Couteau suisse - clé USB 128 Mo
haut de gamme !

Pour le prix normal du magazine (prix au numéro), soit 65€ seulement !

Découpez le coupon d'abonnement, ou abonnez-vous en ligne : www.programmez.com



Sogeti : l'esprit de service



Portrait d'une des plus grandes SSII française : multiples activités, processus industrialisés. Mais un large panel de technologies et de métiers, et de réelles perspectives d'évolution.

Trois activités

Conseil en Hautes technologies : 2000 personnes autour de deux activités : bureau d'étude (R&D) dans les domaines de l'aéronautique, de l'électronique d'une part, informatique scientifique et technique d'autre part, logiciels embarqués notamment. Surtout en Ile-de-France et dans le Sud-Ouest. 500 recrutements en 2005.

Infrastructure services (IS) : services d'exploitation : conseil en architecture technique, infogérance sur site. 3 700 personnes en France, deux tiers à Paris, un tiers en région. 400 recrutements en 2005.

Application Services (AS) : conception, développement, test et maintenance d'applications d'informatique de gestion. 2 300 personnes, dont la moitié en régions. 700 recrutements en 2005.

Le groupe est organisé en quatre sociétés en France, Sogeti IS et Sogeti AS en Ile-de-France, Sogeti Technologies et Sogeti Régions (en charge des activités IS et AS).

La société est divisée en sept régions et une centaine d'agences comptant de 80 à 100 collaborateurs, qui sont autant d'unités opérationnelles, dans une trentaine de villes. L'objectif de cette organisation explique, Dominique Duflo, Directeur des ressources humaines est d'établir une relation de proximité avec les clients. Chaque agence établit son plan de recrutement.

Technologies utilisées : toutes ! des grands systèmes à l'AS/400, J2EE, .net ou Open Source, des progiciels de gestion intégrés à

l'EAI... des centres de services en régions sont spécialisés sur telle ou telle technologie : Java à Belfort, .net dans le sud de la France. L'Open Source n'est pas oublié, la société a mis à la disposition de la communauté une brique logicielle, Telosys, a signé un partenariat début 2005 avec la SS2L Linagora et a récemment remporté, avec Linagora et Bull, un contrat de 40 millions d'euros auprès du ministère des finances.

Les profils recherchés

La part des jeunes diplômés est d'environ 30 %. La moitié des postes proposés le sont en région. Sur la partie Application Services, Sogeti recherche surtout des profils nouvelles technologies ayant entre trois et cinq ans d'expérience et des experts sur l'EAI (Webmethods notamment), sur SAP et des ingénieurs d'étude J2EE ou .net. Sur la partie Infrastructure Services, les profils très recherchés sont ceux de services managers et d'architectes techniques. Enfin, sur le conseil en Hautes technologies, la société recrute des ingénieurs aéronautique ou électronique, ainsi que des profils

plus experts autour des logiciels embarqués ou des systèmes temps réel.

Au-delà de la maîtrise des technologies, Dominique Duflo souligne l'importance de « l'esprit de services : capacité de communication, capacité relationnelle », ainsi que l'adaptabilité et la mobilité intellectuelle. La maîtrise d'une deuxième langue est un préalable.

Autres compétences recherchées : le savoir-faire en matière de management de projets ou de

Carte d'identité

Depuis janvier 2006, Sogeti-Transiciel est devenu Sogeti tout court.

Plus de 9 000 personnes fin 2005 en France, sur un effectif total de 14 000 personnes.

Présent aux États-Unis, Au Royaume-Uni, en Suède, aux Pays-Bas, en Belgique, au Luxembourg, en Suisse, en Espagne 2 800 recrutements en 2005, dont 1 600 en France, correspondant à 1 000 créations de postes. Les chiffres devraient être similaires en 2006.

Comment Sogeti recrute ?

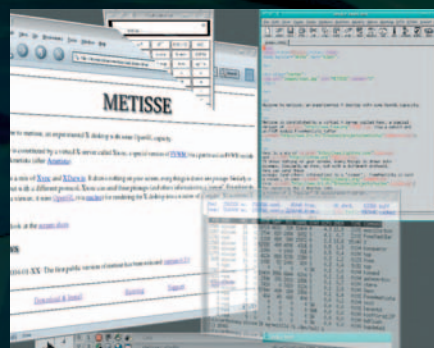
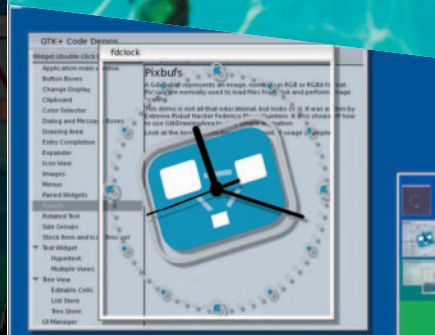
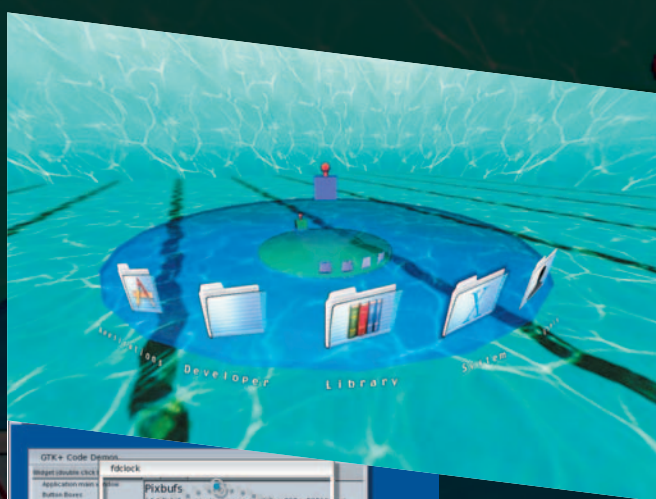
L'équipe dédiée au recrutement compte 25 personnes. Récemment, la société a organisé, en collaboration avec l'Apec, un tour de France qui lui a permis de rencontrer en trois jours et dans onze villes, 700 candidats dont le profil avait fait l'objet d'une pré-qualification.

La SSII organise également des soirées cooptation : les cooptés et les cooptants sont réunis dans un lieu où sont présentées les activités de la société. En avril, cette manifestation a réuni plus de 400 personnes et a donné lieu à plus de 60 recrutements. La cooptation n'est pas rémunérée, souligne Dominique Duflo. Les participants peuvent gagner un Ipod ou des cartes de cinéma par tirage au sort, mais la société mise avant tout sur sa capacité de conviction. Reconnu en décembre 2005, cet événement devrait être plus fréquent en 2006.

Le recrutement se fait en trois étapes : un entretien de sélection mené par les spécialistes du recrutement, le responsable technique de l'agence puis le directeur d'agence.

L'interface fait sa révolution

L'interface utilisateur n'a jamais été une sinécure pour le développeur. Elle l'est de moins en moins. Entre legacy, évolutions des demandes et nouvelles technologies, le développeur s'y perd !



Depuis de nombreuses années les éditeurs tentent de proposer de nouvelles visions de l'interface utilisateur, mais celle-ci n'a finalement que peu changé. Il y a plus de dix ans, Apple travaillait déjà à une révolution de l'interface utilisateur en élaborant toute une nouvelle philosophie de navigation dans les fichiers, les informations et son affichage. Le projet le plus abouti était Project X alias Hot Sauce. Les pages Web, fichiers, données étaient représentées sous forme d'arbres 3D virtuels et on y naviguait de niveau à niveau. Le tout, sur un format de métadonnées unique. Moins abouti, car très ambitieux à l'époque, le projet 3Dfinder était un bureau entièrement 3D.

IBM travaillait aussi à ce type d'interface. Trop révolutionnaire, trop en avance sur son époque. Depuis, l'évolution se fait par petites touches. MacOS X avec l'interface Aqua a montré une nouvelle voie. Dix ans plus tard, la situation a changé. Les nouvelles interfaces se répartissent à plusieurs niveaux. Sur les systèmes, comme dit plus haut, MacOS X a ouvert une nouvelle ère que les autres éditeurs suivent. On voit réapparaître des concepts 3D plus aboutis (et enfin utilisables) comme le projet Looking Glass de Sun. Microsoft travaille aussi sur les interactions 3D au niveau interface et Windows Vista bénéficiera de plus de 3D avec Aero.

Au niveau applicatif pur, les nouvelles interfaces se concentrent sur trois éléments majeurs :

- interface générée dynamiquement grâce à des langages de description selon le contexte
- interface web riche et interactive
- création et génération d'interface « classique » facilitée

Vous l'aurez compris, les nouvelles interfaces cachent une profonde modification de la perception de celles-ci par le développeur dans le développement. Et nous ne sommes qu'au début de cette profonde mutation et l'interface 3D n'en est que le volet le plus spectaculaire. **Programmez !** lève le voile sur cette évolution / révolution. ■ **François Tonic**

direction d'équipe. les compétences sectorielles ou fonctionnelles sont également appréciées. Dominique Duflo mentionne deux tendances : la maîtrise d'ouvrage et l'univers du test. « C'est un sujet auquel nous croyons beaucoup. Nous souhaitons valoriser ce métier qui n'est pas un métier de seconde zone. Nous développons cette filière, en relation avec des écoles ». Sogeti compte quelque 300 collaborateurs sur ce sujet, dont une bonne partie au

Benelux, les Pays-bas étant assez avancés sur ce point, et a développé une méthodologie sur le sujet : Test Management Approach (TMap), « adaptée à différentes situations, depuis le test des applications Web et des systèmes intégrés jusqu'aux applications basées sur des méthodologies de développement itératives telles que Rational Unified Process (RUP). Pour compléter l'approche TMap, Sogeti propose le modèle Test Process Improvement (TPI), métho-

dologie d'analyse et d'amélioration du processus de test global. »

Accompagner les nouveaux collaborateurs

L'intégration est accompagnée, les missions sont suivies par le directeur technique de l'agence et la société a mis en place un système de coaching des nouveaux par des anciens. Leur rôle : « entraîner le collaborateur à naviguer dans l'entreprise », l'aider à développer ses qualités humaines et comportementales, sans lien hiérarchique. Dominique Duflo explique que ce « cycle d'intégration dure deux ans ». Les nouvelles recrues sont formées à la culture maison, à la conduite de réunion, au comportement en clientèle. Des bilans de mission sont faits régulièrement, pour vérifier que les objectifs de la mission sont bien atteints. Un entretien annuel « évaluation et développement professionnel » permet de faire le point sur la carrière du collaborateur, ses aspirations et envies d'évolution, déterminer le cas échéant un programme de formation adapté.

Dissocier l'expérience de la compétence

Dominique Duflo explique que Sogeti dispose de « dispositifs de formations très structurés ». L'expérience en matière de conception, de développement ou d'intégration peut être complétée par une formation courte – 5 jours – nécessaire pour compléter une compétence pour une mission précise. Au-delà de ces formations ponctuelles, il est proposé aux collaborateurs de développer leurs compétences, en ingénierie de développement ou management de projet, en architecture ou dans le domaine du test, au travers d'une série de modules pouvant être suivi en une fois ou répartis. Troisième axe : la reconversion ou le repositionnement de collaborateurs autour de nouvelles offres : test ou Maîtrise d'ouvrage sectorielle par exemple : appréhension des besoins clients, formalisation, spécification, technique de pilotage ou de management de projet, renforcement des compétences dans des domaines fonctionnels : gestion, CRM, logistique, Business intelligence. Ces filières de reconversion sont au nombre de sept : testing, nouvelles technologies Java ou .net, management de projets, Maîtrise d'ouvrage, Business Intelligence, Unix-Oracle-ERP

■ Carole Pitras



Le sens de l'écoute

Eric Santelli est depuis deux ans responsable technique d'une agence

Sogeti parisienne, spécialisée dans le secteur bancaire.

Ingénieur informatique, il a commencé sa carrière classiquement, dans le développement et la conception d'applications. D'abord dans une banque, puis dans des SSII « de toutes tailles », durant cinq ans. Un parcours un peu à contre courant de l'évolution traditionnelle qui consiste à commencer dans une SSII, avant d'intégrer l'entreprise utilisatrice. Un choix qu'il revendique : « j'aime la façon dont on travaille en SSII, le côté challenge des appels d'offres ».

Il a rejoint Sogeti en 2003 pour prendre ce poste de responsable technique d'agence. Selon lui, cette SSII présente le double

gement de projet et des formations. Finalement, pas beaucoup de technique, si ce n'est qu'il doit assurer une veille technologique permanente afin de répondre aux appels d'offres.

Qualité requise pour son poste ? L'écoute. Ce qui lui plaît le plus : « arriver à mettre en phase les souhaits d'évolution des collaborateurs et les contraintes opérationnelles liées aux missions. Parvenir à combiner ces deux contraintes. » Il explique qu'être passé par l'étape du développement lui-même, lui permet de comprendre les problématiques des collaborateurs qu'il encadre et de détecter les difficultés de ceux-ci en cours de mission, afin de leur apporter le support nécessaire, l'aide d'un expert, ou une formation.

Depuis l'été dernier a été mis en place au sein de son agence un système de coaching : une dizaine de collaborateurs au

“J'aime la façon dont on travaille en SSII, le côté challenge”

avantage d'être, du fait de sa taille, présente chez tous les grands donneurs d'ordre, mais de par son organisation en agences, d'avoir su maintenir une relation de proximité avec clients et collaborateurs. Son rôle consiste aujourd'hui à recruter des collaborateurs, réaliser le staffing sur les missions, répondre aux appels d'offres et suivre des missions. Deux activités principales donc, les relations collaborateurs et les relations clients.

Veille technologique

Un rôle auquel il estime avoir été préparé par ses précédentes expériences de mana-

sein de l'agence servent de référents, sans lien hiérarchique, aux juniors, mais également à ceux qui sont en train de changer de fonction, par exemple un concepteur développeur qui est pour la première fois chef de projet. Lui-même sert de coach aux coachs, et s'appuie pour cela sur une expérience de parrainage qu'il avait vécu précédemment. Un premier retour d'expérience positif : une meilleure intégration, sur les aspects relationnels et réseau des nouveaux entrants. Rôle qui requiert aussi sens de l'écoute, disponibilité et envie de partager son expérience.

■ C.P.

Client riche : vers les interfaces agiles

Clairement avantagées par des coûts de déploiement extrêmement bas, les applications Web se sont imposées dans l'entreprise, supplantant pratiquement les applications graphiques classiques. Avec le recul, le prix à payer est lourd : ergonomie rudimentaire, temps de réponse pas toujours très bons, coûts de développement élevés.

Face à ce constat, il faut bien admettre que les applications graphiques d'antan n'avaient pas ces défauts là. Pourtant, on ne regrettera pas forcément leurs coûts de déploiement élevés, et leur maintenabilité médiocre, résultant de l'absence d'une séparation minimale entre la présentation, la logique métier et l'accès aux données. Pourtant, n'est-on pas légitimement en droit d'espérer le meilleur des deux mondes, avec en bonus, une vraie séparation en couche. Cette nouvelle approche, c'est le client riche. Combiné avec un **SOA** qui fournit les services métiers, le **client riche** est l'un des fondements d'un **système d'information agile**.

Ainsi, de nombreuses technologies sont en train d'émerger (ou de ré-émerger) pour nous permettre d'implémenter des clients riches. Parmi ces technologies on compte : **AJAX, Laszlo, XUL, Windows Presentation Foundation (WPF), Java Desktop Network Components (JDNC)**... Pour fonctionner, le client riche nécessite un **environnement d'exécution**. En conséquence, un critère de choix important est la disponibilité en standard de l'environnement d'exécution ou encore sa facilité de déploiement. Ainsi, JDNC nécessite Java, et WPF requiert .NET, AJAX fonctionne dans un navigateur et utilise JavaScript, Laszlo nécessite le plug-in Flash, très répandu.

Les 3 préoccupations

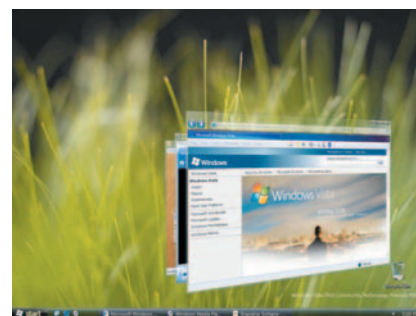
Les interfaces graphiques peuvent être décrites selon 3 axes de préoccupation: l'**aspect visuel** pur, le **binding** (relier les composants aux données affichées), et le comportement (réponse aux événements). Les deux premiers aspects peuvent bénéficier d'une **approche déclarative**.

En utilisant un langage spécifique (comme avec WPF, XUL, Laszlo ...), on peut décrire l'aspect graphique et le binding, sans avoir recours à du code, et surtout en se faisant assister d'**outils totalement visuels**. Seul l'aspect comportement reste implémenté par du code, dont la taille est réduite significativement. Le binding permet également de séparer la partie présentation, tout en bénéficiant d'une approche RAD, mais enfin structurée, et donc plus maintenable.

Les possibilités du **sous-système visuel** varient notablement selon la technologie de client riche. De ce point de vue, Laszlo et XUL sont bien lotis, WPF sera très richement doté, la galaxie AJAX reste, elle, très homogène, quant à JDNC, il est clairement en devenir.

Selon la technologie choisie, le **sous-système de binding** permet de relier l'interface graphique à divers types de données tels que : des documents XML (par exemple retournés par un service Web), des objets classiques (les fameux POJO pour Java), des données issues de la base de données, des données issues de fichiers texte tabulaires. Il peut également gérer le rafraîchissement de l'interface, lorsque les données bindées sont modifiées. Éventuellement, il permet aussi de gérer la synchronisation inverse, à savoir la mise à jour des données bindées lorsqu'elles sont modifiées via l'interface. Bien sûr, les possibilités offertes par le sous-système de binding diffèrent largement selon la technologie de client riche choisie. De ce point de vue, Laszlo n'est pas très bien loti, quant à AJAX, la situation est très différentes selon le framework. Pour l'heure, il semble que c'est bien WPF qui offrira, à sa sortie, le plus

de possibilités dans ce domaine. Quant à JDNC, il semble prometteur, mais il doit vraiment aboutir et faire ses preuves.



Dans le choix d'une technologie de client riche, les possibilités offertes par le sous-système visuel, et le sous-système de binding sont un critère fondamental, au moins aussi important que la facilité de déploiement.

De ce dernier point de vue, AJAX représente un choix de transition pertinent, mais les implémentations sont très variables et l'offre très éclatée. Les autres technologies nécessitent un environnement d'exécution, qui peut être installé via les infrastructures de déploiement automatisées de l'entreprise. Au final, si les technologies les plus prometteuses sont à venir, on peut faire dès maintenant du client riche avec AJAX, Laszlo, ou XUL. Une autre approche, souvent écarté d'office, consiste à développer des applications Windows Form, Swing ou SWT, et à rendre le déploiement transparent en utilisant ClickOnce ou Java WebStart.



■ **Nicolas DASRIAUX** est le directeur technique de NEOXIA, un cabinet d'architectes en systèmes d'information.

Abonnez-vous en ligne
www.programmez.com

Les interfaces faciles : vers la dématérialisation

L'interface utilisateur, ou IHM, ou encore GUI en Anglais, devient une composante complexe et pas toujours aisée à manipuler. Depuis quelques années, la tendance est de découpler fortement le code de l'interface. La dépendance entre les deux couches rendait mal aisée toute maintenance et évolution. Cependant, plusieurs tendances s'observent dans l'architecture d'interface et dans la manière de la créer et de la gérer, dont la dématérialisation pure et simple.

Aujourd'hui, qu'on le veuille ou non, deux tendances existent autour de l'interface : la GUI matérialisée (en dur) et la GUI dématérialisée (non palpable). Ces deux tendances cohabitent parfaitement aujourd'hui, car chaque type d'interface correspond à des projets précis.

L'interface matérialisée

Ce type d'interface est celui que l'on retrouve traditionnellement dans les RAD, dans les sites web. L'interface n'a jamais été une sinécure pour le développeur. Et elle l'est de moins en moins. Bien trop souvent, l'interface en dur demeure orientée code, technique, au lieu d'être réellement orientée utilisateur. C'est-à-dire que l'on part rarement des use cases, de l'utilisateur, pour créer l'interface de l'application, d'un site. Au mieux, l'interface s'adapte a posteriori. L'interface en dur possède des avantages et

des inconvénients. Les avantages :

- rapidité de création via les RAD
- maquettage ultra rapide pour « montrer » rapidement quelque chose
- convient à la quasi totalité des projets
- le concept d'interface découplée (couplage lâche avec le code fonctionnel)
- outillage aujourd'hui très complet et performant.

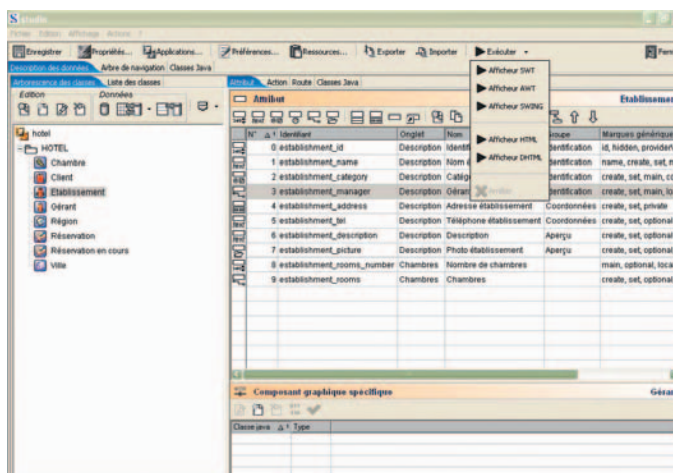
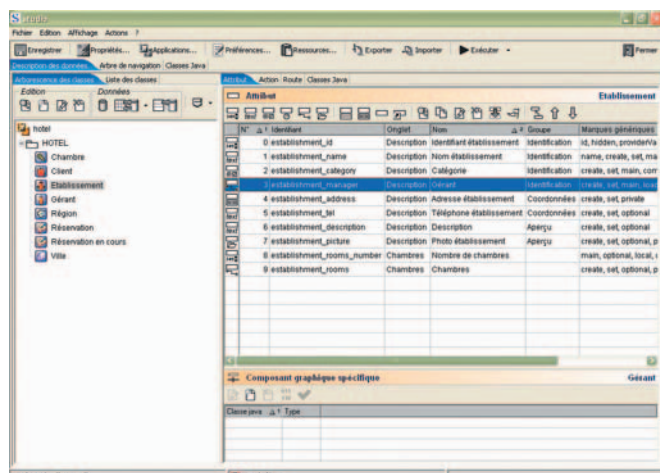
Les inconvénients :

- maintenance lourde et peu pratique dans le cas d'interface très lourde
- multiplication des écrans, donc des fichiers
- internationalisation pas toujours pratique
- indépendance par rapport à l'afficheur (navigateur, terminal...) pas toujours évidente
- l'interface par terminal
- lourdeur des tests

Indépendance des couches

Pour pouvoir gérer sagement une interface, l'idéal est de la rendre indépendante du code fonctionnel et d'exclure tout code des formulaires. Cette indépendance permet de concevoir une interface en totale indépendance. Il suffit ensuite de connecter le code à l'interface. Pour pallier le couplage lâche des couches d'interface au code fonctionnel, au lieu de coder en dur dans l'une ou l'autre couche, il faut rajouter une couche middleware. Cette couche contiendra le code nécessaire pour relier le fonctionnel à l'interface. Plus vous codez en dur dans l'interface même, plus la maintenance sera difficile, sans compter l'évolution, et chaque modification nécessitera une batterie de tests spécifiques, notamment dans la non-régression. Dans les interfaces, la régression doit être une des préoccupations d'une version à une autre.

L'interface en dur continue à exister, cela ne fait guère de doute, au moins à court terme. Difficile de migrer et de refondre tout l'existant. La question peut se poser sur les nouveaux projets. Cependant, le maquettage demeurera essentiel dans de nombreux projets pour établir rapidement un prototype d'application. D'ailleurs, une des tendances actuelles est de pouvoir relier une application, une fonction existante à une architecture service de type SOA. Là, on conçoit une interface totalement indépendante et l'infrastructure service fera la liaison.

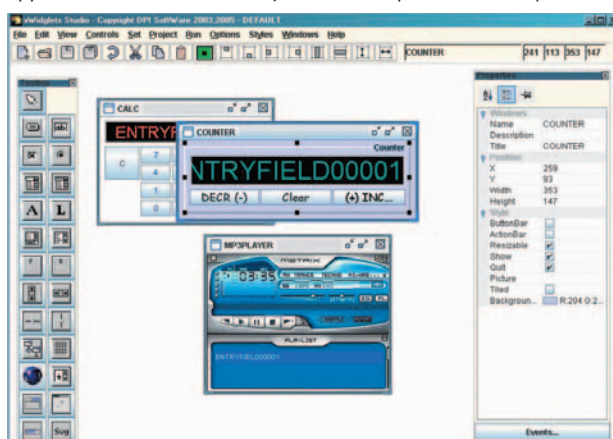


Application de gestion d'hôtel réalisée avec Leonardo.

L'interface dématérialisée : une réalité !

On entend par interface dématérialisée, une interface générée dynamiquement, à la volée selon un contexte utilisateur, selon le type de terminal via un fichier de description, ou générée dans un développement orienté modèle, l'interface se générant à partir du modèle. Les pages web dynamiques sont une forme très rudimentaire d'interface dématérialisée.

Au niveau européen, le projet EMODE a pour but de concevoir et développer une nouvelle approche de l'interface. L'objectif est de pou-



Des objets xWidgets.

voir créer une interface capable de s'adapter dynamiquement au métier des utilisateurs. Le but étant de fournir aux éditeurs et développeurs les outils et spécifications nécessaires.

• L'interface « orientée modèle »

Avec le développement par modèle (UML) et orienté modèle (MDD, MDA, etc.), l'approche que l'on a de l'interface change aussi. Dans le cadre d'un développement basé sur du modèle, l'interface n'existe pas réellement. On définit à partir des use cases, du retour utilisateur, le modèle qui va servir à créer l'interface. On part donc du besoin utilisateur et non d'un besoin technique. On renverse la conception. Quand on sait ce qu'il faut afficher comme données et les traitements et fonctions à appliquer à son modèle pour générer l'application, le modèle d'interface se personnalise et se paramètre en conséquence, mais celle-ci peut, dans le cas d'un environnement de type Leonardi (édité par Lyria) se générer uniquement à l'exécution. « On oublie trop souvent l'utilisateur. Un projet est avant tout un service. Dans un environnement tel que Leonardi, le développeur se focalise sur le modèle, l'application. Réaliser

une interface sans écran est quelque chose de déroutant. L'interface se génère seulement à l'exécution. Avec XML, nous sommes indépendants de l'affichage. Leonardi possède sept afficheurs différents. Pour le développeur et l'utilisateur, c'est totalement transparent. On supportera Ajax dans la prochaine version 3.2 et XUL d'ici à la fin de l'année. » précise Jean-Luc Saouli, P-DG de Lyria.

« Les technologies et méthodologies de type MDA nécessitent du développeur qu'il développe en abstrait », dit Guy Cartigny, Responsable technique développement et intégration,

Compuware France. Avec MDA, on est, encore plus que pour de l'UML ou du MDD, dans l'abstrait. Même si on part toujours de l'utilisateur (les fameux use cases), on passe en MDA, avec plusieurs couches intermédiaires pour définir le modèle, créer ensuite le code selon le langage, la plateforme. « Cependant, si l'interface devient abstraite, on peut toujours personnaliser.

À terme, le développeur va-t-il encore s'intéresser à l'interface ? Quoi qu'il en soit, on redonne de la souplesse. Pour concevoir des interfaces intelligentes, les flux (XML, flux métiers, etc.) se complexifieront. C'est pour cela qu'il est nécessaire de les simplifier, au moins, dans leur manipulation et conception. Le prototypage aura toujours son intérêt dans certains projets. », ajoute Guy Cartigny.

Cependant, partir de l'utilisateur peut poser un problème : fixer l'interface. Or, si les utilisateurs savent peu ou prou ce qu'ils veulent à l'écran ou dans les fonctions, l'expression du besoin n'est pas toujours claire. Et dans ce cas, souvent, la technique pallie au problème en imposant son choix. Or, comme le développement par modèle doit donner toute sa place à l'utilisateur, le risque est de devoir gérer des demandes nouvelles constamment. Car, finalement, quand l'utilisateur sait que l'on peut faire telle chose, il risque de le demander. Bref, si la souplesse de l'interface dans sa conception et sa génération est nécessaire, il faut aussi pouvoir canaliser les besoins utilisateurs pour ne pas avoir une dérive des demandes. Au chef de projet de figer à un moment donné les spécifications...

Les avantages :

- concept abstrait et indépendant
- génération fine et personnalisable
- convient bien aux développeurs, entièrement orienté modèle et aux projets SOA
- rapidité de conception

Les inconvénients :

- maîtriser le concept d'abstraction
- concept encore parfois immature ou flou
- non totalement éprouvé

• Le langage descriptif

Avec l'apparition de langage de type XUL, XAML, UIML ou xWidgets, tous des dialectes XML, l'interface devient totalement virtuelle. Pour le développeur, il s'agit d'une véritable révolution dans la manière de travailler. Rien n'est palpable. L'un des intérêts de ce type d'interface est de pouvoir générer à la volée une interface selon l'utilisateur, le terminal. Tout en créant une indépendance avec le code fonctionnel. Sur UIML, ce projet est actuellement dans l'OASIS. Des projets de conversation de VB vers UIML ou pour .Net existent. Cependant, la multiplication des offres ne fournit pas une bonne lisibilité sur cette technologie. Bien que XML soit la base, les différentes solutions ne sont pas compatibles, ou très difficilement. L'émergence d'Ajax, courant 2005, et des premiers outils de cette technologie, devrait accélérer la mutation de l'interface des sites web, même si Ajax n'est pas adapté à tout et qu'il est encore un peu jeune.

Les avantages :

- maintenance, adaptation ultra rapide des fichiers « d'interface »
- indépendance du code, de l'interface et du système d'affichage
- performance de l'architecture

Les inconvénients :

- des solutions pas encore matures
- manque d'outils (même si cela est de moins en moins vrai)
- multiplicité des langages, le plus souvent incompatibles entre eux
- souvent très verbeux
- change radicalement la conception de l'interface et donc des méthodes de travail du développeur.

■ François Tonic

RubyOnRails : du client riche facile

RubyOnRails (Rails) a été initialement écrit par David Heinemeier Hanson (DHH) pour la société 37Signals, afin de créer une application web : Basecamp. Rails, écrit en Ruby, est basé sur le principe Modèle Vue Contrôleur (MVC) connu depuis longtemps des développeurs du monde objet.

Plusieurs points dirigent Rails et les développements faits avec. DHH martèle depuis le début un principe simple : “Don’t Repeat Yourself” (DRY), “ne vous répétez pas”. Ce principe du développement objet colle parfaitement à Rails et surtout à Ruby.

Nous allons voir une application relativement simple, tout en essayant de couvrir quelques notions de bases : relation 1/N, interface en AJAX, migrations...

Pour cela, nous construirons une petite application de publication de documents avec une gestion d'utilisateurs, création, édition, et suppression d'articles.

Nous rentrerons rapidement dans le vif du sujet en créant, tout d'abord, de quoi écrire des articles et les gérer, puis nous rajouterons de quoi gérer des auteurs et des relations entre documents et auteurs. Les noms de variables seront en anglais, afin de coller au mieux aux principes et méthodes associés à Rails.

Création de l'application

Afin de démarrer notre application, nous devons exécuter la commande rails une première fois, afin d'initialiser l'environnement de développement. On se place dans le répertoire home de notre utilisateur “joe”. Notre application s'appellera “publication” :

```
joe@~/:/> rails publication
```

Voici, rapidement, l'organisation de cet environnement (répertoires principaux) :



■ app : le répertoire contenant

- controllers : les contrôleurs
- helpers : les “helpers” bouts de code destinés à faciliter l'écriture des vues
- models : les modèles
- views : les vues

■ config : le répertoire qui stocke la configuration de l'application :

- database.yml : configuration des bases de données utilisées
- environment.rb : configuration de l'environnement utilisé et configuration annexe de votre environnement
- routes.rb : configuration de routes spéciales (réécriture d'urls, redirections, etc.).
- public : les pages d'erreurs, les feuilles de styles, les javascripts
- script : un jeu de scripts pour gérer votre application

Un modèle, un contrôleur

Depuis la version 0.13, Rails a une fonctionnalité appelée “migrations” permettant de décrire vos modèles (et donc vos tables dans votre base de données) en Ruby, sans se préoccuper du moteur qui sera utilisé par la suite pour votre application.

Éditez le fichier config/database.yml et précisez pour les trois environnements toutes les informations nécessaires (login, mot de passe, nom ou ip du serveur, nom de la base utilisée, driver, commentez la ligne ‘socket’ si besoin).

Pour commencer, nous allons donc décrire le modèle de données de nos documents :

- un identifiant unique (*id*), entier.
- un titre (*title*), chaîne de caractère.
- un texte (*text*), de type text ou similaire.

Créez une migration avec le script *generate* :

```
joe@~/:/publication:> script/generate migration MyMigration
```

Le fichier *001_my_migration.rb* est créé dans le répertoire *db/migrate*. Une migration est composée de deux méthodes : *self.up* et *self.down* pour les deux sens de chaque migration.

```
class MyMigration < ActiveRecord::Migration
  def self.up
    create_table :documents do |t|
      t.column :id, :integer
      t.column :title, :string
      t.column :text, :text
    end
  end
  def self.down
    drop_table :documents
  end
end
```

Exécutez la migration :

```
joe@~/:/publication:> rake migrate
```

Deux tables ont été créées : la table *documents*, et la table *schema_info*. La première n'est pas une surprise c'est celle que l'on a écrit. La deuxième est utilisée par Rails pour suivre les migrations effectuées. Dans votre table *documents*, les trois colonnes ont bien été créées et la colonne “*id*” est même une clef primaire auto-incrémentée.

Création du Modèle

Créez le modèle *Document* avec le script *generate* :

```
joe@~/:/publication:> script/generate model Document
```

Ce script génère le modèle en lui-même (*app/models/document.rb*), et des fichiers annexes (tests, ...).

Note à propos des noms de tables.

Votre table s'appelle “*documents*”, votre modèle

le associé s'appelle "Document" (la table contient plusieurs documents, alors que l'on a une seule classe). Le lien entre le modèle et la table est fait par Rails à partir de leurs noms. En cas de besoin, vous pouvez, dans le modèle, spécifier une table pour le modèle. Pour agir sur le modèle, nous devons créer un contrôleur (avec le script *generate*) :

```
joe@:~/publication:> script/generate
controller documents
```

Le contrôleur est le fichier *app/controllers/documents.rb*.

Note : le nom du contrôleur n'a pas forcément de rapport avec le nom du modèle, puisqu'il n'est en rien "attaché" à un modèle, il peut même en traiter plus d'un.

Le scaffolding permet de générer automatiquement le code nécessaire à l'utilisation basique d'un modèle et d'un contrôleur. Il s'active en rajoutant la ligne dans le contrôleur :

```
scaffold :document
```

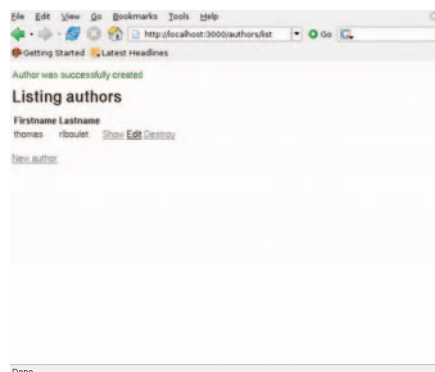
(*document* fait référence au modèle que l'on veut traiter avec le contrôleur).

On lance le serveur via :

```
joe@:~/publication:> script/server
```

Le serveur se lance et est accessible à l'url <http://localhost:3000>. Les urls d'une application Rails sont composées de la façon suivante : <http://hôte/contrôleur/action/>.

Pour arriver, par défaut, sur le contrôleur *documents* nous allons modifier le fichier *config/routes.rb*. Ouvrez le fichier et décommentez la ligne :



```
map.connect "", :controller =>
  "welcome"
```

Remplacez "welcome" par "documents", supprimez le fichier *public/index.html* et rechargez la page précédente. Vous avez de quoi lister, créer, éditer et supprimer des articles.

Relations 1 à N

Pour voir l'intégration des relations 1 à N, nous allons rajouter un modèle "Author" (auteur en anglais) et un contrôleur associé.

Créez le modèle *Author* et le contrôleur *authors* en utilisant *script/generate*. Créez ensuite une migration pour rajouter la table *authors* (*db/migrate/oo2_my_migration.rb*, en copiant la première) la table *authors* doit avoir trois colonnes : *id* (integer), *firstname* (string), et *lastname* (string). Appliquez cette migration et activez le scaffolding (pour le modèle *author*) dans le contrôleur *authors*.

Chargez la page <http://localhost:3000/authors>, vous avez de quoi gérer les auteurs.

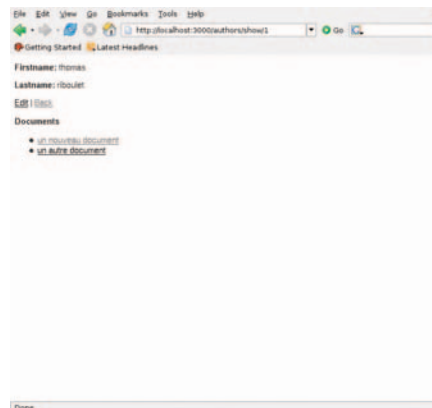
Pour créer la relation 1 à N entre les auteurs et les documents (un auteur écrit plus d'un document) il faut rajouter une colonne dans la table *documents* qui servira à stocker l'id de l'auteur correspondant. Le nom de cette colonne doit suivre une règle de nommage : *nom_du_modèle_id*. Ici ce sera donc "author_id". Créez la migration *oo3_my_migration.rb* et utilisez la méthode "add_column" :

```
add_column :documents, :author_id,
:integer
```

Pour créer les liens entre les modèles, rajoutez dans le modèle *Document* (*app/models/document.rb*) la ligne : *belongs_to :author*, et dans le modèle *Author* la ligne *has_many :documents*. Pour le modèle *Document*, vous obtenez :

```
class Document < ActiveRecord::Base
  belongs_to :author
end
```

Le scaffolding ne gère pas les relations, nous allons devoir écrire du code nous-mêmes. Un script nous permet de partir de ce que le scaffolding utilise, en générant les contrôleurs et les vues, sans toucher aux modèles :



```
joe@:~/publication:> script/generate
scaffold author
joe@:~/publication:> script/generate
scaffold document
```

Nous allons ajouter des fonctionnalités afin de pouvoir créer un document via un formulaire utilisant Ajax, afficher les auteurs dans la liste des documents, et afficher la liste des documents de chaque auteur.

Éditez la vue *app/views/documents/list.rhtml* et rajoutez le code suivant après la ligne :

```
<%= link_to 'New document',:action
=> 'new' %>

<!-- tag de formulaire utilisant Ajax être traité
url : désigne la méthode qui sera
appelée ici, la méthode
new_with_ajax
update : indique l'id du noeud qui
sera mis à jour ->
<%= form_remote_tag(
:url => {:controller => "documents",
:action => "new_with_ajax"},
:html => {:class => 'document'},
:update => 'documents',
:loading => 'Toggle.display(\'new-
document\'); $(\'new-document-
desc\').innerHTML = \'Saving ...\'',
:complete => "($('new-document-
desc').innerHTML = 'Create a new
document');"%>

<div class="info">
<h4>
<span id="new-document-
desc">Create a new document</span>
<a class="expander" href="#"
```

```

onclick="Toggle.display('new-document');
Field.focus('new-document-title');
return false">&raquo;</a>
<!-- un lien pour afficher le formulaire
(ou le masquer) --></h4>
</div>

<div id="new-document" class="content"
style="display: none">
  Title : <%= text_field 'document',
'title', :id => 'new-document-title' %><br />
  Text :<br /><%= text_area
'document', 'text', :id => 'new-document-note'
%><br />
  <label for="new-document-
author">Author :</label>
  <% if Author.find_all.size > 0 %>
  <%= select("document", "author_id",
Author.find_all.collect { |p| [ p.firstname + " " + p.lastname, p.id ] },
{ :include_blank => false }) %>
  <% else %>
  /
  <% end %>
  <br />
  <input type="submit"
value="submit"/>
</div>
<%= end_form_tag %>

```

Rajoutez les balises html, head et body et insérez dans le head les deux lignes suivantes :

```

<%= javascript_include_tag
"prototype" %>
<%= javascript_include_tag
"controls" %>

```

Cela va nous permettre d'utiliser un certain nombre de fonctions javascript, notamment la fonction *Toggle*.

Modifiez la balise table de la liste des documents, pour que celle ci ait l'id "documents" :

```
<table id="documents">
```

Nous allons utiliser un "partial" qui va nous permettre de générer les lignes du tableau depuis la vue ou une méthode du contrôleur. Coupez les lignes entre les balises *table* (de ce même tableau) et copiez les dans le fichier (à créer) *app/views/documents/_documents.rhtml* (le '_' le désigne comme partial) en les modifiant comme suit :

```

<tr>
  <th>Title</th>
  <th>Auteur</th>
</tr>

<tr>
  <% for document in @documents %>
    <tr>
      <td><%= document.title
%></td>
      <td>
        <% if document.author %>
          <%= link_to
document.author.firstname + " " +
document.author.lastname, :controller =>
'authors', :action => 'show', :id =>
document.author.id %>
        <% else %>
          /
        <% end %>
      <td><%= link_to 'Show',
:action => 'show', :id => document %></td>
      <td><%= link_to 'Edit',
:action => 'edit', :id => document %></td>
      <td><%= link_to 'Destroy',
{ :action => 'destroy', :id => document },
:confirm => 'Are you sure?' %></td>
    </tr>
  <% end %>

```

Dans la même vue, remplacez le contenu coupé par la ligne :

```
<%= render_partial "documents" %>
```

Il nous reste à écrire la méthode "new_with_ajax" dans le contrôleur *documents* :

```

def new_with_ajax
  @document =
Document.new(params[:document])
  if @document.save
    @documents =
Document.find(:all)
  # appel du

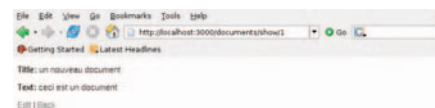
```

```

partial
render_partial
'documents'
else
render :action
=> 'new'
end
end

```

Rechargez la page <http://localhost:3000/documents/list>. Vous pouvez ajouter des articles sans quitter cette page (il faut des auteurs ins-



crits pour que la liste d'auteurs apparaisse). Éditez maintenant la vue *app/views/authors/show.rhtml* et ajoutez le code suivant :

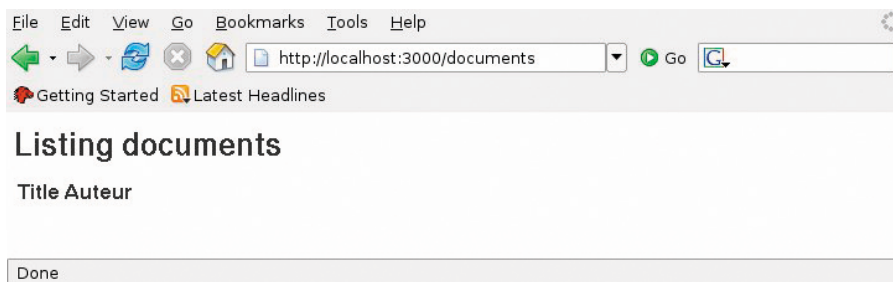
```

<h3>Documents</h3>
<ul>
  <% for document in @author.
documents %>
    <li><%= link_to
document.title, :controller => 'documents',
:action => "show", :id => document.id
%></li>
  <% end %>
</ul>

```

Vous avez désormais une application complète permettant de créer des documents, des auteurs, des les éditer, de les supprimer.

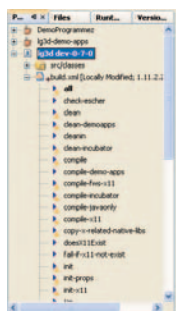
■ **Thomas Riboulet**
Consultant Open Source Uperto



Champ	Valeur
Working Directory	Le répertoire dans lequel vous souhaitez déposer les sources sur votre machine
CVS Server Type	pserver
Port	2401
User Name	Votre nom de login java.net
Repository Path	/cvs
Password	Votre mot de passe de login java.net. Inutile <i>a priori</i>
Module(s)	lg3d-core
Revision or Tag	dev-0-7-0

Il ne m'a pas été nécessaire de cliquer sur le bouton 'Login' de la boîte de dialogue. Le mot de passe dans ce cas est évidemment inutile. Ne pas cliquer sur le bouton 'Select...' en face de module(s), ça ne fonctionnerait pas. Donnez directement le nom du module, soit lg3d-core. Rien ne vous interdit de tester l'existence d'une nouvelle version, par exemple, en donnant dev-0-7-2.

3 Compiler le projet



Depuis le menu de NetBeans, allez dans File/Open Project. Depuis la boîte de dialogue, rendez-vous dans le répertoire que vous avez indiqué lors de la configuration CVS. Vous devez y trouver une icône de répertoire de projet, de nom lg3d-core. Sélectionnez ce répertoire et validez. Il est intéressant de noter que le projet dispose d'un fichier de construction ANT personnalisé, ce qui a permis de spécifier moutlt cibles. Nous y reviendrons un peu plus loin. Pour l'instant, comme pour tout projet traditionnel, compilez et lancez Looking Glass en pressant la touche F6. Pour stopper Looking Glass, cliquez sur la petite icône à tête de mort, en bas et à droite du bureau.

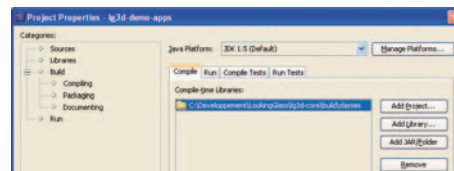
4 Télécharger et compiler les démos

Lors de ce premier essai, vous avez pu constater que le bureau est désespérément vide. Nous allons donc télécharger les applications de démonstration, à nouveau via CVS sous Netbeans. Les champs à remplir dans la boîte de dialogue CVS sont les mêmes que précédemment. Simplement, on donnera comme nom de module, lg3d-demo-apps, et on veillera à donner le même répertoire de travail (Working Directory) que pour Looking Glass lui-même. Bien entendu, on choisira une version en adéquation avec la version de Looking Glass. Après téléchargement, allez dans le menu File/New Project, puis choisissez 'Project with Existing Files'. Donnez, par exemple, comme nom de projet 'lg3d-demo-apps' et décochez la case 'Set as main project', afin que Looking Glass reste le projet principal. Dans la boîte de dialogue suivante faites pointer sur le répertoire sur :

vos_repertoire/lg3d-demo-apps/src/classes

Enfin, vous devez faire pointer le classpath de votre projet sur les classes de Looking Glass. Pour cela, allez dans les propriétés du projet et cliquez sur 'Add JAR/Folder' pour ajouter le répertoire lg3d-core/build/classes

(voir ci-dessous). Le projet est prêt. Il ne vous reste plus qu'à le construire-



re, en choisissant 'Build' dans son menu contextuel (voir ci-dessous à gauche). Après la construction, pressez la touche F6, ce qui revient à lancer Looking Glass lui-même, puisque c'est lui le projet principal. De nouvelles applications se sont intégrées au bureau, une aide à l'utilisation notamment (voir ci-contre).

5 Intégrer une nouvelle application à Looking Glass

En compilant les applications de démonstration, nous avons indirectement fait connaissance avec la démarche pour intégrer une nouvelle application au bureau 3D. On crée un projet avec un classpath pointant sur Looking Glass. La documentation de ce dernier nous apprend que toute application dont le Jar est déposé dans lg3d-core/ext/app est automatiquement intégrée au bureau. La pratique est un peu différente. D'abord, vous devrez peut-être créer le répertoire app. Ensuite, les Jars qui y sont déposés ne sont pas intégrés au CLASSPATH, au moins, en mode développement. Autrement dit, l'existence de l'application est reconnue, mais celle-ci ne peut fonctionner. Un moyen simple d'y remédier est de modifier la cible run-dev du fichier de construction (build.xml) personnalisé de Looking Glass que nous avons mentionné plus haut, en ajoutant un élément classpath à la tâche java de la cible run-dev

```
<classpath>
<pathelement location="${ext.dir}/app/demoprogrammez.jar"/>
</classpath>
```

Si vous avez des difficultés à comprendre le contenu du fichier build.xml, vous pouvez vous reporter à vos *Programmez !* 77 et 78. Revenons à notre application, et éditez son fichier manifest.mf pour y ajouter une entrée dans cet esprit :

Implementation-Title: MonApplication

Sachant que l'entrée Main-Class, également requise, sera automatiquement ajoutée par NetBeans, il ne nous reste plus qu'à faire que le Jar de l'application soit systématiquement déposé dans lg3d-core/ext/app après construction. Voici la recette. Éditez le fichier build.xml standard de votre application et ajoutez cette cible :

```
<target name="post-jar">
<copy file="${dist.jar}" todir=" ../lg3d-core/ext/app"/>
</target>
```

À la découverte de Looking Glass

SOURCES DE L'ARTICLE
WWW.PROGRAMMEZ.COM

Looking Glass est un bureau en 3D codé en Java et créé à l'initiative de Sun. Remplacera-t-il un jour Gnome, KDE ou le bureau de Windows ? L'avenir nous le dira. En attendant, c'est une passionnante occasion de faire de la 3D.

Rien de mieux que de spectaculaires projets ou de riches bibliothèques, pour promouvoir un langage. Sun semble l'avoir compris mieux que personne et n'en finit pas de lancer des projets dans le domaine du multimédia. Looking Glass est un projet ambitieux de bureau en 3 Dimensions, construit au-dessus de l'API Java 3D et de JAI. Le déve-



loppement de Looking Glass a d'abord été entrepris par Sun qui l'a finalement donné à la communauté Open Source. D'abord écrit pour Solaris et Linux, Looking Glass tourne aujourd'hui également sous Windows. Sous les systèmes de type Unix, Looking Glass peut réellement être un bureau de remplacement, puisqu'il suffit d'avoir un serveur X actif pour le faire tourner. Sous Windows, une fenêtre 'plein écran' recouvre le bureau natif. Attention, ni la 3D, ni Java ne sont particulièrement économes en ressources systèmes. Pour faire fonctionner la bête, vous devrez posséder une grosse configuration. Sous Linux, une carte graphique, avec un pilote OpenGL récent et prenant en charge l'accélération matérielle, est requise. 512 Mo de Ram sont un minimum pour le développement, de même qu'une CPU à 1.4 GHz sous Windows et 2 Ghz sous Linux, nous dit-on sur <https://lg3d.dev.java.net/> la page du projet. Votre serveur a travaillé avec une CPU à 2 GHz sous Windows et n'a pas constaté que ça allait trop vite au lancement du bureau.

Looking Glass en est encore dans les tous premiers stades de développement. Mais il n'en est pas moins, d'ores et déjà, excitant de développer de petites applications dans le cadre de ce bureau si on aime faire de la 3D, ou pourquoi pas, participer activement au projet. Nous allons maintenant voir comment compiler et faire tourner Looking Glass, puis comment écrire une application de type Hello World, tradition oblige.

1 Les outils logiciels

Vous devrez utiliser une JDK 5.0, ce qui est une formalité. Une variable d'environnement JAVA_HOME pointant sur le répertoire d'installation de la JDK doit être définie. Votre PATH doit pointer sur le sous-répertoire bin de la JDK. Vous devez ensuite télécharger (voir encadré ressources en fin d'article) la SDK de la Java Advanced Imaging API, version 1.1.2 minimum. Celle-ci vient avec un installateur automatique. Au tour maintenant de l'API Java 3D, version 1.4.0 minimum. Pas d'installateur automatique cette fois. Décompactez l'archive dans le répertoire jre de la JDK (et non dans une JRE autonome). Looking Glass est écrit avec NetBeans. Même s'il est possible d'utiliser n'importe quel outil de développement Java, NetBeans va remarquablement bien et votre serveur vous conseille de télécharger NetBeans 4.1 sans attendre. Enfin, vous vous procurerez la Javadoc de Looking Glass. Étrangement, celle-ci est séparée du projet. Avant de se lancer dans l'aventure, il faut être conscient que le projet évolue vite et parfois avec des modifications importantes, comme une refonte d'une partie des APIs. Il se peut que la Javadoc ne soit pas tout à fait à jour, ou même que des éléments du projet soient à retoucher par vos soins. Par exemple, j'ai dû corriger des scripts de lancement de Looking Glass en mode autonome. Ces scripts résident dans le sous-répertoire du projet. Ces désagréments, somme toute mineurs, sont inévitables avec un projet à ce stade de développement. Tout ce qui est dit dans cet article se base sur des expérimentations faites avec la version 0.7.0. De nouvelles moutures seront sans doute disponibles à l'heure où vous lirez ces lignes.

2 Se procurer les sources de Looking Glass



La première chose à faire est de s'enregistrer à <http://www.java.net>, ce qui revient simplement à créer un login. Les sources de Looking Glass sont en CVS. Le plus simple est de les télécharger depuis Netbeans. Pour cela, allez dans le menu Versioning | CVS | Check out... de NetBeans. Passez la première boîte de dialogue en cliquant sur le bouton Checkout. Vous arrivez alors à une boîte de dialogue semblable à celle ci-contre. Les champs de la boîte de dialogue sont à remplir comme suit :

6 Anatomie d'une application Looking Glass

Voici le code du premier exemple du tutorial de Looking Glass:

```
public class Tutorial1 {
    public static void main(String[] args) {
        new Tutorial1();
    }

    public Tutorial1() {
        Frame3D frame3d = new Frame3D();
        SimpleAppearance app = new SimpleAppearance(0.6f, 0.8f, 0.6f);
        Box box = new Box(0.04f, 0.03f, 0.02f, app);
        Component3D comp = new Component3D();
        comp.addChild(box);
        frame3d.addChild(comp);
        frame3d.setPreferredSize(new Vector3f(0.08f, 0.06f, 0.04f));
        frame3d.setEnabled(true);
        frame3d.setVisible(true);
    }
}
```

Les applications Looking Glass, API java 3D oblige, (cf. *Programmez ! 60* et *61*) sont des arborescences. Dans l'exemple, l'extrémité (ou feuille) de la branche est Box, un parallélépipède rectangle géré par Component3D, nœud sur la branche. Vient au-dessus de cela, Frame3D, autre nœud, qui fait la liaison entre notre branche et L'univers, c'est-à-dire Looking Glass lui-même. SimpleAppearance a pour rôle de définir le rendu des feuilles. Tout ce qui apparaît sur l'écran est doté d'un objet dérivant de Appearance pour définir le rendu.

7 Aller plus loin

Tout cela est très bien, mais une application de bureau doit avoir un affichage réactif. Imaginons, par exemple, une console qui affiche des lignes de sortie d'une commande. Comment obtenir cela ? En l'état, le tutorial n'en dit rien. Pour développer une application, la première chose à faire est l'inventaire des classes à notre disposition dans l'API. Les triangles, cubes, sphères, et autres panneaux préfabriqués nous déchargeront d'un grand travail, mais ne résolvent pas le problème d'un affichage changeant. Pour cela, une seule solution : créer une texture à la volée et l'appliquer sur un panneau, par exemple. Le code ci-dessous illustre le procédé.

Un Hello World en 3D

```
package programmez.lg3d;

import java.awt.*;
import java.awt.image.*;
import javax.vecmath.Vector3f;
import org.jdesktop.lg3d.wg.*;
import org.jdesktop.lg3d.utils.shape.*;
import org.jdesktop.lg3d.sg.*;
```

```
import org.jdesktop.lg3d.sg.utils.image.*;
import org.jdesktop.lg3d.utils.eventaction.*;
import org.jdesktop.lg3d.utils.c3danimation.*;
```

```
public class DemoProgrammez extends Frame3D {

    private final float depth = 0.005f;
    private final float border = 0.010f;
    private float width, height;
    private Appearance appearance;

    public DemoProgrammez() {
        height = getScreenHeight() * 0.5f;
        width = height;

        setPreferredSize(new Vector3f(width, height, depth));

        appearance = new SimpleAppearance(
            0.6f, 1.0f, 0.6f, 1.0f, SimpleAppearance.DISABLE_CULLING);

        GlassyPanel facade
            = new GlassyPanel(
                width + border, height + border,
                depth, appearance);

        GlassyPanel sousfacade
            = new GlassyPanel(width, height, border, appearance);

        SimpleAppearance panelapp
            = new SimpleAppearance(1.0f, 1.0f, 1.0f,
                SimpleAppearance.ENABLE_TEXTURE
                | SimpleAppearance.DISABLE_CULLING);
        try {
            panelapp.setCapability(Appearance.ALLOW_TEXTURE_WRITE);
            panelapp.setTexture(doLogo());
        } catch (Exception e) {
            throw new RuntimeException(e);
        }

        FuzzyEdgePanel panel
            = new FuzzyEdgePanel(width, height, border, panelapp);

        Component3D comp = new Component3D();
        comp.addChild(facade);
        comp.addChild(sousfacade);
        comp.addChild(panel);
        addChild(comp);
    }

    Texture doLogo() {
        BufferedImage bimage;
        Graphics2D g;
        int fontSize = 48;
        int iwidth = 300;
```

```
int iheight = 300;
String text = "Programmez!";

bimage = new BufferedImage(iwidth, iheight,
    BufferedImage.TYPE_4BYTE_ABGR);
Font font = new Font("Serif",
    Font.BOLD | Font.ITALIC, fontSize);

g = (Graphics2D)bimage.getGraphics();
g.setRenderingHint(RenderingHints.KEY_TEXT_ANTIALIASING,
    RenderingHints.VALUE_TEXT_ANTIALIAS_ON);
FontMetrics fontMetrics = g.getFontMetrics();
g.setColor(Color.WHITE);
g.drawRect(0, 0, iwidth, iheight);
g.setColor(Color.BLUE);
g.setFont(font);
g.drawString(text, 10, iheight/2);
g.dispose();
TextureLoader loader = new TextureLoader(bimage);
return loader.getTexture();
}

public static void main(String[] args) {
    Frame3D f3d = new DemoProgrammez();
}
```

```
f3d.setEnabled(true);
f3d.setVisible(true);
}
```

Celui-ci est simple. Nous créons une `BufferedImage` et nous écrivons dessus à la suite de quoi nous la transformons en texture, grâce à un objet `TextureLoader`. Il reste un point fastidieux à gérer. Vous pouvez remarquer dans le code le placement empirique de notre logo. Pour faire un travail fin, il convient de tenir compte du fait que dans un objet `BufferedImage`, l'unité de coordonnées est le pixel, ce qui est classique en 2D. Sous Looking Glass, l'unité est le mètre, ce qui est classique en 3D. Il conviendrait donc de mettre en place un système de conversion. Et bien entendu les cartes graphiques travaillent avec un nombre de pixels au pouce (soit 25.4 mm), nombre différent sous X et sous Windows qui plus est. Le système de conversion est donc :

pixels → pouces → mètres

Le tout est de connaître au départ le nombre de pixels au pouce sur la plate-forme. Ce nombre s'obtient ainsi :

```
java.awt.Toolkit.getScreenResolution();
```

■ Frédéric Mazué fmazue@programmez.com

KAPITEC SOFTWARE, Une Offre d'Outils Performants et Compétitifs Pour Réussir Vos Projets Web

Développez des Applications Web...



L'outil de Développement Web le plus productif pour créer visuellement des applications dynamiques et du reporting Web avec un minimum de codage

Testez la Tenue de Charge...



Le premier logiciel de Test de Charge et de Stress Web à offrir performance, fiabilité et facilité de mise en oeuvre à un prix abordable

Analysez votre Trafic Web...



La seule solution d'Analytique Web professionnelle adaptée aux besoins et aux budgets de tout type d'utilisateurs et d'entreprises

Gagnez le maximum d'efficacité et de compétitivité, et offrez le meilleur de votre savoir-faire à vos clients tout en maîtrisant vos coûts informatiques

Téléchargez dès maintenant votre version d'évaluation sur www.kapitec.com



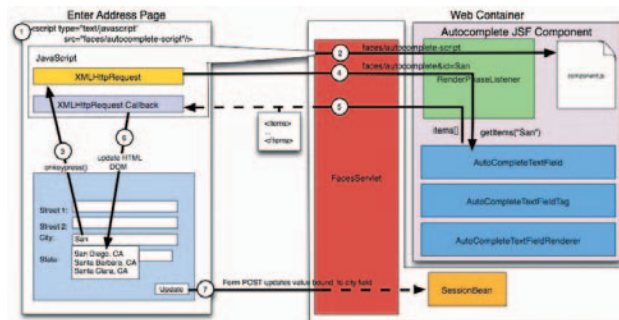
The Most Comprehensive Selection of Software Tools

14, avenue Jean Bouin
31620 Fronton - FRANCE
Tél.: 05 34 27 90 03 - Fax : 05 34 27 90 04
sales@kapitec.com - www.kapitec.com

Les approches de développement avec AJAX

Pratique

Ajax, un terme à la mode ! Qu'est-ce que c'est ? Est-ce une révolution ? Comment peut-on aborder le sujet ? Quels sont les impacts sur mon projet ? Quelle implémentation choisir ? Pour répondre à ces interrogations, je vais vous exposer ma réflexion avec une approche multi-techno, à savoir PHP – Java et .Net.



Les créateurs de site web ont pris conscience de l'importance de replacer les utilisateurs au cœur d'interfaces plus riches et plus interactives. C'est ainsi que Jesse James Garrett a lancé, début 2005, le concept AJAX, basé sur un système d'échange de données asynchrone.

Son objectif est de modifier des pages web dynamiquement en injectant, au moyen de javascript, du contenu récupéré sur le serveur.

Concrètement, AJAX repose sur l'exploitation :

- De rafraîchissement des données en temps réel
- De la gestion de la cinématique
- Du glisser – déplacer
- De la validation des données saisies en temps réel
- Des actions de type Blur – key press – hover
- ...

Néanmoins, selon les bibliothèques AJAX, certaines actions ne sont pas réalisables, comme l'indexation du contenu dans les moteurs de recherche, la gestion des périphériques contrairement aux applications « client riche » et « client-serveur » traditionnelles, ou encore la gestion du bouton « retour » de votre navigateur et les bookmarks.



Voici quelques sites de référence qui traduisent clairement ce que l'on peut faire avec AJAX :

- Gmail <http://mail.google.com/mail/help/intl/fr/about.html> Ce client mail propose une interface extrêmement réactive à la manipulation et préférée par de nombreux utilisateurs à leur client mail traditionnel.
- Google Suggest, le mécanisme d'autocomplétion. <http://www.google.com/webhp?complete=1&hl=en>
- Le portail netVibes <http://www.netvibes.com/>
- Rialto <http://rialto.application-servers.com/wiki/>

Techniquement sur quoi repose ce concept ?

Même si Ajax est un acronyme de terme technique, il n'en est pas pour autant une révolution technologique, étant donné que ce concept repose sur la combinaison intelligente de technologie pas si récente que ça.

- L'utilisation du langage de formatage standard (XHTML, CSS).
- L'échange des données au format XML
- L'invocation des données en mode asynchrone (objet XMLHttpRequest ou activeX xmlhttp)
- L'utilisation du javascript pour manipuler astucieusement le DOM.

Cette description est celle du modèle pur d'AJAX, mais il est fréquent de trouver des variations substituant la manipulation des données XML par l'utilisation du formalisme JSON (JavaScript Object Notation) par exemple, qui permet de transposer les types de données complexes entre les langages serveurs et le javascript.

Comment procéder à l'intégration d'AJAX dans vos projets ?

D'abord, il est préférable de maintenir une organisation classique du code (designs patterns MVC ou MVC2) afin de stabiliser votre logique métier. Ensuite, vous ajouterez l'interactivité souhaitée avec AJAX. Plutôt que votre application retourne une page HTML, elle collectera du XML, une chaîne de caractères à évaluer (ex : JSON) ou une partie de flux HTML que l'on intégrera côté javascript, avec la propriété innerHTML par exemple.

Pour procéder à cette intégration, nous pouvons distinguer deux approches, l'une pure javascript qui garantit l'interopérabilité avec tous les serveurs, l'autre basée sur une architecture serveur générant le javascript de communication à la volée.

L'approche javascript

Elle consiste à charger des fichiers javascript dans votre page HTML. Puis, en se reposant sur les fonctions mises à votre disposition par les bibliothèques préalablement chargées, vous devez écrire vos fonctions javascript spécifiques au comportement que vous souhaitez réaliser.

Deux solutions majeures s'offrent à vous.

La première architecture javascript repose sur le très riche projet DojoToolkit :

<http://dojotoolkit.org/>

Dojo toolkit, encore en cours de développement, se présente comme une API de développement, regroupant un grand nombre de fichiers javascript :

<http://www.rosanbo.com/paul/index.php/2005/04/03/41-ajax-avec-le>



La seconde repose sur la capitalisation du projet Ruby on Rail, avec notamment le fichier prototype.js (qui gère les problématiques d'appels asynchrones) et les bibliothèques scriptaculous ou openRico (qui gèrent les Drag and Drop, les effets de cinématique et de comportement). Ci-dessous, voici un exemple de complétion réalisé avec scriptaculous :

Mise en pratique : Contrôler PHP

```
$oData = new personneSql();
$aPers = $oData->getMail($_POST["word"]);
$flux="<ul>";

foreach ($aPers as $key => $val){
    $flux.="<li>". $val["NOM"]. "=>". $val["EMAIL"] . "</li>";
}

$flux.="</ul>";
return $flux;
```

Votre script doit retourner une liste html UL et LI.

Mise en pratique : Couche html

```
<html>
<head>
    <script src="lib/prototype.js"
    type="text/javascript"></script>
    <script src="src/scriptaculous.js"
    type="text/javascript"></script>
</head>
<body>
<form>
    <input type="text" id="word"/>
    <div id="autocomplete_choices"><div/>
</form>

<script language="javascript">
    new Ajax.Autocompleter("word", "autocom
    plete_choices", "indexScriptaculos.php");
</script>
```

```
</body>
</html>
```

La fonction Ajax.Autocompleter précise le champ de collecte du mot clé, puis le champ à mettre à jour et enfin le script à appeler, côté serveur.

L'approche serveur

Elle consiste à manipuler une librairie écrite en langage serveur permettant de générer le flux javascript de communication à votre place.

Concrètement, cela revient généralement à instancier un objet, puis déclarer les fonctions que vous souhaitez exposer au fonctionnement asynchrone, et enfin lancer la directive de génération du javascript que vous intégrez dans votre couche de présentation via un taglib ou un moteur de template par exemple.

Les implémentations disponibles vont plus loin que la simple abstraction du mécanisme d'invocation asynchrone. En effet, les bibliothèques se présentent plus souvent comme de petits frameworks applicatifs avec une richesse technico-fonctionnelle plus ou moins variée.

Pour faciliter votre choix, voici une liste de critères :

- Support des langages côté serveur
- Compatibilité avec les navigateurs
- Richesse applicative (autocomplétion – widget ...)
- Support des proxy http
- Qualité du code
- Qualité de la documentation
- État du développement (bêta ou stable)
- Possibilité de surcharger le fonctionnement de l'API
- Simplicité d'utilisation
- Conformité au modèle Ajax pur
- Souplesse d'intégration
- Propreté d'intégration

Les implémentations PHP

CPAINT est l'une des bibliothèques orientée objet les plus avancées dans le monde PHP, respectant le modèle AJAX pur et idéal pour manipuler des formulaires métiers.

<http://cpaint.booleansystems.com/>

Néanmoins, les possibilités de cette librairie ne proposent pas de réaliser simplement de l'autocomplétion, je vais donc réaliser ici un système de recherche.

Côté Serveur :

```
include_once ('dictionnaireSql.php') ;
require_once("cpaint/cpaint2.inc.php");

$cp = new cpaint();
$cp->register('getDef');
$cp->start();
$cp->return_data();

function getDef($word) {
    global $cp;
    $oDicoData = new dictionnaireSql($word) ;
    $aDef = $oDicoData->getDefinition();
    foreach ($aDef as $key => $val){
        $cp->set_attribute('id', $val['LIB']);
        $cp->set_data($val['DEFINITION']);
    }
    return;
}
```

Côté client :

```
<script src='cpaint2.inc.js' type='text/
javascript'> </script>
<script language="javascript">
    var cp = new cpaint();
    cp.set_transfer_mode('get');
    cp.set_response_type('xml');
    cp.set_debug(true);
```



```
function searchDef() {
    cp.call('indexcpaint.php', 'getDef',
    def_results,
    document.getElementById('word').value);
}

function def_results(result) {
    document.getElementById('result').value =
    result.getElementsByTagName('ajaxResponse')
    .item(0).firstChild.data
}
</script>
```

D'autres solutions sont également intéressantes :

- XAJAX fournit une liste de comportements prêts à l'emploi. <http://www.xajaxproject.org/>
- XOAD, également nommé NAJAX, repose sur l'utilisation de JSON et a reçu le prix de l'innovation en août 2005 par phpclasses.org http://wiki.xoad.org/index.php?title=Wiki_Home
- HTML_AJAX du projet PEAR http://pear.php.net/package/HTML_AJAX/

Les implémentations Multi-plates-formes

SAJAX (ASP / ColdFusion / Perl / PHP / Ruby ...) : Pour fonctionner il suffit de respecter une seule contrainte : le nom de la fonction javascript destiné à l'appel asynchrone (x_calculateBudget) doit être le même que le nom de la fonction AJAX implémentée côté serveur (calculateBudget).

<http://www.modernmethod.com/sajax/>

Les implémentations JAVA

DWH est un framework fonctionnant sur le même principe que SAJAX et il peut être combiné avec n'importe quel framework du type Struts ou Tapestry :

<http://getahead.ltd.uk/dwr/overview/dwr>.

<http://getahead.ltd.uk/dwr/examples>

Ajax Tags est une librairie fournissant un nou-

veau tag JSP qui simplifie l'utilisation du mode asynchrone. (<http://ajaxtags.sourceforge.net> et <http://ajaxtags.no-ip.info>)

Cet exemple d'autocomplétion permet de retrouver une liste de données collectées à partir d'une servlet

```
<form>
    Make: <input id="model" name="model"
    type="text" size="30" class="form-autocom
    plete" />
    Model: <input id="make" name="make"
    type="text" size="30" />
</form>

<ajax:autocomplete

baseUrl="${pageContext.request.contextPath}
/GetModel.view"
source="make"
target="model"
parameters="model={model}"
className="form-autocomplete"
progressStyle="throbbling"
minimumCharacters="1" />
```

Les implémentations .Net

Pour .Net, vous trouverez sur le web de nombreux exemples d'utilisation d'Ajax en .NET, s'appuyant sur le framework 1.1 et 2.0 bêta. Dans le premier cas, l'utilisation est verbeuse et demande pas mal de code client. Dans le second, les exemples s'appuient sur l'interface System.Web.UI.ICallBackEventHandler.

En gros, le principe est assez simple. Côté client, on doit trouver deux fonctions : une qui exécute la requête au serveur, et une qui récupère le résultat du serveur. L'utilisateur doit coder la méthode javascript qui traite le retour du serveur. En clair, on a deux méthodes côté serveur : RaiseEventCallBack : cette méthode traite l'envoi du client.

GetCallbackResult : cette méthode renvoie le résultat au client.

Conclusion

En résumé, j'aurais tendance à préconiser une architecture javascript, afin de limiter toute adhérence avec les langages serveur et de bénéficier de la richesse fonctionnelle des solutions proposées comme Scriptaculos ou OpenRico (inégalées selon moi par les implémentations serveur). Néanmoins, le langage javascript n'est pas toujours bien maîtrisé par les personnes en charge du développement métier, dans ce cas, il est donc plus judicieux d'utiliser une librairie côté serveur dès lors que les besoins restent simples.

D'une manière générale, que vous choisissiez une architecture javascript ou serveur, il sera de toute évidence nécessaire d'implémenter en javascript le comportement spécifique demandé par votre application en se reposant ou non sur les fonctions javascript vous facilitant la manipulation des données.

Pour faciliter vos tâches d'implémentation AJAX, je ne peux que vous conseiller de vous munir de Firefox et de plusieurs extensions ou plug-ins comme :

Live http Headers qui permet de visualiser les trames http en temps réel

View Rendered Source Chart pour visualiser votre flux HTML de manière structurée

Venkman, le débbugger javascript

Enfin, si on souhaite identifier la charge de travail qui distingue un développement MVC classique par rapport à un développement MVC incluant AJAX, cela revient à qualifier la charge de travail relative à l'écriture d'un contrôleur (qui retourne uniquement des données structurées) et à l'écriture de code javascript permettant d'intégrer le comportement demandé. Finalement, on peut constater qu'Ajax n'est pas révolutionnaire mais qu'il apporte une interactivité, souvent très demandée par les utilisateurs.

■ Johann Lemaitre, SQLI

Abonnez-vous en ligne
www.programmez.com

Soyez acteur de votre carrière

Rejoignez SQLI vous avez un rôle à jouer !

Vous souhaitez...

Partager notre passion des défis technologiques
pour être au cœur des stratégies d'innovation des grands projets
e-business de demain

Enrichir votre expérience et élargir vos compétences
en travaillant sur des projets complexes de grande envergure


**Evoluer dans un environnement qui vous permet
d'exprimer pleinement vos talents**, et donne de la valeur
à vos idées

**Bénéficier d'un cadre favorisant la responsabilisation
individuelle**, les opportunités de progression et d'évolution
multiples, la mobilité géographique...

... SQLI vous parle d'avenir.

1ère Société de Service certifiée CMMI en 2004, modèle de qualité des développements logiciels aujourd'hui incontournable, le Groupe SQLI est le leader français des "pure players" e-business, avec plus de 1100 collaborateurs répartis sur 11 agences en France, 2 en Suisse et 2 au Maroc.

Architecte en Systèmes d'Information nouvelle génération, SQLI propose à ses clients un accompagnement global, du Conseil en amont, à la TMA en aval en mettant sa forte densité technologique au service de solutions métier et projets de pointe.

 **Groupe SQLI** - Paris, Nantes, Poitiers, Bordeaux, Toulouse, Montpellier, Aix-en-Provence, Lyon, Strasbourg, Belfort, Dijon, Lausanne, Genève, Rabat, Casablanca

Siège social : 268 avenue du Président Wilson - 93210 La Plaine Saint-Denis
Contacts : Alexandra Lecordier - Delphine Cousin - Joëlle Lessard
jobsqli@sqli.com - 01 55 93 26 00 - www.sqli.com



Nous recherchons des...

Chefs de Projet

Ingénieurs

Développement

Directeurs de Projet

Architectes en SI

Experts techniques

Contactez-nous

jobsqli@sqli.com - www.sqli.com

**SQLI, 1ère SSII à placer le
management de ses
collaborateurs au cœur de
sa stratégie en adoptant le
modèle People-CMM, dédié
à la qualité de la gestion
des Ressources Humaines**



les architectes des solutions e-business nouvelle génération

Persistance des données avec Noas PHP

SOURCES DE L'ARTICLE
WWW.PROGRAMMEZ.COM

Le langage PHP et son nouveau moteur objet ne cesse de monter dans l'estime des développeurs. Depuis sa sortie officielle, en juillet 2004, il existe de plus en plus d'alternatives, du point de vue logiciel et librairies spécifiques, pour accroître la productivité et la qualité des projets. Focus sur le dernier-né, Noas PHP, un framework MVC complet intégrant les fonctionnalités nécessaires aux applications les plus complexes.

Au même titre qu'un développeur J2EE ou .Net, le développeur PHP orienté objet dispose d'une gamme de logiciels satisfaisante. Les éditeurs WYSWYG ont laissé leur place aux IDE qui n'ont pas grand-chose à envier chez leurs homologues d'autres langages. On trouve des plug-ins pour générer des classes PHP depuis UML ou encore, des éditeurs équipés d'un débogueur. Tout comme Java, PHP est libre et bénéficie d'une communauté de développeurs et de documentalistes, très réactive. La seule ombre à ce tableau idéaliste, et pas des moindres, est la disparité des moyens de réalisation. Il n'y a pas de « façon » homogène pour développer une application avec PHP. Cela n'est pas gênant dans l'absolu, mais les équipes de développement sont systématiquement amenées à créer de zéro leurs propres solutions qui, pour des raisons de coût et de délais, sont généralement d'une réutilisation très limitée. En passant par un framework, on optimise les développements sur tous les points. Il est difficile de proposer une solution qui couvre tous les besoins, mais c'est dans cette volonté que le framework Noas PHP a été conçu et réalisé : Produire, pour PHP, une couche technique robuste et extensible, en majeure partie dédiée à l'intra net, extra net, au front office, back office et aux portails d'entreprise.

Installation

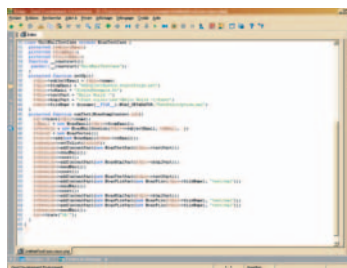
Noas PHP n'est pas une extension PHP, ni un module Apache, encore moins un de ces composants « exécutables ». Il s'agit purement et simplement d'un ensemble de classes et d'un fichier de configuration. Procurez-vous l'archive en prenant soin de choisir celle qui correspond à l'extension des scripts PHP5 reconnue par votre serveur (.php ou .php5, ou utilisez un logiciel pour renommer tous les fichiers à votre convenance). Décompressez le contenu de l'archive dans un répertoire neutre, que nous appellerons « NOAS_INSTALL ». Il ne vous reste plus qu'à définir correctement les constantes du fichier de configuration « noas-server-conf.inc ».

Définissez la constante « NOAS_SEPARATOR » à « \ » si vous développez sous Windows ou à « / » pour Linux.

Définissez « NOAS_HOME » à « NOAS_INSTALL ».

Définissez « NOAS_WORK_DIR » en spécifiant le répertoire dans lequel vous souhaitez que le framework génère ses fichiers temporaires. Veillez à ce que PHP ait le droit d'y créer des dossiers et des fichiers.

Définissez « NOAS_LOG_DIR » en spécifiant le répertoire dans lequel vous souhaitez que le framework génère les fichiers de logs. Veillez à



ce que PHP ait le droit d'y créer des dossiers et des fichiers. Définissez « NOAS_PHP_EXTENSION » en spécifiant l'extension PHP5 reconnue par votre serveur (.php, .php5 ou autres). La configuration étant terminée, nous pouvons débiter la découverte d'une partie du framework,

au travers d'un petit test unitaire de création / enregistrement / chargement / suppression d'un objet métier.

Base de données

Pour stocker vos informations, vous avez le choix entre les diverses bases supportées par le framework : Oracle, MySQL, SQL Server, Postgres, SQLite, etc. La table que nous allons utiliser correspond aux messages envoyés par des internautes depuis un formulaire. Voici sa structure au format SQLite :

```
CREATE TABLE t_user_message (
  sid BIGINT NOT NULL PRIMARY KEY ,
  email VARCHAR ( 60 ) NOT NULL ,
  subject VARCHAR ( 200 ) NOT NULL ,
  message TEXT NOT NULL ,
  date_register TIMESTAMP NOT NULL ) ;
```

Nous avons également besoin d'une séquence pour contrôler les numéros des enregistrements. Nous créons donc également la séquence « user_message_seq ».

```
CREATE TABLE noas_exchange_sequence (
  NAME VARCHAR(100),
  VALUE BIGINT ) ;
INSERT INTO noas_exchange_sequence (NAME, VALUE)
VALUES
(' user_message_seq ', 1 ) ;
```

Nous vous rappelons que le processus de création de séquence dépend du SGBDR utilisé. Consultez la documentation pour plus d'information.

Mapping objet / relationnel

Pour mettre en relation la classe métier « UserMessageEntity » (UserMessageEntity.class.php) et la table « t_user_message », utilisons un « proxy » qui nous permettra de contrôler le transfert des objets. Le framework fournit une variante, « Wizard », basée sur un fichier de description XML. La syntaxe est intuitive, hormis quelques éléments que nous allons préciser.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE exchange-description PUBLIC "-//Noas PHP,
The Framework.//DTD Exchange Description 1.0//FR"
"http://noas.sourceforge.net/dtds/noas-exchange_1_0.dtd">
<exchange-description>
  <entity>UserMessageEntity</entity>
  <exchange>t_user_message</exchange>
  <code>USER_MSG</code>
  <lazy>true</lazy>
  <desc>false</desc>
  <field>
    <type>INTEGER</type>
    <code>USER_MSG_ENTRYID</code>
    <property>SID</property>
    <exchange>sid</exchange>
    <size>10</size>
    <notnull>true</notnull>
    <readonly>true</readonly>
    <writeonly>true</writeonly>
    <primary>true</primary>
    <sequence>user_message_seq</sequence>
  </field>
  ...
  <field>
    <type>TIMESTAMP</type>
    <code>USER_MSG_DATE_REGISTER</code>
    <property>dateRegister</property>
    <exchange>date_register</exchange>
    <notnull>true</notnull>
    <readonly>true</readonly>
    <writeonly>true</writeonly>
  </field>
</exchange-description>
```

L'élément « <type/> » correspond au type du champ dont les valeurs possibles sont : "VERSION", "INTEGER", "NUMERIC", "STRING", "TEXT", "BLOB", "BOOLEAN", "TIMESTAMP".

L'élément « <code/> » sert à mieux distinguer les champs et peut être utilisé comme identifiant de libellé GUI.

L'élément « <property/> » correspond à la propriété de classe associée.

L'élément « <exchange/> » correspond à la colonne de table associée.

Les éléments « <readonly/> » et « <writeonly/> » correspondent aux sens d'écriture autorisés.

L'élément « <sequence/> » correspond à la séquence numérique à utiliser pour incrémenter automatiquement la valeur du champ.

Les fichiers de description ne se limitent pas qu'aux simples champs. Vous pouvez également déclarer des agrégats ou des références qui

seront gérées automatiquement (mise à jour et suppression en cascade) en utilisant les éléments « <aggregate/> » et « <reference/> ». Notre « proxy » ne fera pas grand-chose, si ce n'est se référencer auprès du manager et définir la date de création. (UserMessageEntity Proxy.class.php).

Test unitaire

```
////////////////////////////////////
//
////////////////////////////////////
// File : UserMessageTestCase.class.php
////////////////////////////////////
//
////////////////////////////////////
...
protected function runTest(NoasDumpContext $dc){
    $oTrans = new NoasExchangeTransaction("demo");
    $oTrans->open();

    $dc->trace("create ...");
    $oTrans->beginTransaction();
    $oUserMsg = new UserMessageEntity();
    $oTrans->create($oUserMsg);
    $oTrans->commit();

    $oUserMsg->setSubject("Hello world !");
    $oUserMsg->setEmail("sona@noas.com");
    $oUserMsg->setMessage("Noas PHP framework,
ready for download !");

    $dc->trace("save ...");
    $oTrans->beginTransaction();
    $oTrans->save($oUserMsg);
    $oTrans->commit();

    $dc->trace("find ...");
    $oCriterion = new UserMessageEntity();
    $oCriterion->setEmail("sona@noas.com");
    $oList = $oTrans->find($oCriterion);
    $oIterator = $oList->iterator();
    if($oIterator->hasNext()){
        $oFound = $oIterator->next();
        $oTrans->beginTransaction();
        $oTrans->refresh($oFound);
        // $oTrans->delete($oFound);
        $oTrans->commit();
    }
    $oTrans->close();
}
...
}
```

■ Réoka DJOHOU - Ingénieur Études

<http://noas.sourceforge.net>

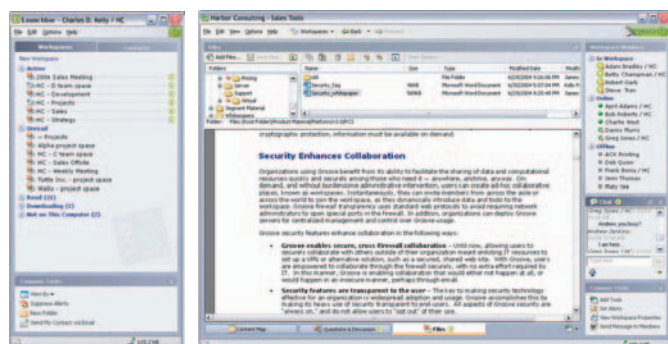
<http://www.sourceforge.net/projects/noas/>

Groove Virtual Office : une architecture originale dédiée au travail en équipe

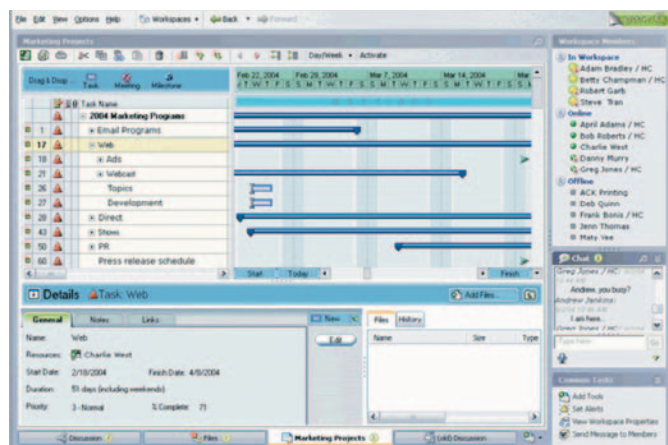
SOURCES DE L'ARTICLE
WWW.PROGRAMMEZ.COM

Groove Virtual Office est un outil sécurisé permettant à des personnes locales, distantes ou mobiles de travailler ensemble. L'outil s'architecture autour de la notion d'espaces de travail : un environnement regroupant un ensemble d'interlocuteurs prédéfinis. Il est construit par assemblage d'outils basiques pour répondre à un besoin particulier. Par exemple, gérer la relation avec un partenaire, suivre le déroulement d'un projet, organiser un service... Chaque utilisateur peut créer à volonté des espaces de travail et y inviter d'autres personnes.

Une des particularités de Groove Virtual Office provient de sa richesse fonctionnelle. Le produit fournit en standard de nombreux outils de communication : chat, messagerie instantanée, forum, VoIP, mais également des outils d'organisation et de collaboration : calendrier partagé, carnet d'adresses partagé, mise en commun de fichiers, gestion de projets, co-édition et coprésentation de documents, etc. Groove Virtual Office s'interface très facilement avec la suite logicielle Microsoft Office.



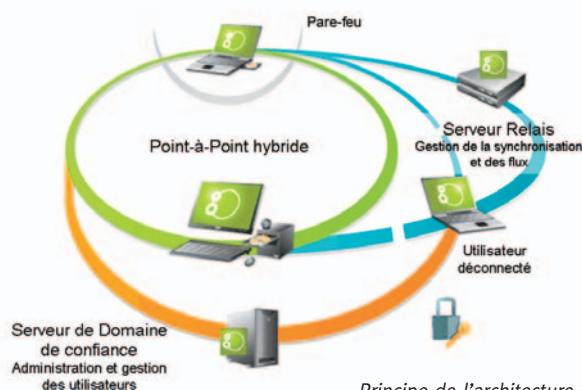
Interface utilisateur de Groove Virtual Office.



Une architecture technique hors norme

Groove Virtual Office se distingue par une architecture hors norme, dite Peer to Peer Hybride. L'approche P2P permet à chaque utilisateur de disposer de ses informations sur sa propre machine. Il peut ainsi travailler dans des lieux différents (bureau, maison, transports), en étant connecté ou déconnecté du réseau (Internet ou réseau d'entreprise). Lorsqu'il se reconnecte, les espaces de travail partagés avec d'autres utilisateurs ou entre plusieurs machines sont automatiquement re-synchronisés.

Le mode hybride fait intervenir des serveurs relais : Lorsque plusieurs personnes d'un même espace de travail ne sont pas connectées simultanément, le serveur relais remplit le rôle de zone tampon, permettant de stocker de façon temporaire les éléments, fragmentés et cryptés, des contenus échangés entre ces personnes. Le serveur relais permet également d'assurer un fonctionnement « cross-firewall » : On peut ainsi relier des réseaux d'entreprises distantes, transformant la plate-forme en Extranet sans coûts d'infrastructure (télécoms, réseaux) complémentaires : Une solution souple pour travailler avec des partenaires externes



Principe de l'architecture Groove.

sur différents projets et créer en fonction de ses besoins de véritables « VPN à la demande ». De plus, dans leur configuration adaptée aux entreprises (configuration dite « sous domaine de confiance »), les technologies Groove permettent de bénéficier d'une gestion centralisée des utilisateurs, des machines et des usages.

En adoptant ce type de fonctionnement, Groove Virtual Office tire ainsi parti des bienfaits des approches P2P et Client-Serveur.

Une plate-forme adaptable et ouverte

Si la plate-forme dispose déjà de nombreux outils intégrés, il est pourtant possible d'en étendre encore les capacités par différents moyens et notamment en ayant recours aux Web Services. Groove Virtual Office expose des services qui permettent à une application externe de pouvoir interagir avec les données stockées dans les espaces de travail. C'est ce point que nous allons illustrer dans la suite de l'article.

Team Alarm, une mise en œuvre des Web Services sous Groove Virtual Office

Le scénario d'usage

Vous disposez d'un espace de travail que vous partagez avec plusieurs autres membres de votre équipe. Votre objectif est d'initier une réunion en regroupant l'ensemble des membres au sein de cet espace de travail. Pour cela, vous devez donc vérifier la disponibilité de ces personnes. L'application que nous allons mettre en œuvre se charge d'effectuer cette surveillance pour vous. Après avoir sélectionné un espace Groove, l'application va analyser périodiquement le statut des personnes sous Groove et vous avertir dès que les personnes seront toutes disponibles.

Les pré-requis

La mise en œuvre de l'exemple présenté nécessite de disposer :

- De Groove Virtual Office installé et d'un compte utilisateur (Téléchargement possible d'une version d'évaluation gratuite).
- Du kit de développement Web Services.

Ces deux éléments peuvent être téléchargés sur <http://programmez.homesetprocess.com>. L'installation et la création d'un compte ne demandent que quelques minutes.

Le cheminement

Différents Web Services disponibles sous Groove vont être utilisés dans notre exemple :

Le Web Services « Account » va permettre de déterminer les identités Groove disponibles sur la machine. Par souci de simplification dans cet exemple, nous considérons qu'une seule identité est disponible.

Le Web Services « Spaces » permet ensuite d'obtenir la liste des espaces de travail créés par l'utilisateur Groove courant.

Le choix de l'un des espaces et le recours au Web Service « Members » nous permettent par la suite de disposer de la liste des membres de l'espace choisi.

Nous pouvons finalement interagir avec les utilisateurs et les espaces, en utilisant un Web Service spécifique nommé « Local » dont le rôle

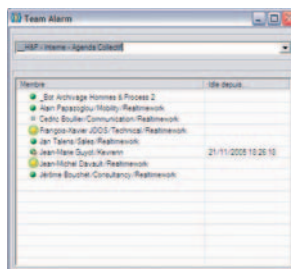
est de gérer toutes les interactions de l'utilisateur Groove local avec l'interface graphique. Il permet notamment d'envoyer des invitations et d'ouvrir la fenêtre d'un espace de travail.

Le développement

Le développement de l'application va se faire en C# sous Visual Studio 2003. Pour cela, créez un nouveau projet de type Application Windows. Nommez le projet « TeamAlarm ».

L'interface graphique de l'application est simple. Sur le « form », ajoutez simplement

- une comboBox qui va servir à lister l'ensemble des espaces
- une listView qui va servir à lister l'ensemble des membres d'un espace donné ainsi que leur statut
- un timer : pour relancer la vérification de façon périodique
- un imageList pour stocker les images représentant les statuts



L'application en ordre de marche

Ajoutez ensuite une référence vers les Web Services fournis par Groove. Pour cela, dans le navigateur de solutions, effectuez un clic droit sur « référence », puis cliquez sur « Add Web Référence ». Saisissez le chemin d'accès suivant « C:\Program Files\Groove Networks\Groove Development Kit\Web Services\Lib\XML\ 2.0\wsdl\GrooveAccounts.wsdl » et cliquez sur « Aller à ». Les méthodes disponibles sur ce service web sont listées. On vous demande également de saisir un nom de référence Web. Entrez « gwsAccounts » puis cliquez sur « Ajouter la référence ».

Reproduisez cette opération avec les fichiers « GrooveSpaces.wsdl », « GrooveMembers.wsdl » et « GrooveLocal.wsdl » en nommant respectivement les références « gwsSpaces », « gwsMembers » et « gwsLocal ».

Les capacités d'appel d'un web service nécessitent de disposer d'une URL d'appel et d'un token de sécurité.

Dans la méthode « Load » du « form » principal, on retrouve donc le code suivant :

```
// Positionnement de la base d'URL par défaut utilisée pour les appels
Web Services locaux
this.strURL = "http://localhost:9080/GWS/Groove/2.0/";

// Obtention du token de sécurité
this.getRequestKey();

// Obtention de l'identité Groove installée sur le poste local
this.getIdentity();

// Obtention de la liste des espaces de travail installés sur le compte
Groove local
```

```
this.getSpaces();
```

La méthode `getRequestKey()` lit le token de sécurité dans la base de registre. Disponible uniquement lorsque Groove est démarré, sa valeur change à chaque nouveau lancement. Ce token permet de s'assurer que l'appel s'effectue en local et que l'application appelante dispose bien des droits d'accès à Groove.

```
private void getRequestKey()
{
    Microsoft.Win32.RegistryKey pRegKey =
Microsoft.Win32.Registry.CurrentUser.OpenSubKey("Software\\Groove
Networks, Inc.\\Groove\\WebServices");
    this.strRequestKey = (string)
pRegKey.GetValue("LocalRequestKey");
}
```

Une fois les éléments nécessaires obtenus, il est possible de réellement lancer des appels Web Services. C'est ce qui est effectué dans les méthodes « `getIdentity` » et « `getSpaces` »

```
private void getSpaces()
{
    try
    {
        // Création d'une instance du
service Groove Spaces
        gwsSpaces.GrooveSpaces
svcSpaces= new gwsSpaces.GrooveSpaces ();

        // Remplissage du Header
avec les informations nécessaires
        // Pour le service Groove
Spaces, le token de sécurité
        // ainsi que l'identité (sous
forme d'URI) de l'utilisateur Groove courant
        // sont nécessaires

        svcSpaces.GrooveRequestHeaderValue = new
gwsSpaces.GrooveRequestHeader();

        svcSpaces.GrooveRequestHeaderValue.GrooveRequestKey =
this.strRequestKey;

        svcSpaces.GrooveRequestHeaderValue.GrooveIdentityURL = this.identity
URL;

        // Positionnement de l'adresse
du service Groove Spaces
        // sous la forme d'une adresse
générique + d'une URI spécifique
        // correspondant aux espaces
de l'utilisateur Groove courant
        svcSpaces.Url = this.strURL +
```

```
"Spaces/" + this.identity.Spaces;

        // Lecture de l'ensemble des
espaces Groove disponibles
        gwsSpaces.Space[]
pTabSpaces = svcSpaces.Read("");

        // Pour chaque espace dispo
nible, création d'une instance d'une classe de stockage
        // et ajout de cette instance
dans la comboBox
        foreach (gwsSpaces.Space
pSpace in pTabSpaces)
        {
            this.cbListeEspaces.Items.Add(new clsEspace(pSpace));
        }

        // Tri de la comboBox en ordre
alphabétique
        this.cbListeEspaces.Sorted =
true;

        // Purge du service Groove
Spaces
        svcSpaces.Dispose();
    }
    catch
    {
        MessageBox.Show
("Impossible de lister les espaces\r\n(Le compte est-il bien ouvert ?)");
    }
}
```

Les appels aux Web Service se font sur un modèle qui est assez statique : seul le nom du Service Web invoqué change d'une fois sur l'autre. Dans le cas des espaces Groove, la comboBox est remplie à l'aide d'instances d'une classe créée spécialement. Le rôle de cette classe est certes de stocker l'objet « Space » initial, mais également de surcharger la méthode « `ToString` » de base, afin de contrôler l'affichage de la comboBox.

```
// Propriété privée interne
private gwsSpaces.Space space = null;

public clsEspace(gwsSpaces.Space iSpace)
{
    this.space = iSpace;
}

public override string ToString()
{
    return this.space.Name;
}
```

Une fois la comboBox remplie, l'application est ainsi prête à l'emploi et en attente d'une action utilisateur. Lorsque ce dernier choisit un espace Groove dans la liste, la méthode « GetMembers » est appelée. Celle-ci permet de récupérer l'ensemble des membres de l'espace sélectionné et d'analyser leur statut.

```
private bool getMembers()
{
    // Boolean permettant de savoir si l'en-
    // semble de membres de l'espace sont disponibles
    bool boolAllConnected = true;

    // Purge de la listView de stockage des
    // membres
    this.lvMembres.Items.Clear();

    // Récupération et Cast de l'espace
    // Groove choisi dans le ComboBox
    clsEspace pCurrentSpace = (clsEspace)
    this.cbListeEspaces.SelectedItem;

    // Création d'une instance du service
    // Groove Members
    gwsMembers.GrooveMembers
    svcMembers= new gwsMembers.GrooveMembers();

    // Remplissage du Header avec les
    // informations nécessaires
    // Pour le service Groove Members,
    // le token de sécurité
    // ainsi que l'identité (sous forme d'URI)
    // de l'utilisateur Groove courant
    // sont nécessaires
    svcMembers.GrooveRequestHeaderValue
    = new gwsMembers.GrooveRequestHeader();

    svcMembers.GrooveRequestHeaderValue.GrooveRequestKey =
    this.strRequestKey;

    svcMembers.GrooveRequestHeaderValue.GrooveIdentityURL = this.
    identity.URI;

    // Positionnement de l'adresse du service
    // Groove Members
    // sous la forme d'une adresse générique
    // + d'une URI spécifique
    // correspondant aux membres de
    // l'espace Groove courant
    svcMembers.Url = this.strURL +
    "Members/" + pCurrentSpace.strMembersURI;

    // Lecture de l'ensemble des membres de
    // l'espace Groove courant
    gwsMembers.Member[] pTabMembers =
    svcMembers.Read();
```

```
// Vérification que la personne courante
// n'est pas le seul membre de l'espace choisi
if (pTabMembers.Length > 1)
{
    // Pour chacun des membres
    // foreach (gwsMembers.Member
    // pMember in pTabMembers)
    {
        // Si le membre
        // courant correspond à l'identité Groove courante
        // Alors il est exclu
        // du traitement
        if
        (!pMember.URI.Equals(this.identity.URI))
        {
            //
            // Analyse de la propriété "TimeWentIdle"
            // Si le
            // champ TimeWentIdle est égal à 01/01/0001 00:00:00, c'est que la
            // personne n'est pas Idle
            string
            pStrTimeWentIdle = pMember.TimeWentIdle.ToString();
            if
            (pStrTimeWentIdle.ToString().Equals("01/01/0001 00:00:00"))
            pStrTimeWentIdle = "";

            //
            // Création d'une nouvelle entrée dans la ListView
            string[]
            pStrElementsItem = new string [2] {pMember.Name,pStrTimeWentIdle};
            ListViewItem pItem = new ListViewItem(pStrElementsItem);

            // L'objet
            // membre est rattaché au ListViewItem pour usage ultérieur
            pItem.Tag
            = pMember;

            //
            // Analyse du statut du membre dans l'espace.
            // 1.
            // Actif dans l'espace ?
            // Le
            // membre est actif si la propriété ActiveInSpace n'est pas vide
            if
            (pMember.ActiveInSpace != null &&
            (!pMember.ActiveInSpace.Equals("")))
            {
                if (pMember.TimeWentIdle.ToString().Equals("01/01/0001 00:00:00"))
                {
                    // 1.1 Actif Non Idle ?
```

```

        pltem.ImageIndex = 3;

        boolAllConnected = boolAllConnected && true;
    }
else
{
    // 1.2 Actif Idle ?

    pltem.ImageIndex = 4;

    boolAllConnected = false;
}

// 2. Connecté à Groove ?

// Le membre est connecté si la propriété Awareness a la valeur
"ONLINE"
else if
(pMember.Awareness != null &&
pMember.Awareness.ToUpper().Equals("ONLINE"))
{
    if (pMember.TimeWentIdle.ToString().Equals("01/01/0001 00:00:00"))
    {
        // 2.1 Connecté Non Idle ?

        pltem.ImageIndex = 1;

        boolAllConnected = boolAllConnected && true;
    }
else
{
    // 2.2 Connecté Idle ?

    pltem.ImageIndex = 2;

    boolAllConnected = false;
}
}
}

```

```

// 3. Non connecté ?
else
{
    pltem.ImageIndex = 0;

    boolAllConnected = false;
}

// Ajout
de l'item courant à la ListView
this.lvMembres.Items.Add(pltem);
}

// Si toutes les personnes sont
connectées et disponibles alors on lance une alarme
if (boolAllConnected)
    sendAlarm();

return true;
}
else
{
    MessageBox.Show("Aucun
autre membre dans cet espace !");
return false;
}
}

```

Pour chacun des membres, le statut est analysé. Il peut prendre les valeurs Offline, Online, Online mais idle, Actif, Actif mais idle. Chaque statut est illustré au travers d'une icône stockée dans le composant ImageList et rattachée à la ListView.

Dans le cas où un profil est « idle », on affiche le temps d'inactivité. La valeur d'un booléen global est évaluée pour chaque itération sur un membre. Cela permet au final de bénéficier d'une valeur spécifiant si tous les membres sont ou non disponibles.

Cette méthode est appelée périodiquement – toutes les 10 secondes ici – afin d'évaluer les changements de statuts des membres.

Lorsque les personnes sont disponibles, on affiche un message à l'utilisateur, lui demandant s'il souhaite que les membres soient invités à rejoindre l'espace de travail sélectionné. Cette dernière action est effectuée en utilisant le service web « Local ».

■ **Fabrice BARBIN**

fabrice.barbin@hommesetprocess.com

Responsable Technologies Hommes & Process



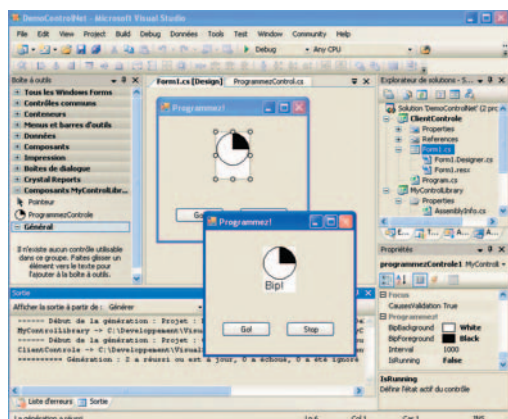
Unique distributeur officiel des technologies Groove (Groove Networks Inc / Microsoft Corp) sur le territoire français, Hommes & Process fournit des solutions innovantes et des services d'optimisation du travail en équipe.

Réaliser un contrôle utilisateur sous Visual Studio .Net

SOURCES DE L'ARTICLE
WWW.PROGRAMMEZ.COM

La réutilisation de code est un point fort de la programmation objet. Réutiliser du code avec un simple clic de souris s'obtient en créant un composant s'intégrant dans un environnement de développement visuel. Nous réalisons aujourd'hui cela sous Visual Studio .Net.

Un contrôle utilisateur est un composant logiciel intégré à la boîte à outils de Visual Studio, pour être utilisé lors de la conception visuelle de fiches, ou Form, pour reprendre l'appellation .Net. Selon son type, le contrôle sera ou non visible à l'exécution de ladite fiche, mais sera dans tous les cas visible en phase de conception. Réaliser un tel contrôle utilisateur et assurer son intégration est un travail simple quand on a saisi le principe, pourvu que l'on ait une connaissance de base du langage utilisé, ainsi que des attributs qui sont une fonctionnalité des langages .Net. Nous allons faire ce travail ensemble. Nous utilisons Visual Studio 2005 bêta, mais ce que nous dirons est valable pour les versions antérieures et postérieures, ainsi que les versions Express. Nous avons choisi le langage C#, mais n'importe quel autre langage .Net aurait pu être choisi. De même, le contrôle pourra être utilisé avec n'importe quel autre langage .Net. Nous écrivons également un petit client, lui aussi en C#, pour mettre en œuvre le contrôle. Comme d'habitude, vous trouverez toutes les sources sur notre site : www.programmez.com.



Notre contrôle, intégré à l'environnement Visual Studio

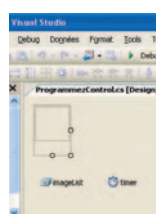
1 Notre objectif

Comme le montre la figure nous créons un contrôle avec un cadran rotatif. La rotation est cadencée par un timer. Lorsqu'un tour complet est effectué, le message "Bip !" s'affiche sous le cadran et un événement personnalisé est émis. Si vous voulez approfondir le mécanisme des événements personnalisés sous .Net, vous pouvez vous reporter à *Programmez ! 82*. L'animation est réalisée en affectant une image à un contrôle PictureBox au rythme des événements émis par le timer. Les images, 4 en tout, sont rangées dans un contrôle ImageList. Notre

contrôle en contient donc trois autres, Timer, PictureBox, ImageBox. Notre contrôle est configurable depuis l'environnement. En plus des propriétés héritées, nous définissons 4 propriétés pour gérer la couleur de fond et la couleur d'avant-plan du bip, la vitesse d'animation et le départ/arrêt de l'animation. Enfin, bien que cela n'apparaisse pas sur l'illustration, notre contrôle expose l'événement personnalisé dans l'environnement. L'écriture d'un gestionnaire d'événement client se fera donc comme avec tout contrôle .Net.

2 Création du projet

La première chose à faire est de créer un projet de type 'Windows Control Library'. À la suite de quoi, un espace de nom est créé, qui contient une classe dérivant de la classe System.Windows.Forms.UserControl. C'est la classe de base de tout contrôle visuel. Si vous souhaitez créer un composant non visuel, il vous suffit de modifier le code pour que votre classe dérive de System.ComponentModel.Component. La classe UserControl dérive elle-même de la classe Component. Modifiez ensuite le nom de votre classe et de son fichier source à votre convenance. Ceci est suffisant pour notre propos. Toutefois, si vous devez développer un composant plus élaboré que notre exemple, vous tirerez bénéfice à examiner la



Comme les fiches, les contrôles disposent d'un concepteur visuel.

documentation des deux classes de base, qui contiennent moult méthodes très pertinentes pour vous faciliter la tâche. Déposez ensuite les composants de votre contrôle dans le concepteur visuel. Pour bien faire, les éléments constitutifs du contrôle devraient être redimensionnés à la volée, lorsque le conteneur est redimensionné. Notre exemple ne le fait pas, par souci de ne pas alourdir le code. C'est une formalité que d'implémenter cette fonctionnalité dans le gestionnaire de l'événement Resize.

3 Écrire le code

Le code complet de notre contrôle est donné dans l'encadré.

Le code complet de notre contrôle

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
```

```

using System.Data;
using System.Text;
using System.Windows.Forms;

namespace MyControlLibrary
{
    [DefaultEvent("OnBipHandler"), DefaultProperty("IsRunning"),
    ToolboxBitmap(
    @"C:\Developpement\VisualStudio2005Beta2\DemoControlNet\Ressources\icone.bmp")]

    public partial class ProgrammezControle : UserControl
    {
        public delegate void BipHandler(object source,
        EventBip e);

        [Category("Programmez!"),
        Description("Envoyé à chaque tour de cadran")]
        public event BipHandler OnBipHandler;

        private Rectangle bipZone;
        private int imageCourante = 0;

        protected bool isrunning;
        protected bool bip = false;
        protected int interval = 1000;
        protected Color bipbackground = Color.White;
        protected Color bipforeground = Color.Black;

        [Category("Programmez!"),
        Description("Définir la vitesse"),
        DefaultValue(1000)]
        public int Interval
        {
            get { return interval; }
            set
            {
                interval = value;
                timer.Interval = interval;
            }
        }

        [Category("Programmez!"),
        Description("Définir l'état actif du contrôle")]
        public bool IsRunning
        {
            get { return isrunning; }
            set
            {
                isrunning = value;
                timer.Enabled = isrunning;

                if (isrunning == false)
                {
                    bip = false;

```

```

                    imageCourante = 0;
                    pictureBox.Image =

                    faireBip();
                }
            }
        }

        [Category("Programmez!"),
        Description("Couleur d'arrière plan du bip")]
        public Color BipBackground
        {
            get { return bipbackground; }
            set
            {
                bipbackground = value;
                Invalidate();
            }
        }

        [Category("Programmez!"),
        Description("Couleur d'avant plan du bip")]
        public Color BipForeground
        {
            get { return bipforeground; }
            set
            {
                bipforeground = value;
                Invalidate();
            }
        }

        public ProgrammezControle()
        {
            InitializeComponent();
            bipZone = new Rectangle(0, 51, 51, 20);
            pictureBox.Image = imageList.Images[0];
        }

        private void timer_Tick(object sender, EventArgs e)
        {
            imageCourante++;
            pictureBox.Image =
            imageList.Images[imageCourante % 4];
            if (imageCourante > 3)
            {
                imageCourante = 0;
                bip = true;
                faireBip();
                envoyerBip();
            }
            if (imageCourante == 1)
            {
                bip = false;
                faireBip();

```

```

    }
}

private void faireBip()
{
    Invalidate();
}

private void envoyerBip()
{
    EventBip eb = new EventBip("Bip !!");
    if (OnBipHandler != null)
        OnBipHandler(this, eb);
}

private void ProgrammezControle_Paint(object
sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    g.FillRectangle(new SolidBrush(bipback
ground), bipZone);

    if (bip == true)
        g.DrawString("Bip!",
                    new Font("Arial",
12),
                    new SolidBrush(bip
foreground), bipZone);
}

public class EventBip : EventArgs
{
    private string message;

    public EventBip(string message)
    {
        this.message = message;
    }

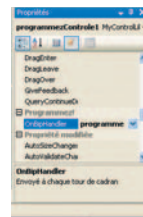
    public string Message
    {
        get { return message; }
    }
}
}

```

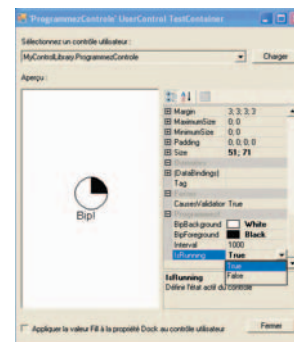
Commentons ce code, en commençant par la classe de l'événement personnalisé. Cette classe se situe à la fin de notre code. Pour une raison mal déterminée, votre serveur a rencontré des difficultés à l'exécution si la classe est placée en début de code. Venons en maintenant au contrôle proprement dit. Juste avant la classe viennent des attributs. Visual Studio .Net utilise les attributs et le mécanisme de réflexion pour intégrer les contrôles à l'environnement. Les trois premiers attributs, puisque placés avant la classe, concernent celle-ci dans son ensemble. DefaultEvent permet de définir l'événement par défaut. Moyennant quoi,

lorsque l'utilisateur du contrôle double-clique dessus en phase conception, un squelette de code pour le gestionnaire de l'événement est automatiquement généré. DefaultProperty définit la propriété sélectionnée par défaut dans la feuille de propriété. Enfin, ToolboxBimap pointe sur l'image 16x16 qui servira de symbole visuel pour le contrôle dans la boîte à outils. Vous veillerez à adapter le chemin absolu du fichier .bmp à votre configuration.

Vient ensuite la déclaration du délégué pour invoquer les gestionnaires d'événements clients, puis la déclaration de l'événement lui-même. Cette déclaration est précédée de deux attributs. L'attribut Category permet de définir une entrée dans la feuille de propriété, tandis que l'attribut Description définit la chaîne d'aide qui apparaît lorsqu'on clique sur la propriété depuis la feuille. Ensuite viennent les propriétés de la classe, elles aussi dotées d'attributs, selon les mêmes principes. Lorsqu'on ajoute un attribut DefaultValue la valeur est utilisée par Visual Studio lorsqu'il effectue des réinitialisations des propriétés. Quand le code est écrit, il est possible de tester le composant en compilant et exécutant le projet. Le composant apparaît alors dans un conteneur prévu à cet effet. La feuille de propriétés du conteneur est directement utilisable pour tester les réactions de votre composant aux changements des valeurs.



La chaîne d'aide d'une propriété ou d'un événement se définit avec un attribut.



Test du comportement du composant en phase de développement.

4 Un peu de décoration

Pour afficher notre bip dans le composant, nous avons choisi de passer par un événement Paint déclenché par l'invocation de la méthode Invalidate du composant. Cette démarche n'est pas obligatoire. On peut peindre le composant depuis n'importe quelle méthode. Le tout est alors de se procurer un objet Graphics. Deux possibilités, et dans chaque cas, l'objet doit être libéré après utilisation :

```

Graphics g = this.CreateGraphics();
// ou bien
// Graphics g = Graphics.FromHwnd(this.Handle);

/* dessiner ici puis */
g.Dispose();

```

Rappelons que dans le cas d'un gestionnaire d'événement OnPaint, la méthode Dispose ne doit pas être invoquée manuellement puisque l'émetteur de l'événement s'en charge.

5 Utiliser le contrôle dans une application client

Créez un nouveau projet de type 'Windows Application'. La boîte à outils contenant tous les composants est alors disponible. Nous devons y intégrer notre composant, ce qui se fait depuis le menu contextuel, puis par 'Sélectionner les éléments'. Dans la boîte de dialogue qui suit, cliquez sur 'Parcourir' pour naviguer vers l'assembly, c'est-à-dire le fichier .dll de notre contrôle. Validez. Le contrôle est automatiquement intégré à la boîte à outil et prêt à être utilisé. Une référence à l'assembly est automatiquement ajoutée au projet. Notre client d'exemple comporte deux boutons pour démarrer ou arrêter l'animation en affectant la propriété `IsRunning` du contrôle.

Notre client utilisant le contrôle.

```
//etc
using MyControlLibrary;

namespace ClientControle
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            programmezControle1.IsRunning = true;
        }

        private void button2_Click(object sender, EventArgs e)
        {
            programmezControle1.IsRunning = false;
        }
    }
}
```

```
private void programmezControle1_OnBipHandler(
    object source, EventArgs e)
{
    MessageBox.Show(e.Message);
    programmezControle1.IsRunning = false;
}
}
```

Nous avons aussi écrit un gestionnaire d'événement pour afficher le message contenu dans l'événement personnalisé. Pour cela, faites un double-clic sur le contrôle, et automatiquement, Visual Studio va générer un squelette pour la méthode. On remarque avec satisfaction que c'est bien le type de notre événement personnalisé qui est passé à la méthode. Plus généralement, Visual Studio génère automatiquement, dans le corps de la méthode `InitializeComponent`, tout le code travaillant avec les propriétés du contrôle, comme le montre l'extrait ci-dessous.

Extrait du code client automatiquement généré

```
this.programmezControle1.BipBackground =
    System.Drawing.Color.White;
this.programmezControle1.BipForeground =
    System.Drawing.Color.Black;
this.programmezControle1.IsRunning = false;
this.programmezControle1.Location =
    new System.Drawing.Point(90, 26);
this.programmezControle1.Name =
    "programmezControle1";
this.programmezControle1.Size =
    new System.Drawing.Size(51, 71);
this.programmezControle1.TabIndex = 0;
this.programmezControle1.OnBipHandler +=
    new MyControlLibrary.ProgrammezControle.BipHandler(
    this.programmezControle1_OnBipHandler);
```

■ Frédéric Mazué fmazue@programmez.com

Offre 2006

Abonnez-vous et recevez ce magnifique
Couteau suisse - clé USB 128 Mo
haut de gamme !

Pour le prix normal du magazine (prix au numéro), soit 65€ seulement !

Découpez le coupon d'abonnement, ou abonnez-vous en ligne : www.programmez.com



Les bases du langage Erlang (2^e partie)

Il y a deux mois, nous avons découvert Erlang, un langage Open Source très original. Aujourd'hui nous nous familiarisons avec son expressivité.

Dans le numéro 81, nous avons écrit le traditionnel programme HelloWorld en Erlang. En voici le code :

```
-export([direbonjour/0]).  
  
direbonjour() ->  
    io:format("Hello World!!\n").
```

Nous constatons que la syntaxe d'Erlang n'a rien de commun avec des langages clonés, tels que Java ou C# :-). Nous allons donc commenter ligne par ligne ce code riche en enseignements.

1 Notion de module et d'unité de Programmation

En Erlang, l'unité de programmation est le module. Concrètement, un fichier contenant du code. Il doit y avoir correspondance entre le nom du fichier, celui-ci étant doté d'une extension .erl et le nom du module. Normalement, tout module commence par une ligne déclarant le nom du module. C'est ce que fait notre première ligne de code qui déclare un module hello pour un code résidant dans un fichier hello.erl. Remarquez bien le point, indispensable, qui termine la ligne de code.

2 Notion d'interface de module et d'arité de fonction

Ensuite, un module doit déclarer les fonctions qui seront visibles de l'extérieur du module. Ces fonctions sont dites exportées. La deuxième ligne de code signifie donc que notre module exporte une seule fonction : direbonjour/0. Les noms des fonctions exportées sont toujours énumérés dans une liste, même lorsqu'il n'y a qu'une seule fonction, comme c'est le cas ici. Un nom de fonction exporté va toujours de pair avec l'arité de ladite fonction, afin d'en constituer la signature. On appelle arité d'une fonction, le nombre de paramètres qu'elle reçoit lorsque qu'on l'invoque. Notre fonction reçoit zéro paramètre, elle est donc identifiée par direbonjour/0. À nouveau, notez bien le point qui termine l'instruction.

Un module peut fort bien contenir des fonctions qui ne sont pas exportées. Dans ce cas, les fonctions sont privées et inaccessibles de l'extérieur, tout comme les méthodes privées d'une classe sont inaccessibles, pour faire une analogie avec la programmation objet. Pour prendre une autre analogie, nous pouvons dire que l'ensemble des fonctions exportées constitue une interface publique, et l'on peut voir le module comme

une boîte noire, tout comme un objet est une boîte noire, hormis son interface publique

Remarquons qu'un module peut très bien contenir des fonctions de même nom, mais d'arités différentes, que ces fonctions soit exportées ou non. Ainsi, notre module pourrait contenir une fonction recevant un paramètre. Sa signature serait alors direbonjour/1. Si cette fonction est privée, elle sera de toute façon inaccessible. Si elle est exportée, elle sera accessible par son nom et à condition de recevoir un argument. Deux fonctions de même nom, mais d'arités différentes doivent être exportées explicitement, comme ceci pour notre exemple :

```
-export([direbonjour/0, direbonjour/1]).
```

3 Anatomie d'une fonction

Vient ensuite et enfin, le code proprement dit. Notre fonction est définie sur deux lignes. Nous aurions pu écrire le tout sur une seule ligne. Cette présentation est une convention qu'il est pertinent de suivre. Vient donc le nom de la fonction, suivi de -> qui, intuitivement, indique le chemin suivi par le flux d'exécution, autrement dit le code qui sera exécuté. Ce code est sur la deuxième ligne et consiste en l'appel de la fonction format du module io. Tous les appels de fonction, en dehors du module courant, sont écrits ainsi en Erlang : module : fonction. Encore une fois, remarquez le point final qui termine notre fonction. Si notre fonction faisait plusieurs choses séquentiellement, imprimer deux lignes de suite par exemple, les séquences seraient séparées par des virgules, jusqu'au point final :

```
direbonjour() ->  
    io:format("Hello World!!\n"),  
    io:format("Abonnez vous :-)\n").
```

4 Evaluation et valeur de retour

Intéressons nous maintenant de plus près à ce qui se passe à l'exécution. Si vous essayez l'exemple, vous voyez Hello World apparaître sur l'écran, ce qui correspond à une attente légitime. Ce qui est plus surprenant est le mot ok qui apparaît lui aussi. Avec les langages fonctionnels, lorsqu'on appelle une fonction, on provoque toujours une évaluation, ou une séquence d'évaluation. La dernière chose évaluée est de facto la valeur de retour de la fonction. Une fonction a donc toujours une valeur de retour. Une fonction "void" n'existe pas en Erlang. Lorsque nous avons déclaré notre fonction direbonjour, nous n'avons

rien dit quant à la présence de cette valeur de retour, puisqu'il y en a toujours une, ni rien quant à son type, car Erlang est dynamiquement typé. Si vous consultez la documentation au sujet de la fonction `format` du module `io`, vous y apprendrez que `format` retourne `ok` quand tout se passe bien. Comme `io:format` est la dernière chose évaluée dans notre fonction `direbonjour`, `ok` est retourné automatiquement par notre fonction, et en mode interactif il apparaît sur la console. Il est possible de se débarrasser de ce `ok` qui, sans cela, pourrait parasiter la sortie d'un programme en mode non interactif. Supposons que nous voulions lancer notre programme depuis un script Shell ou Batch nous procéderions ainsi :

```
erl -noinput -s hello direbonjour -s erlang halt
```

afin d'appeler successivement `hello:direbonjour`, à ce stade le `ok` n'apparaît déjà plus, puis `erlang:halt` pour arrêter le système Erlang. Le lecteur voudra bien se reporter à la documentation du module `erl` pour s'informer de toutes les possibilités de lancement d'un programme en ligne de commande. Mais fondamentalement, qu'est-ce donc que ce `ok` ?

5 Les types Erlang

Erlang comporte très peu de types, simples ou composés. Commençons par les types simples :

- Les atomes. En Erlang un atome est une suite de lettres inséparables. Pour l'humain, un atome est un mot dont la signification est conventionnelle. Par exemple : `ok`, `error`, `true`, `false` sont des atomes dont les noms parlent d'eux mêmes. Quand l'appel à la fonction `io:format` réussit, celle-ci retourne l'atome `ok`.
- Les entiers. En Erlang un entier a la particularité de ne pas être limité en taille, comme c'est le cas avec beaucoup de langages fonctionnels et contrairement à ce que l'on rencontre avec des langages impératifs tels que C++/C#/Java, etc.
- Les réels. Des nombres à virgule flottante, avec éventuellement une notation scientifique, comme dans la majorité des langages.
- Les identifiants de processus. Un identifiant de processus est une valeur construite par le système Erlang lorsqu'un processus est lancé. C'est grâce à cet identifiant qu'il est ensuite possible d'accéder au processus, un peu comme une référence permet d'accéder à une instance d'objet en programmation objet.
- Les références. Il s'agit d'identifiants uniques, générés à la demande du programmeur par le système.

Voyons maintenant les types composés :

- Les listes. Celles-ci sont omniprésentes en programmation fonctionnelle et donc en Erlang. Une liste est une suite de valeurs, délimitée par des crochets en ce qui concerne la syntaxe. Erlang étant dynamiquement typé, les éléments de la liste peuvent être de n'importe quel type, y com-

pris composé. Ainsi :

```
[1, "deux", [5,6,7]]
```

est une liste de 3 éléments qui sont dans l'ordre : un entier, une chaîne de caractères et une autre liste d'entiers.

En outre, en Erlang, une chaîne de caractères écrite sous la forme "deux" n'est qu'un sucre syntaxique. En interne, une chaîne de caractères est aussi une liste de valeurs ACII. Ainsi

```
x" ou bien [$d,$e,$u,$x] ou bien [100, 101, 117, 120]
```

sont une seule et même chose en Erlang.

- Les tuples. Ils sont semblables aux listes, avec cette différence, que l'ordre des éléments contenus dans le tuple est fixe et a une signification conventionnelle définie par le programmeur. Les tuples sont délimités par des accolades `{}`.
- Les structures ou "records". Ce sont des tuples particuliers, car dotés d'un nom et dont chaque élément est doté lui aussi d'un nom. Nous aurons l'occasion de manipuler des structures dans un article à venir.
- les binaires ou "binaries". Ce sont des suites d'octets. Ce type est apparu tardivement en Erlang, comme alternative aux listes, lorsqu'il est besoin d'optimiser le traitement de large flux de données. Nous aurons l'occasion de travailler avec des binaires dans un article à venir.

6 Travailler avec des paramètres

Il est maintenant temps d'écrire du code. Vous trouverez ce code dans un module nommé `prog82` sur le site du magazine. Modifions notre fonction `direbonjour`, afin qu'elle reçoive un paramètre. Un paramètre n'est plus ni moins qu'une variable. Son nom doit donc commencer par une majuscule :

```
io:format("Bonjour ~s~n", [Nom]),
io:format("Abonnez vous ~s~n").
```

7 Filtrage de motifs et flux d'exécution

Avec les langages traditionnels, tout un arsenal de `if`, de `then`, de `break`, de `continue`, etc. permet d'orienter le déroulement de l'exécution. En Erlang, on procède par filtrage de motifs (*pattern matching* en anglais). Le système examine une valeur en la comparant à des modèles, des motifs, et prend une décision en conséquence. Soit une fonction de nom `'un'` qui retourne un atome selon le paramètre qu'elle reçoit :

```
case Param of
    1 ->
        un_nombre;
    "un" ->
        une_chaine;
```

```

        _ ->
            un_autre
    end.

```

Le paramètre Param est filtré dans la construction case/of/end et selon la valeur reçue, un des trois atomes est retourné. Essayez dans l'interpréteur interactif les appels :

```

prog82:un("un").
prog82:un(2).
prog82:un("deux").

```

et constatez. Le caractère souligné `_` est une sorte de joker. Il veut dire n'importe quoi. Tout ce qui n'est pas filtré avant arrive là. Les filtres sont examinés de haut en bas, dans l'ordre d'écriture du programme. On place donc toujours les filtres les plus particuliers au début et les plus généraux ensuite. En ce qui concerne la syntaxe, notez bien la présence des `;` et du `.` final.

8 Les clauses

Essayez ceci dans l'interpréteur :

```
file:list_dir(".").
```

Le répertoire courant est listé, mais plein d'informations parasites apparaissent, car `list_dir` retourne son résultat dans un tuple (cf la documentation). Nous voudrions lister le contenu d'un répertoire sur une colonne, sans ces informations parasites. Commençons par filtrer ce tuple :

```

print_dir(Chemin) ->
    ResultDir = file:list_dir(Chemin),
    case ResultDir of
        {error, _} ->
            ResultDir;
        {_, FilesList} ->
            print_dir_aux(FilesList)
    end.

```

`ResultDir` est une variable contenant le tuple. Nous filtrons. Si le premier élément du tuple est `error`, nous retournons le résultat tel quel. Sinon, le deuxième élément du tuple est la liste des fichiers du répertoire. Nous passons alors cette liste à une fonction auxiliaire, privée, que voici :

```

print_dir_aux([]) ->
    ok;
print_dir_aux([Fichier | Fichiers]) ->
    io:format("~s~n", [Fichier]),
    print_dir_aux(Fichiers).

```

Il n'y a ici qu'une seule et même fonction écrite en deux clauses. Les clauses permettent de faire un filtrage de motifs directement sur les paramètres reçus. Si la fonction reçoit une liste vide, elle s'arrête après avoir retourné l'atome `ok`. Voilà pour la première clause. Si la liste n'est pas vide c'est la deuxième clause qui est exécutée. Dans ce cas, nous séparons la liste en son premier élément et son reste au moyen de la syntaxe `[Fichier|Fichiers]`. Nous imprimons ensuite ce premier élément, puis nous passons le reste de la liste à notre fonction qui est donc une fonction récursive. Dès que la liste est épuisée, nous rentrons automatiquement dans la première clause et l'exécution s'arrête. Ce traitement récursif est la façon de faire naturelle en Erlang, alors que ce serait un exercice de style dans d'autres langages. Comme pour le filtrage de motifs, on placera la clause la plus particulière avant les clauses plus générales.

9 Les gardes

Les gardes sont un autre moyen d'orienter le flux d'exécution et enrichissent la syntaxe des clauses. Soit, une fonction divisant son premier paramètre par son second :

```

divise(_, Diviseur) when Diviseur == 0 ->
    {error, "Division par zero"};
divise(Dividende, Diviseur) ->
    Dividende/Diviseur.

```

Quel que soit le dividende, si le diviseur est nul nous retournons un tuple composé de l'atome `error` et d'une chaîne expliquant l'erreur. (Étant entendu que l'erreur est seulement mathématique, car pour le système, tout va bien.) Sinon, nous retournons le résultat de la division. Nous avons dit peu sur le langage et pourtant, nous savons déjà faire énormément de choses, car Erlang, avec ses clauses et ses filtrages de motifs est très expressif et intuitif dès qu'on s'y habitue. Rien de tel qu'un peu de pratique pour se familiariser...

■ Frédéric Mazué
fmazue@programmez.com

NOUVEAU

ÉTUDIANTS :

1 an d'abonnement = 39 € seulement !

Joindre la copie de votre carte étudiant au règlement et au coupon d'abonnement et envoyer au service abonnement
 Formulaire d'abonnement en ligne : www.programmez.com

Programmez un jeu pour votre Pocket PC en .NET

Le mois dernier, nous avons observé l'architecture de NetBubbles, un jeu écrit en Visual Basic .NET pour Pocket PC. Dans cette 2^e partie, nous allons terminer notre visite du code et regarder quoi faire pour qu'il fonctionne sur Smartphone.

Un peu de récursivité

Lorsque le joueur sélectionne une bille, le jeu doit partir à la recherche de toutes les billes contiguës de même couleur. C'est parfait pour utiliser la récursivité. La fonction Test de ClasseGame s'occupe de ce travail. Il suffit de lui donner les coordonnées de la bille de départ, et sa couleur, pour qu'elle scanne récursivement le plateau de jeu et sélectionne toutes les billes contiguës de même couleur. Voici son code :

```
Public Function Find(ByVal x As Integer, _
                    ByVal y As Integer, _
                    ByVal wColor As Integer) As Integer

    Dim wCompteur As Integer

    If Me._Matrix(x, y).Color <> wColor Then
        Return 0
        Exit Function
    End If

    If Me._Matrix(x, y).Statut <>
        ClassBubble.EnumBubbleStatus.Normal Then
        Return 0
        Exit Function
    End If

    Me._Matrix(x, y).Statut =
        ClassBubble.EnumBubbleStatus.Selected
    wCompteur = 1
```

```
'Et on affiche la bille comme pointée
Me._Matrix(x, y).Show(Me._Bitmap)

'On regarde en haut si ok
If y > 0 Then wCompteur += Find(x, y -
1, wColor)

'On regarde en bas si ok
If y < Me.Rows - 1 Then wCompteur +=
Find(x, y + 1, wColor)

'On regarde à gauche
If x > 0 Then wCompteur += Find(x - 1,
y, wColor)

'On regarde à droite
If x < Me.Cols - 1 Then wCompteur +=
Find(x + 1, y, wColor)

Return wCompteur

End Function
```

Le principe est simple ; à chaque position, elle vérifie dans toutes les directions s'il y a une correspondance. Si oui, la nouvelle position est sélectionnée et devient la position initiale pour un nouveau test. La méthode Test est donc appelée récursivement.

Et la gravité dans tout ça ?

Lorsque les billes sont sélectionnées, le joueur peut confirmer son choix en cliquant sur l'une d'entre elles. S'il clique ailleurs, la sélection est annulée par un appel à la méthode CancelSelection de ClasseGame.

Lorsque la sélection est confirmée, les billes sélectionnées sont éliminées et celles qui ne sont plus supportées tombent, car elles sont soumises à la loi de la gravité (non mais !). Pour réaliser cette opération et afficher tout, au fur et à mesure des événements (ça fait plus vrai), la méthode MakeGravity de la classe ClasseGame est utilisée. Elle remplace les billes colonne par colonne en faisant tomber les billes qui sont dans le vide.

Après on referme les colonnes

Une fois les billes replacées bien solidement les unes contre les autres, il faut vérifier que la sélection n'a pas vidé complètement des colonnes. Si c'est le cas, elles sont comblées par un déplacement des colonnes placées à gauche vers la droite. CloseColumn de la classe ClasseGame effectue cette opération.

On vérifie que le jeu n'est pas terminé

La dernière opération consiste à vérifier que le jeu n'est pas terminé. Soit il n'y a plus de bille sur le plateau, soit il n'y en a plus au moins 2 contiguës de la même couleur. C'est TestFin de ClasseGame qui est chargé de cette vérification. Elle aussi utilise la récursivité.

MouseDown est le centre des opérations

Toutes ces opérations sont réalisées dans l'événement MouseDown de FormMain. C'est là que toutes les actions prennent leur source, car c'est toujours suite à un clic du joueur que l'état du jeu change. Je vous invite à analyser en détail le code de cet événement.

Avec du son c'est mieux

Les sons dans FreeBubbles sont des fichiers wav inclus au projet en tant que ressources incorporées. Pour les jouer, il est nécessaire d'utiliser les API de Windows CE. Pour ma part, je trouve bien plus simple d'utiliser la librairie CodePPC (www.codeppc.com). Cette librairie (gratuite) permet, entre autres, de jouer des sons à partir de fichiers wav et aussi, à partir de ressources incorporées. Par exemple, pour jouer la ressource « So7.wav » du projet « NetBubbles » il suffit d'écrire :

```
wSound = New CodePPC.General.Sound( _
    CodePPC.General.Tools.GetResource("NetBub
bles.S07.wav"))

wSound.Play()
```

Par défaut, la méthode Play joue le son de manière asynchrone. Si vous lui passez False

en paramètre, le son sera joué de manière synchrone, et la main sera rendue à la fin du morceau et non immédiatement après le début. La ressource incorporée est chargée grâce à la méthode `GetResource`, aussi fournie par la librairie.

Le mystère du numéro de série

Non, ce n'est pas le titre d'un nouveau film, mais simplement une explication sur l'utilisation d'un numéro de série lié à chaque partie, afin de pouvoir les rejouer à volonté.

La génération du plateau de jeu est confiée au hasard. On tire au hasard la couleur de chaque bille dans la limite du nombre maximum de couleurs de la partie (3, 4 ou 5). Tirer au hasard est en fait un bien grand mot, car le générateur de nombres aléatoires joue exactement la même séquence si on l'initialise avec le même nombre de départ.

Le seul moyen de jouer une séquence vraiment aléatoire est d'initialiser le générateur avec une valeur qui change tout le temps. La propriété `Microsoft.VisualBasic.Timer` est parfaite pour ça. Elle retourne le nombre de millisecondes écoulées depuis minuit. Il suffit donc de prendre comme numéro de série la valeur courante de cette propriété et de l'utiliser pour initialiser le générateur de nombres aléatoires, afin d'obtenir un plateau de jeu différent à chaque nouvelle partie. En mémorisant avec le score cette valeur (au format hexadécimal, ça tient moins de place à l'écran) on peut, à volonté, recréer la partie liée au score.

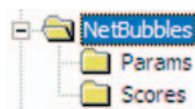
Pour initialiser le générateur de nombres aléatoires il suffit d'utiliser la méthode `Randomize(valeur)` ou `valeur`, dans notre cas, est soit la valeur courante de `Microsoft.VisualBasic.Timer`, ou le numéro de série d'une partie à rejouer. Simple comme principe, non ? C'est dans le constructeur de `ClassGame` qu'est initialisé le générateur de nombres aléatoires.

Le stockage des scores

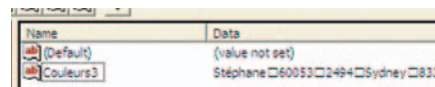
Pour le stockage des scores, plusieurs solutions s'offrent à nous. Il est possible d'utiliser un fichier plat, un fichier xml, et pourquoi pas, un moteur de base de données (pas très économique comme solution je vous l'accorde). Personnellement, j'ai décidé de stocker les scores dans la base de registre. Comme on ne garde que les 10 meilleurs scores par type de partie, on ne doit stocker que 30 éléments maximum. Comme la base de registre est « cachée » c'est encore mieux pour contrecar-

rer les tricheurs (faible sécurité me direz-vous mais plus importante que de simples fichiers tout de même).

L'organisation des scores dans la base de registre est simple. Les infos sont stockées dans le chemin « `HKEY_LOCAL_MACHINE\SOFTWARE\NetBubbles` » qui contient les 2 clés `Params` et `Scores`. `Params` sert, vous l'avez certainement compris, à stocker les paramètres. `Scores` stocke les scores dans 3 variables, `Score3`, `Score4` et `Score5` sous la forme d'une chaîne de caractères, formatée.



Le format est simple : Chaque score est composé du Nom, du numéro de série de la partie et du score. Chaque élément est séparé par le code ascii 1. Chaque score est séparé du précédent par le caractère ascii 2.



Par défaut, le Compact Framework 1 ne sait pas manipuler la base de registre. Là encore, la librairie `CodePPC` va nous aider. Elle propose des méthodes pour gérer les données de type chaîne dans la base de registre. Ces méthodes fonctionnent de la même manière que celles du Framework. Les méthodes `GetSetting` et `SaveSetting` permettent donc de lire et écrire dans la base de registre une chaîne placée sous la clé « `HKEY_LOCAL_MACHINE\SOFTWARE\NomApplication\Clé` » où `NomApplication` et `Clé` sont passés en paramètre.

C'est la méthode `LireScores` du module `MainModule` qui effectue la transformation des infos de la base de registre vers les objets `ClassScore` de l'application. Cette méthode est appelée dans les premiers instants de `Main`, le point d'entrée de l'application. Chaque score étant stocké dans la collection correspondant à son nombre de couleur. Voici le code de `LireScores` :

```
Private Sub LireScores(ByVal wRegKey As String, _
    ByRef wLesScores() As ClassScore)

    Dim wBuffer As String
    Dim wTabScores() As String
    Dim wTabScore() As String
    Dim i As Integer
    Dim c As Integer

    REM Lecture des scores depuis la base de
```

```
registre
    REM Le séparateur de scores est le code
    ASCII 2
    REM Le séparateur de champs dans un
    score est le code ASCII 1

    wBuffer =
    CodePPC.General.Tools.GetSetting("NetBubbles", _
        "Scores", wRegKey)

    wTabScores = Strings.Split(wBuffer,
    Chr(2))

    If wTabScores.Length <= 10 Then
        c = wTabScores.Length - 1
    Else
        c = 4
    End If
    ReDim wLesScores(c)

    REM Pour chaque score, création de l'objet score
    REM et ajout au tableau
    For i = 0 To c
        wTabScore =
        Strings.Split(wTabScores(i), Chr(1))
        If wTabScore.Length = 3 Then
            wLesScores(i) = New
            ClassScore(wTabScore(0), _
                wTabScore(1), wTabScore(2))
        End If
    Next

    REM On trie les scores (le plus grand en
    premier)
    Array.Sort(wLesScores)

    End Sub
```

Voici le code qui appelle cette méthode pour la lecture des 3 groupes de scores :

```
REM Lecture des scores 3,4 et 5 couleurs
LireScores("Couleurs3", LesScores3)
LireScores("Couleurs4", LesScores4)
LireScores("Couleurs5", LesScores5)
```

`LesScore3`, `LesScores4` et `LesScores5` étant des tableaux d'objets `ClassScore`.

Comment adapter ce programme au Smartphone ?

Développer en .NET pour Smartphone nécessite d'avoir installé le kit de développement

pour Smartphone 2003. Vous pouvez télécharger ce kit depuis le site de Microsoft ou encore, profiter de l'opération CodePPC/Microsoft qui vous permet de recevoir chez vous tous les kits de développement Pocket PC et Smartphone contre l'envoi d'une enveloppe timbrée à votre adresse (voir les modalités à cette adresse : <http://www.codeppc.com/res-sources/kitsdev.htm>)

Vous trouverez les sources de NetBubbles pour Smartphone sur le site www.programmez.com accompagnant cet article. Les modalités pour installer ces sources sont exactement les mêmes que celles pour Pocket PC vues dans la première partie de cet article (voir le fichier *LisezMoi.txt*). Vous devez juste les placer dans des répertoires différents, car les projets portent le même nom. Personnellement, je place mes projets Pocket PC dans un dossier « PPC » et mes projets Smartphone dans un dossier « SMP », ceci évite tout problème.

Ce qui va changer avec la version Smartphone est assez mince. Tout d'abord, les billes devront être plus petites pour que le même plateau de jeu entre dans l'écran. Ensuite, il faut remplacer le stylet par une sélection clavier, ce qui oblige à matérialiser le curseur de sélection sur le plateau et à réagir aux événements liés aux touches de direction et de validation. Il faut aussi créer un nouveau projet Smart Device ciblant Smartphone et non Pocket PC. 90 % du code est réutilisable, le plus gros du travail étant de recréer les fenêtres et les menus au format Smartphone.



NetBubbles sur Smartphone

Voici le code de la méthode DrawPositionOnly de ClassGame qui pour Smartphone affiche le curseur de sélection :

```
Public Sub DrawPositionOnly(ByVal
wFormGraphics As Graphics)

    REM On trace le bitmap du jeu
    wFormGraphics.DrawImage(_Bitmap,
Me.Left, Me.Top)

    REM On ajoute le rectangle de position
    Dim wBubble As ClassBubble =
_Matrix[_Position.X, _Position.Y]
    Dim x As Integer = wBubble.Position.Left
+ Me.Left - 1
    Dim y As Integer = wBubble.Position.Top
+ Me.Top - 1
    wFormGraphics.DrawRectangle( _
New Pen(System.Drawing.Color.Coral), x, y,
15, 15)
    wFormGraphics.DrawRectangle( _
New Pen(System.Drawing.Color.Coral), x + 1,
y + 1, 13, 13)

End Sub
```

Il suffit ensuite de réagir à l'événement KeyDown de FormMain pour permettre à ce « curseur » de bouger et de pouvoir sélectionner une bille :

```
Private Sub FormMain_KeyDown(ByVal sender As Object, _
ByVal e As
System.Windows.Forms.KeyEventArgs)
Handles MyBase.KeyDown

    If TheGame.Is Nothing Then Exit Sub

    Dim g As Graphics = Me.CreateGraphics

    Select Case e.KeyCode

        Case 39

            TheGame.Deplacement(ClassGame.DeplacementsEnum.Droite, g)

        Case 37

            TheGame.Deplacement(ClassGame.DeplacementsEnum.Gauche, g)

        Case 40
```

```
TheGame.Deplacement(ClassGame.DeplacementsEnum.Bas, g)
```

Case 38

```
TheGame.Deplacement(ClassGame.DeplacementsEnum.Haut, g)
```

Case 13

```
If TheGame.ActiveSelection(g) Then
TimerGameOver.Enabled = True
```

End Select

g.Dispose()

End Sub

La méthode Deplacement de ClassGame s'occupe de déplacer correctement le curseur suivant la touche tapée par le joueur alors que ActiveSelection lance le traitement lié au choix d'une bille.

Dans le projet Pocket PC, c'est l'événement MouseDown de FormMain qui est le noyau de l'application. Dans la version Smartphone, c'est l'événement KeyDown qui devient le centre de toutes les opérations. L'idéal, pour le futur, sera de créer une version où les deux modes cohabitent afin de permettre aux joueurs Pocket PC ayant un clavier de l'utiliser pour jouer.

Conclusion

Développer un jeu avec le Compact Framework est assez simple, car il permet un accès simple et efficace au GDI. Il lui manque encore des fonctions, mais des bibliothèques, comme celle fournie gratuitement par CodePPC, permettent de combler très simplement ces lacunes qui ont d'ailleurs en partie disparu dans le Compact Framework 2.

■ Stéphane Sibué



webmaster@codeppc.com

Responsable développement
chez Antéïs (www.anteis.fr).

Microsoft MVP Windows Mobile
Devices Webmaster / Fondateur de CodePPC
(www.codeppc.com)



Abonnement "éco" 1 an

soit
l'équivalent
de

3 Numéros GRATUITS



économisez 30%

Tarif normal :
65 €, prix de vente au numéro

Offre Eco **45€** seulement !
(Tarif France Métropolitaine)

Abonnement 1 an + Clé USB 128 Mo



économisez 38%

NOUVELLE VERSION

Ce stick mémoire haut de gamme est multi-standard (Windows, Linux, Mac OS) à transfert ultra rapide et plug & play.
Nouvelle version : inclut en supplément le logiciel commercial Travel It Easy, (pour XP), suite d'utilitaires pour utilisation nomade (sécurité, sauvegarde au format Outlook Express etc...)

Tarif normal :
11 numéros à 5,95 € = 65,45 €
+ Clé USB 128 avec Travel It Easy + Frais d'envoi = 15,00 € (valeur environ)

Total : ~~80,45 €~~

Offre **49,50€** seulement !
(Offre limitée à la France Métropolitaine)



Abonnez-vous et recevez ce magnifique Couteau suisse - clé USB 128 Mo haut de gamme !

65 € seulement !
Pour le prix normal du magazine (prix au numéro)
Offre limitée à la France Métropolitaine

OUI, je m'abonne ! ou abonnement en ligne : www.programmez.com

Gratuit

Je peux dès aujourd'hui obtenir **gratuitement ma licence Démo Charge**, (version co-promoted) sur :
www.kapitec.com/Offres/Programmez/DC2006.html

- ☐ **OFFRE CLÉ USB + 1 an (11 numéros)** au prix de 49,50 € TTC *.
- ☐ **OFFRE COUTEAU SUISSSE-CLÉ USB + 1 an (11 numéros)** au prix de 65 € TTC *.

* Offres limitées à la France métropolitaine. Envoi sous délai moyen de 4 semaines.

- ☐ **ABONNEMENT SEUL 1 an (11 numéros)** au prix de 45 € TTC. Tarif France métropolitaine.

Tarifs hors France métropolitaine : CEE et Suisse : 51,83 € - Algérie, Maroc, Tunisie : 55,95 € - Canada : 64,33 € - Tom : 79,61 € - Dom : 62,84 € - Autres : nous consulter

☐ M. ☐ Mme ☐ Mlle Entreprise : Fonction :

Nom : Prénom :

Adresse :

Code postal : Ville :

Tél : E-mail :

- ☐ Je joins mon règlement par chèque à l'ordre de Programmez !
- ☐ Je souhaite régler à réception de facture

A remplir et retourner sous enveloppe affranchie à :
Programmez ! - Service Abonnements - 22 rue René Boulanger - 75010 Paris.
abonnements.programmez@groupe-gli.com

**Offre limitée,
valable jusqu'au
1^{er} mars 2006**

Le renvoi du présent bulletin implique pour le souscripteur l'acceptation pleine et entière de toutes les conditions de vente de cette offre.

Conformément à la loi Informatique et Libertés du 05/01/78, vous disposez d'un droit d'accès et de rectification aux données vous concernant.

Par notre intermédiaire, vous pouvez être amené à recevoir des propositions d'autres sociétés ou associations.

Si vous ne le souhaitez pas, il vous suffit de nous écrire en nous précisant toutes vos coordonnées.

PROG 83

Programmez! LE MAGAZINE DU DÉVELOPPEMENT

Maîtriser MySQL 5

- Auteur : collectif
- Editeur : O'Reilly
- Prix : 40 €

Mysql 5 est disponible depuis quelques mois et les ouvrages autour de ce SGBD fleurissent. O'Reilly propose un livre sur le



sujet, mais avec sa qualité habituelle. Il s'agit de maîtriser MySQL 5 du point de vue développeur et du code. Il aborde le côté client et serveur avec à chaque fois, des études de cas et des recommandations. Nous

avons particulièrement apprécié les parties sur les procédures stockées et les interfaces C, Java, C#. On peut regretter l'absence d'une section administration / déploiement pour que le livre soit complet, mais dans sa configuration actuelle, il fait partie des références sur MySQL.

Le langage Java 3^e édition

- Auteur : Irène Charon
- Editeur : Lavoisier
- Prix : 45 €

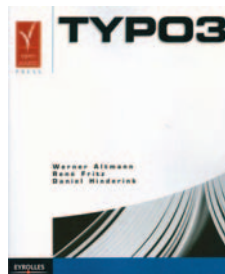


Mis à jour pour tenir compte de Java 5 et des dernières avancées du langage, l'ouvrage se veut une introduction à la programmation Java en abordant les concepts de base et la pratique du langage, via les différents éléments

(tableaux, classes, paquetages, interfaces, exceptions, etc.). Un chapitre entier (bizarrement estampillé 12) aborde la JDK 5. Un ouvrage idéal pour ceux qui ne connaissent pas Java et qui veulent s'y mettre en partant sur de bonnes bases. Une ouverture sur J2EE et J2ME aurait été appréciable.

TYPO3

- Auteur : collectif
- Editeur : Eyrolles
- Prix : 45 €



Comment créer un site collaboratif et gérer du contenu simplement et si possible rapidement ? En open source ou en commercial, les solutions ne manquent pas. Sans faire de bruit, TYPO3 est beaucoup utilisé en entreprises et plus de 1 000 composants existent pour l'étendre, il est écrit en PHP. Les auteurs participent directement ou indirectement au projet. L'ouvrage se segmente très simplement : installation, partie pour les utilisateurs, pour les administrateurs et les développeurs. La partie développement occupe plus de la moitié du livre. TYPO3 possède son propre langage, TYPOSCRIPT, longuement abordé dans ce chapitre. Pour tout ce qui concerne l'extensibilité, le framework TYPO3 bénéficie d'un traitement très détaillé.

Blogs pour les pros

- Auteur : collectif
- Editeur : Dunod
- Prix : 25 €

Qu'est-ce qu'un bon blog pour un professionnel ? Si on sait ce qu'est un weblog, comment un professionnel doit-il aborder ce nouveau mode de communication ? Les auteurs tentent de répondre à cette question. Ils reviennent tout d'abord sur le blog, sa définition, son fonctionnement de base et, bien entendu, son phénomène. Puis ils démontent les objections classiques pour ne pas faire un

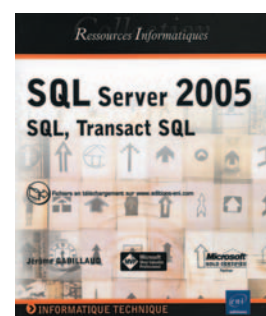
blog. Les contre arguments sont intéressants, mais pas toujours totalement convaincants. Le blog, pour réussir, doit comporter quelques principes éthiques de base : être transparent et honnête. Le blog du pro peut servir à échanger, partager son expérience, ses compétences. En quelque sorte, un CV concret.



Encore faut-il sortir du lot. Ce qui n'est pas forcément évident. Une des originalités du livre est de proposer des interviews de responsables et patrons d'entreprises, souvent connus, qui donnent leur avis sur la question. Dommage que le futur des blogs ne soit pas plus développé et mis en valeur...

SQL Server 2005 : SQL, Transact SQL

- Auteur : Jérôme Gabillaud
- Editeur : Eni éditions
- Prix : 27,14 €



Vous avez installé SQL Server (version complète ou express) et vous souhaitez commencer à utiliser SQL ou Transact SQL dans votre base et découvrir les principales fonctions du nouveau SGBD de Microsoft ? L'auteur vous offre un tour d'horizon de tout cela, de l'installation à la notification. La partie consacrée à Transact SQL est la plus complète. Une bonne introduction à SQL Server 2005.

Démarrer avec Linux, les ouvrages

- **Essayer Linux (O'Reilly, 34 €)** : livré avec Mandrake Move, l'auteur fait un tour d'horizon (assez sommaire) des possibilités de Linux et des applications livrées avec le système. Orienté débutant. Le prix est un peu élevé.
- **Débuter sous Linux avec Mandriva (Eyrolles, 32 €)** : livré avec Mandriva 2006. Une bonne introduction à l'usage au quotidien de Linux et des applications. Le shell n'est pas oublié ni la sécurité. Bon point : un pense bête détachable.
- **Ubuntu (Eyrolles, 25 €)** : après une déferlante Knoppix en 2004 - 2005, Ubuntu devient un sujet à la mode. Ici, les auteurs abordent le système, son utilisation, ses paramètres. À noter que CampusPress propose lui aussi un ouvrage sur Ubuntu (Ubuntu Linux, 36,90 €)

Choisissez le Master M2IRT : le diplôme des futurs ingénieurs en informatique

- ➔ Titre certifié de niveau 1 homologué par l'état
- ➔ École publique d'informatique (partenaire du Syntec)
- ➔ Plus qu'un diplôme, une expérience professionnelle reconnue

vous êtes déjà dans le futur

www.itin.fr
m2irt@itin.fr

Tél. : 01 34 20 63 63

AIRE 01 30 30 16 02



CCI

Chambre de commerce et d'industrie
Versailles Val-d'Oise / Yvelines

ENSEIGNEMENT FORMATION

Ici commence l'avenir



Microsoft
Developer Days
2006

Windows Vista, Office « 12 », WinFX...
Du 7 mars au 4 avril, venez découvrir
comment développer les applications
de nouvelle génération. Inscrivez-vous sur
www.microsoft.com/france/devdays

© 2006 Microsoft Corporation. Tous droits réservés. Microsoft, Visual Studio, le logo Visual Studio et « Votre potentiel, notre passion. » sont soit des marques, soit des marques déposées de Microsoft Corporation aux États-Unis d'Amérique et/ou dans d'autres pays.

Nouveau Visual Studio 2005. La différence saute aux yeux.

Vous voyez la différence ? Quand vous aurez commencé à coder, vous vous en rendrez compte immédiatement. Le nouveau Visual Studio® 2005 vous propose plus de 400 nouvelles fonctionnalités, des contrôles Web et Windows® et des insertions de codes semi-automatiques qui réduisent les tâches ennuyeuses et répétitives. Vous pouvez ainsi vous concentrer sur la création de nouvelles fonctionnalités. Pour découvrir le nouveau Visual Studio 2005, consultez www.microsoft.com/france/vs2005

Microsoft
Visual Studio 2005