

Programmez!

Mensuel • Mars 2004 • N°62 • 5,95 €

OPEN SOURCE.COM

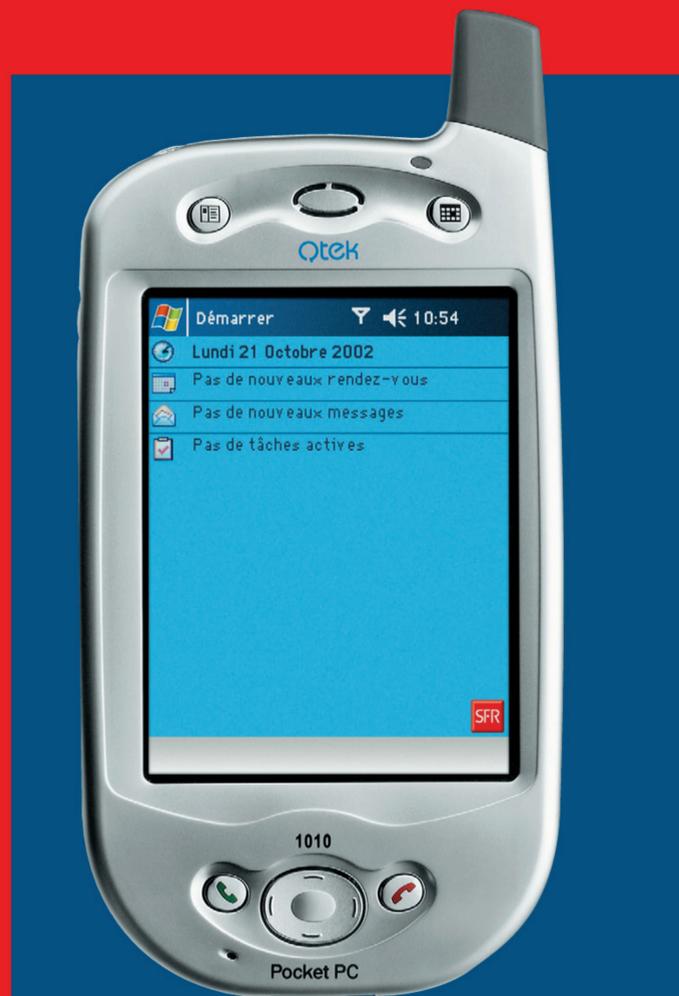
LINUX ET OPEN SOURCE EN ENTREPRISE

DÉVELOPPER, MAINTENIR, MIGRER

PANORAMA DES LOGICIELS

L'ÉTAT PROPOSE

UN OPEN SOURCE "À LA FRANÇAISE"



DÉVELOPPER POUR POCKET PC ET SMARTPHONE

EMPLOI

LES SSII RECRUTENT À NOUVEAU, SURTOUT EN RÉGIONS

PROGRAMMATION

MULTITHREAD en JAVA



PRATIQUE

PYTHON et ZOPE :

Mise en œuvre XML-RPC

TECHNOLOGIE

Infopath :

Les formulaires dynamiques de l'avenir ?

LUDIQUE

Linux sur une PLAYSTATION 2

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

M 04319 - 62 - F: 5,95 €



Virus d'hiver



MYDOOM, DUMARU, BAGLE. L'année 2004 s'annonce encore plus virale que 2003, baptisée l'année des virus. Ce sera probablement aussi l'année où la méthode de détection heuristique sera remise en question.

MYDOOM a été programmé pour lancer une attaque en déni de service sur les sites de SCO et de Microsoft respectivement ; attaque qui a duré entre le 1er et le 14 février. Microsoft propose 250 000 \$ à toute personne offrant des informations qui permettront la capture de l'auteur du ver. Ceci relève du budget de 5 millions de dollars consacré par Microsoft pour la capture de pirates et auteurs de virus.

MYDOOM

A eux seuls, MYDOOM et ses variantes ont infecté des millions d'ordinateurs à travers le monde. Techniquement, rien de particulièrement innovateur : MYDOOM.A et MYDOOM.B (alias NOVARG) sont des vers qui arrivent par email avec une pièce jointe. L'adresse email d'expéditeur peut être une adresse usurpée, ce qui cause d'ailleurs un problème supplémentaire de trafic, les filtres antivirus renvoyant les alertes d'e-mails infectés aux expéditeurs qui ne sont évidemment pas les vrais expéditeurs. La pièce jointe attachée contaminée est nommée, selon les cas : *body*, *data*, *document*, *readme*, *doc*, *text*, *file*, *test* ou *message*. Cette pièce jointe est en réalité un fichier exécutable qui peut prendre l'apparence d'un fichier compressé.zip, ou d'un fichier.htm, .doc ou .txt alors qu'en réalité il s'agit d'un exécutable. Le texte du message de l'email est, selon les cas :

- *Mail transaction failed. Partial message is available.*
- *The message contains Unicode characters and has been sent as a*

binary attachment.

- *The message cannot be represented in 7-bit ASCII encoding and has been sent as a binary attachment.*

Ceci, afin de faire croire au destinataire qu'il s'agit d'une erreur d'envoi, pour l'inciter à ouvrir la pièce attachée.

Le ver s'ajoute au démarrage de Windows en créant un fichier nommé *shimgapi.dll* et *taskmon.exe* et en modifiant la base de registre de manière à être lancé au démarrage de Windows. Au démarrage de la machine, le ver se comporte en tant que ser-

essaiera de s'envoyer à d'autres personnes ayant des noms courants dans le même nom de domaine (exemple : *bob@domaine.com*, *alice@domaine.com* etc). Enfin, le ver détourne les résolutions DNS de sites antivirus sur les machines attaquées, afin d'empêcher l'utilisateur de se connecter au site de son antivirus ou d'effectuer les mises à jour de signatures.

Il se copie dans le répertoire des téléchargements de KAZAA sous les noms *winamp5*, *icq2004-final*, *activation_crack*, *strip-girl-2.obd*, *com_patches*, *rootkitXP*, *office_crack* ou *nuke2004*.

MYDOOM.C

Cette variante exploite les portes dérobées laissées par MYDOOM.A et B. Ainsi, MYDOOM.C remplace ses prédécesseurs à distance. Ainsi, les machines infectées par MYDOOM.A et B peuvent se

jour à distance, de la même manière que Windows Update ou que n'importe quel logiciel légitime y a été mentionnée. C'est donc la première fois que ce genre de mutation complète d'un ver à distance est utilisé.

DUMARU

un e-mail composé d'erreurs

DUMARU est un autre ver qui a la particularité de s'envoyer par des e-mails corrompus, qui ne peuvent être lus correctement que par certains logiciels de messagerie. Le but : contourner les protections et filtres de messageries.

Et les protections ?

Avec toutes ces récentes attaques, qui sont pourtant pétrées par des vers assez simples, une question se pose de plus en plus : pourquoi la combinaison des scanners heuristiques et à signatures, qui sont censés détecter aussi les virus inconnus, n'ont-ils pas protégé les machines cet hiver, tout comme l'été dernier ? Pourtant d'un point de vue comportemental, le ver peut facilement être détecté par les actions qu'il commet :

- Arrivée par pièce jointe exécutable, qui de plus contient un nom truqué.
- Ajoute au démarrage deux fichiers : *Taskmon.exe* s'ajoute par la clé registre : `HKCU\Software\Microsoft\Windows\CurrentVersion\Run` et : `HKLM\Software\Microsoft\Windows\CurrentVersion\Run`
- Le fichier *Shimgapi.dll* s'inscrit au démarrage par la clé registre : `HCR\CLSID\{E6FB5E20-DE35-11CF-9C87-00AA005127ED}\InProc Server32`
- Envoi des e-mails en passant par le port 25 (SMTP).
- Ouvre des ports pour gagner accès à la machine.

■ Eyal Dotan

Directeur Recherche Développement, Tegam International



veur TCP/IP en ouvrant les ports TCP 3127 à 3198 et rend la machine accessible à distance via ces ports. MYDOOM.A recherche ensuite des adresses emails dans les fichiers *htm*, *asp*, *php*, *adb*, *dbx*, *sht*, *tbb*, *pl*, *wab*, *txt*, et utilise son propre moteur SMTP pour s'envoyer à ces adresses. Il

retrouver infectés par MYDOOM.C sans même ouvrir des e-mails. Il est intéressant de noter que cette technique a été prévue dans le rapport préventif de recherche nommé "GoodLuck" et dont Programmez a été parmi les premiers à parler. La possibilité qu'un programme malveillant se mette à



Jacques SAURET, ADAE : Un Open source à la française

L'ADAE, Agence pour le Développement de l'Administration Electronique vient de fêter son 1er anniversaire. L'organisme interministériel a présenté mi-février le "Projet ADELE", un plan 2004-2007, qui propose 140 actions et 300 nouveaux services. Doté d'un budget de 1,8 milliard d'euros, il a comme objectif un gain de productivité de 5 à 10 milliards d'euros par an, au seul niveau emblématique des achats publics.



Avec l'objectif de "Simplifier la vie des Français", ADELE veut rendre les services accessibles pour l'utilisateur : le guichet d'une administration devrait ressembler à une agence de voyage, qui regroupe toutes les informations pratiques nécessaires. Entre autres projets : Copernic (paiement des impôts en ligne) et la Carte d'identité Electronique. www.adae.pm.gouv.fr

Jacques Sauret (directeur de l'Adae) a commenté son plan, à l'occasion de conférences de presse et d'un débat lors de Solutions Linux. Sans langue de bois, il regrette l'absence de

concurrence sur le poste de travail. Concernant le coût des licences il a annoncé un bras de fer avec Microsoft : "Aujourd'hui, il ne faut pas se faire d'illusions, nous sommes "captifs", et cela va durer encore deux à quatre ans, au moins. Nous allons entamer des discussions avec Microsoft pour obtenir un prix unique pour l'ensemble des administrations, de façon à obtenir des économies d'échelle".

Jacques SAURET maintient le cap proposé par ses prédécesseurs : ouvrir l'administration à l'Open Source, mais pas les yeux fermés. Aucun objectif pré-défini pour

faire migrer les utilisateurs : "5 à 20% ?", il faudra faire l'expérience.

"L'argent public ne sert pas à subventionner SSII et éditeurs"

Concernant l'achat de prestations logicielles, un round de discussion avec le Syntec est prévisible :

"L'argent public ne sert pas à subventionner SSII et éditeurs de logiciels. Les SSII n'ont pas le choix : elles doivent évoluer. Quand l'argent public a été dépensé pour un service, l'administration devrait disposer des droits d'usage pour les autres ser-

vices de l'état. Des SSII développent pour une administration et vont le revendre à d'autres à un bon prix.

On ne devrait acheter que de l'innovation, pas de l'itération. Si les briques de base sont mutualisées, les SSII ne revendront pas plusieurs fois leurs développements. Les SSII doivent changer de modèle économique : il y aura toujours autant, voire plus d'argent dépensé, mais en innovation, en intégration, en formation".

■ Jean KAMINSKY

Le gouvernement français propose une troisième voie, entre le logiciel libre et propriétaire : la brique applicative mutualisée...

PROGRAMMEZ : Y a-t-il une volonté "politique" concrète de soutenir le Logiciel libre ?

Jacques Sauret : Au niveau du gouvernement, la position est très claire : on soutient le logiciel libre, on va faire en sorte que ça devienne de plus en plus mature, que ça devienne une alternative extrêmement forte. Mais au moment de l'appel d'offre, c'est ouvert, la concurrence joue, on ne veut pas imposer impérativement le libre, ni l'interdire. Le débat est souvent mal posé. Il est temps de dépasser l'opposition entre libre et propriétaire. La vraie solution, c'est la normalisation. Après les réseaux et les flux de données (XML), il faut travailler sur des

briques applicatives réutilisables, que les industriels pourront revendre dans des solutions intégrées. Si les codes sources, les briques alternatives sont en libre, l'"acheteur public" n'achètera plus des licences, mais des fonctions d'intégration, de conduite du changement, d'implémentation locale. C'est le sens de l'histoire. Aujourd'hui, le libre ne peut pas décoller, car on est sur des licences d'origine nord américaine, GPL notamment, qui interdisent, si on participe à un système commun, de réutiliser dans ses propres produits et commercialiser. Aucun industriel n'est intéressé à rentrer dans ce système, sauf pour nuire aux autres. Quand

tel grand industriel américain veut embêter tel autre, il met 1 milliard de dollars sur la table... ! L'idée c'est de faire en sorte que les SSII, petites et grandes, françaises voire européennes, retrouvent un modèle économique, en échappant au dilemme actuel : utiliser du libre sans avoir le droit de l'utiliser (commerciallement), ou des licences propriétaires, ou encore développer "from scratch". Nous proposons un nouveau choix, une logique "collective" qui permettrait de faire émerger ce qui est mon souhait - une industrie des éditeurs de logiciels nationale, sur des souches libres.

(Propos recueillis par JK)

Programmation des Pocket PC et Smartphone

La mobilité gagne du terrain chaque jour. Les PDA font désormais partie intégrante du système d'information de l'entreprise. Les besoins en développement d'applications sont par conséquent devenus importants. La programmation s'effectue avec les outils standard, sans poser de problèmes particuliers.

En effet, la spécificité du Pocket PC est surtout fonctionnelle, techniquement il n'y a pratiquement plus de différence entre les applications serveur et celles développées pour cet appareil. Ce dossier sera plus particulièrement consacré aux outils Microsoft et à la programmation en Visual Basic, C++ et C#. D'autres langages, dont Java en particulier sont bien entendu utilisables.

Les organiseurs, encore appelés ordinateurs de poche ou PDA n'étaient au départ que des Filofax électroniques. Depuis, leurs fonctions se sont largement étendues, et les Pocket PC, par exemple, ont la même puissance que celle des PC de bureau d'il y a sept ou huit ans. Même si les fonctions PIM (Personal Information Management) que sont le Calendrier, le Carnet d'adresses, les Notes restent importantes, les applications bureautiques fournies en standard en font un véritable ordinateur personnel, léger et facilement transportable. À partir de là, il n'y a plus de raison de ne pas y ajouter les applications métier de l'entreprise. Le Pocket PC est en effet capable de gérer des bases de données



relationnelles, de se connecter de manière sécurisée au système d'information de l'entreprise, directement sur le réseau, ou par accès distant. La fonction qui lui manque le plus est le téléphone. Ce vide est comblé par deux autres appareils, le Smartphone et le Pocket PC Phone Edition. Nous nous limiterons ici à la programmation du Smartphone tournant sous Windows CE, c'est-à-dire le SPV d'Orange. Pour faire simple, on peut dire que le Smartphone est un téléphone avec des fonctions de Pocket PC, tandis que le Phone Edition est un Pocket PC auquel on a ajouté la téléphonie. Ces trois appareils sont mus par le même système d'exploitation qui est Windows CE. Curieusement, Microsoft n'a jamais donné la signification exacte de "CE". Certains ont pensé à Consumer Electronics. Il semble plus probable que ce soit en fait Compact Edition.

Les multiples versions de Windows CE

Il est vite apparu que Windows CE pouvait être traité de différentes manières, en fonction des besoins des utilisateurs, de leurs goûts, des demandes du marché et enfin des évolutions technologiques. Par exemple, l'appareil lui-même peut être doté, ou non d'un clavier pour la saisie des données et les écrans peuvent être monochromes, ou couleurs. Windows CE s'est donc décliné en tout un ensemble de systèmes qui paraissent quelque peu différents, mais qui sont basés sur la même architecture. Ce système tourne sur des matériels aussi divers que les Pocket PC, les téléphones

"intelligents" Smartphone, les Handheld PC (petits appareils munis d'un clavier) les caisses enregistreuses, ou encore les terminaux type Client Léger. Il est devenu bien difficile de s'y retrouver, d'autant plus que, pour ajouter à la confusion, Microsoft mélange le nom des appareils avec celui du système d'exploitation. Le Pocket PC en est le cas typique. On ne sait plus très bien si ce terme désigne l'appareil lui-même, ou la version Windows CE qui tourne dessus. La dernière version s'est d'abord appelée Windows CE 4.2, puis Windows CE .NET et enfin Windows Mobile 2003. Mais, par commodité, beaucoup le nomment également Pocket PC 2003 ! Le Pocket PC a connu trois versions différentes, le Pocket PC sorti en avril 1999, le Pocket PC 2002 sorti en septembre 2001 et enfin le Pocket PC 2003 présenté en juin 2003. Comme nous le verrons plus loin, leur programmation nécessite des outils différents.

Le compagnon du PC

Le Pocket PC est limité physiquement. Le système d'exploitation, ainsi que les programmes de bureautique livrés en standard sont stockés dans une mémoire flash. Il est donc possible par flashage, d'effectuer des mises à jour ou même de changer complètement de système. Compaq a ainsi fourni une version spécifique de Linux pour ses Pocket PC. Les données et les programmes supplémentaires sont stockés dans une mémoire dynamique qui a besoin d'être alimentée en permanence. En cas de panne, toutes les données sont perdues. Cet appareil peut accepter de nombreuses extensions matérielles telles que des cartes-mémoires pour le stockage de grandes masses de données (des cartes de 4 Go sont d'ores et déjà disponibles sur le marché), des cartes d'accès aux réseaux Ethernet, Wi-Fi ou encore Bluetooth. Ces deux derniers standards sont d'ailleurs souvent directement intégrés dans les modèles haut de gamme. De nombreux constructeurs offrent également des cartes plus spécialisées, comme des lecteurs codes à barre, des appareils photographiques, ou des GPS.

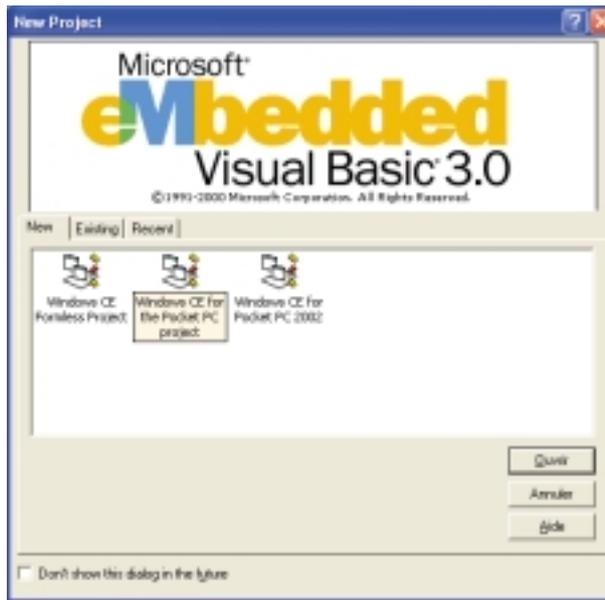


Figure 1 : Chaque plate-forme Pocket PC nécessite un SDK particulier. Après les avoir installés, il suffit de choisir l'appareil cible lors de la création du projet.

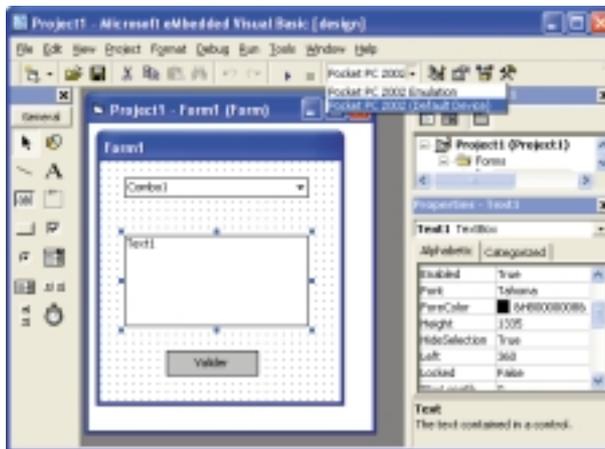


Figure 2 : L'interface de programmation de eMbedded Visual Basic ne déroutera pas les développeurs habitués à ce langage. Remarquez la liste déroulante qui permet de sélectionner l'appareil sur lequel l'application doit être testée : une machine réelle, ou bien l'émulateur

Un Pocket PC ne s'utilise jamais seul. Il ne peut développer toute sa puissance que s'il est relié à un serveur (ordinateur de bureau), avec lequel il peut échanger des informations. Les échanges ne se limitent pas à de simples opérations de transfert, mais à une opération complexe de synchronisation. Ainsi, si par exemple de nouveaux contacts ont été saisis sur l'appareil et que parallèlement, d'autres contacts ont été enregistrés sur l'ordinateur de bureau, l'opération de synchronisation modifiera le contenu des deux appareils de manière à ce que toutes les modifications soient prises en compte et que les listes de Contacts soient identiques. Le programme maître d'œuvre de

ces opérations se nomme ActiveSync. C'est lui qui est utilisé pour l'installation de nouvelles applications sur le Pocket PC. De très nombreux types de fichiers ne peuvent pas être transférés sans modifications. Il s'agit en premier lieu des fichiers Office, c'est-à-dire des fichiers Word et Excel. Ces applications bureautiques sont fort heureusement compatibles entre elles, mais elles n'utilisent pas les mêmes formats de fichiers. La conversion fera donc perdre certaines informations de formatage pour les fichiers texte et fera disparaître les graphiques dans le cas de fichiers Excel. La conversion doit également être effectuée pour une autre raison : il s'agit de réduire le volume de fichiers. Sur le PC, Microsoft Office ne regarde pas à la dépense, et fait une consommation effrénée de la place en mémoire et du disque dur. Sur le Pocket PC, la mémoire est comptée, il faut donc faire preuve de beaucoup plus d'ascétisme.

Les outils de programmation proposés par Microsoft

Microsoft a simplifié le travail des développeurs d'applications pour Windows CE, en proposant un environnement de développement complet baptisé eVT (eMbedded Visual Tools 3.0). Cet environnement est composé de deux outils indépendants qui permettent de développer des applications soit en VB (eMbedded Visual Basic 3.0), soit en C++ (eMbedded Visual C++ 3.0). Pour travailler, il suffit d'ajouter les SDK nécessaires. Chaque plate-forme Windows CE a besoin d'un SDK spécifique contenant les API et les bibliothèques qui lui sont propres. Il existe un SDK pour le Pocket PC, un pour le Pocket PC 2002, un autre pour le Pocket PC 2003, et des SDK pour les différentes versions de Smartphone. Des SDK supplémentaires peuvent aussi être nécessaires, dans le cas d'applications particulières qui font appel à des matériels spécifiques connectés au Pocket PC, tels que des lecteurs de codes à barres, ou des GPS. Ces SDK sont alors fournis par le constructeur du matériel. Microsoft propose une autre version de eVT 3.0, baptisée eMbedded Visual Tools

3.0 – 2002 Edition. Il ne s'agit en fait que de la version standard, à laquelle les SDK Pocket PC 2002 et Smartphone 2002 ont été ajoutés. L'environnement eVT, ainsi que le SDK sont téléchargeables gratuitement, sur le site de Microsoft, mais ils n'existent malheureusement qu'en anglais. Le fichier "pèse" environ 320 Mo, aussi pour ceux qui ne disposent pas d'une liaison haut débit, il est possible de commander un CD moyennant environ \$12. La configuration minimale pour supporter eVT est un Pentium III à 150 MHz avec au moins 720 Mo d'espace disponible sur le disque dur et 32 Mo de mémoire vive. Le logiciel fonctionne sous Windows 98 Seconde Edition, Windows NT 4.0, Windows 2000 et Windows XP. L'outil

eVB 3.0 est très simple à utiliser, mais il est destiné à des applications très simples elles aussi. Contrairement à ce que son nom indique, eVB n'est pas réellement un outil de programmation en Visual Basic. Le langage utilisé est plutôt du VBScript. Celui-ci ne supporte que les données de type Variant (pas les User Defined Types) et ne contient pas tous les contrôles des versions desktop de Visual Basic. De plus, il ne peut pas être utilisé avec les nouvelles générations Windows Mobile 2003 de Pocket PC. Pour ces derniers, il faut passer à eMbedded Visual C++ 4.0. Il n'existe pas de version 4.0 pour eVB. Ce langage étant trop simpliste, il est conseillé de se tourner vers Visual Studio .NET pour les développements en VB. Cet environnement est beaucoup plus puissant, il permet d'utiliser à la fois les langages Visual Basic et C# et C++. En revanche, il n'est pas gratuit ! Les premières versions de Visual Studio .NET ne permettaient pas de développer directement pour le Pocket PC. Il fallait ajouter une SDE (Smart Device Extension). Celle-ci n'est plus disponible et il faut par conséquent passer directement à la version VS .NET 2003 (ou effectuer une mise à jour). Cette dernière version ne nécessite pas de SDK pour le Pocket PC, mais ne permet pas encore de développer pour le Smartphone. Il faut savoir enfin que les applications développées en Visual Basic ont besoin d'un runtime pour tourner sur le Pocket PC. Les applications développées en code managé (.NET) nécessitent l'installation du .NET Compact Framework sur les Pocket PC et les Pocket PC 2002. Ce Compact Framework est inclus dans les versions Windows Mobile 2003 (Pocket PC 2003). (Figure 1)

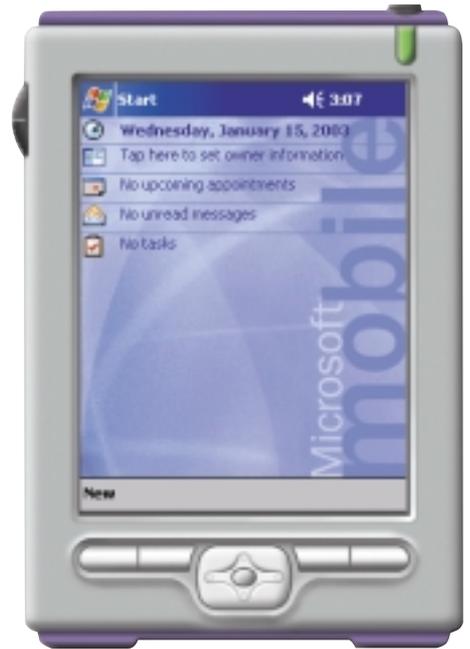


Figure 3 : L'émulateur a tout d'un "vrai" Pocket PC. Les clics de souris sur l'écran remplacent les clics du stylet

Le portage des applications

Le portage d'applications déjà écrites pour Windows ne pose pas de gros problèmes, mais prendra tout de même du temps. La majorité des contrôles VB par exemple pourra être transférée en copier-coller et les Propriétés seront préservées. Il faudra tout de même modifier leur taille et leur disposition pour les loger dans le tout petit écran de l'appareil. En revanche, certains contrôles n'existent pas pour le Pocket PC et là il faudra réécrire le code. Pour porter une application, il sera toujours préférable d'utiliser Visual Studio .NET que eVB. Comme nous l'avons vu, ce dernier est plus proche de VBScript que de Visual Basic et il sera souvent plus simple dans ce cas de réécrire entièrement l'application.

Tester les applications

Il n'est pas absolument obligatoire de posséder un Pocket PC ou un Smartphone pour commencer un développement. eVT et Visual Studio contiennent en effet des émulateurs logiciels de des appareils. Étant donné que ceux-ci s'exécutent sur le poste de travail qui est bien plus puissant que le Pocket PC, les applications tournent beaucoup plus vite, ce qui fait gagner un peu de temps en développement. De plus, ces émulateurs sont facilement personnalisables. Ils utilisent simplement une image bitmap de l'appareil sur laquelle un écran est placé en surimpression. Sous eVT par exemple, cette image se trouve dans le répertoire `\Windows CE Tools\wce300\Pocket PC 2002\emulation\pda_1.bmp`. Il est donc facile de remplacer le skin par une photo du modèle réel. En revanche, il faut savoir que si ces émulateurs peuvent être utilisés comme une aide au début, il n'est pas réaliste de s'en servir pour tester les applications. D'abord, parce toutes les fonctions des Pocket PC et Smartphone ne sont pas supportées et ensuite, parce que certaines de ces fonctions sont implémentées différemment sur les appareils réels. Au final, c'est donc avec eux qu'il faudra terminer les tests. Pour cela, il existe plusieurs possibilités de connexion. La première consiste à placer le PDA sur sa base et utiliser

Les points importants à respecter

- Sur ces appareils, la mémoire est une denrée rare. Il faut optimiser au maximum les applications.
- Quelle que soit la méthode de synchronisation des données choisie, il faut prendre en compte le fait que la bande passante sera faible (GPRS) et/ou pas toujours très fiable (modem RTC).
- L'Interface Homme Machine doit être simple et ergonomique.
- L'utilisateur attend un résultat immédiat à ses requêtes. L'application doit être réactive, aussi il est impératif d'optimiser les données. La plus grande attention est donc requise au niveau des jointures, des Order by ainsi que des clés.
- Il est également important de sécuriser les transmissions et d'optimiser les flux en compressant systématiquement les données. Il faut penser aussi que les Web Services sont un peu plus verbeux.

ActiveSync. Cela fonctionne bien, mais les transferts sont longs, ce qui est un peu pénalisant. Une méthode bien meilleure avec les Pocket PC consiste à les connecter au réseau local, à l'aide d'une carte CF-Ethernet par exemple. Le chargement des applications est alors très rapide et le débogage s'en trouve grandement facilité. Le grand luxe enfin, consiste à relier le Pocket PC au réseau par une liaison sans fil Wi-Fi (802.11b), ou Bluetooth. C'est de très loin la solution la plus pratique, puisque l'appareil peut être manipulé facilement. Pour finir, il est également possible d'utiliser des logiciels qui reproduisent l'écran du Pocket PC sur le poste de travail. Les applications du Pocket PC peuvent alors être lancées directement en cliquant sur cet écran. Parmi les logiciels utilisables, le plus simple est ActiveSync Remote Display. Ce programme fait partie d'un ensemble gratuit nommé Windows Mobile Developer Power Toys qui contient de nombreux outils intéressants pour le développeur, généralement en ligne de commande, pour lancer des applications sur le Pocket PC, ou transférer des fichiers. (Figure 2 et 3)

Développer des applications

Tout est maintenant en place pour développer notre première application. Sous Visual Studio .NET, passez par le menu Fichier/Nouveau/Projet, sélectionnez le langage voulu dans la liste de gauche (Visual Basic, ou Visual C#),

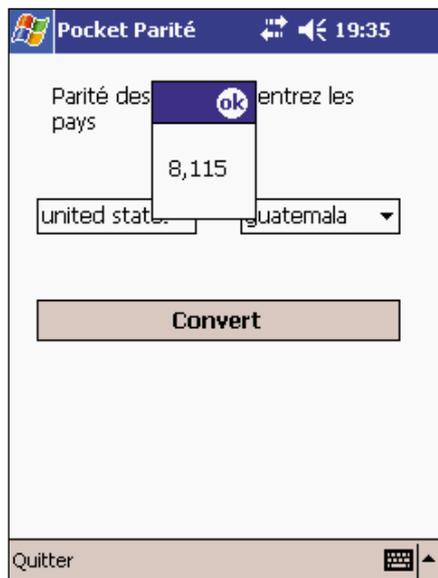


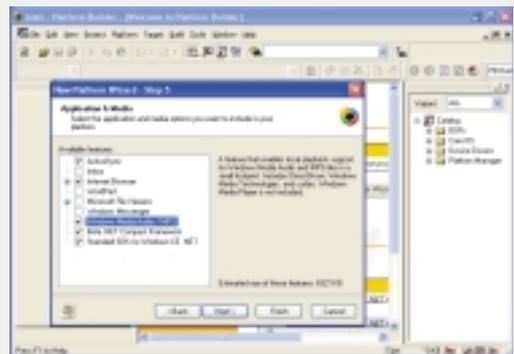
Figure 4 : L'application ci-dessus fait appel à un Web Service pour calculer les parités entre différentes monnaies. Le code tient en seulement quelques lignes

Désactiver les sécurités des Smartphone

Il n'est pas possible d'installer des logiciels non authentifiés par Orange sur un Smartphone. Le but de l'opérateur est de protéger la machine contre les virus et autres programmes malveillants, qui pourraient par exemple lancer des communications à l'insu du propriétaire. Si cette fonction protège les utilisateurs, elle pose de gros problèmes aux développeurs. La certification par Orange coûte environ 300 €, ce qui est peu pour un éditeur, mais important pour le développeur de sharewares ou de freewares. Pendant longtemps, la seule solution a consisté à utiliser un programme " pirate " nommé SmartPhoneROM. Ce n'est fort heureusement plus nécessaire. Orange propose à ceux qui en font la demande, de déverrouiller leur appareil à distance en se connectant sur le site développeurs (voir encadré sur les sites)

PlatformBuilder

PlatformBuilder n'est pas à proprement parler un outil de développement. En effet, il ne s'adresse pas aux développeurs, mais aux OEM désireux de réaliser leur propre plate-forme. Windows CE est en effet constitué d'un ensemble de modules indépendants. Avec PlatformBuilder les fabricants personnalisent leur système, en intégrant les modules dont ils ont besoin. PlatformBuilder permet ainsi de créer un système d'exploitation sur mesure. Lorsque la plate-forme a été créée, elle peut être exportée sous forme de SDK qui sera ensuite récupéré par eMbedded Visual Tools, pour créer des applications propres à cette nouvelle plate-forme. Comme pour la plupart des outils de développement destinés à Windows CE (mis à part Visual Studio) PlatformBuilder est téléchargeable gratuitement sur le site de Microsoft.



PlatformBuilder permet de construire son propre système d'exploitation Windows CE en choisissant les options à la carte

puis cliquez sur l'icône Application Smart Device. Choisissez ensuite la cible Pocket PC, puis sous Visual Basic par exemple, placez un contrôle TextBox dans lequel, sans surprise, vous placerez la propriété Text à Hello World. Windows CE utilise ses propres API, mais vous remarquerez que fort heureusement, le nom des fonctions, leur syntaxe ainsi que les propriétés des contrôles sont très proches de l'API Win32. Choisissez alors l'exécution sur l'émulateur, ou bien sur le Pocket PC réel. Si vous faites ce dernier choix, activez auparavant une connexion ActiveSync. Cela fait, lancez l'exécution. La compilation va s'effectuer, et le programme être automatiquement téléchargé sur la cible, puis lancé. Pour arrêter ce programme, cliquez sur la petite croix placée en haut à droite. Vous remarquerez qu'elle a été ajoutée automatiquement par l'outil de développe-

ment. Le code d'une application utilisant des Web Services est disponible sur le CD d'accompagnement. Voici comment procéder pour la mise en route. Recopiez le dossier WSDData dans le sous dossier wwwroot de Intetpub. Déclarez-le en tant que Web Service. Lancez ensuite Visual Studio et ouvrez le projet Datalib, situé bien sûr dans le dossier Datalib. Dans l'Explorateur de solutions effectuez un clic droit sur Solution 'Datalib', puis sélectionnez Ajouter/Projet existant. Sélectionnez le projet Pocketdata.csdproj situé dans le dossier Pocket Data. Enfin, dans le projet Pocketdata actualisez la référence sur le Web Service. Lancez la compilation. N'oubliez pas de donner le nom du serveur, c'est-à-dire celui qui contient le Web Service. Enfin, pour que cela fonctionne, connectez le Pocket PC par le réseau, ou bien utilisez le Loopback si le

Pocket PC est relié par sa station d'accueil. Le CD d'accompagnement contient également un projet encore plus simple à mettre en œuvre, puisqu'il utilise un Web Service disponible librement sur le site de XMethods à l'adresse www.xmethods.net/sd/2001/CurrencyExchangeService.wsdl. Ce Web Service fournit les parités entre la quasi-totalité des monnaies du monde. Ce projet se trouve dans le dossier Call_XMethods. Lancez-le, et compilez-le. Passez sur le Pocket PC et choisissez les monnaies dans les listes déroulantes, et cliquez sur le bouton Convert. Vous devez impérativement avoir une connexion à Internet valide (par ActiveSync par exemple), pour que ce programme fonctionne. (Figure 4)

Les spécificités du Pocket PC Phone Edition et du Smartphone

Le Pocket PC Phone Edition n'est rien d'autre qu'un Pocket PC standard avec les fonctions de téléphone. Sa programmation est par conséquent identique à celle du Pocket PC. Le premier Pocket PC Phone Edition a été présenté par Bouygues début 2003 sous le nom de iPDA. Le même appareil se retrouve désormais chez les deux autres opérateurs, Orange le nomme Qtek 1010 et SFR NetPDA. A l'heure où nous écrivons ces lignes, ces appareils

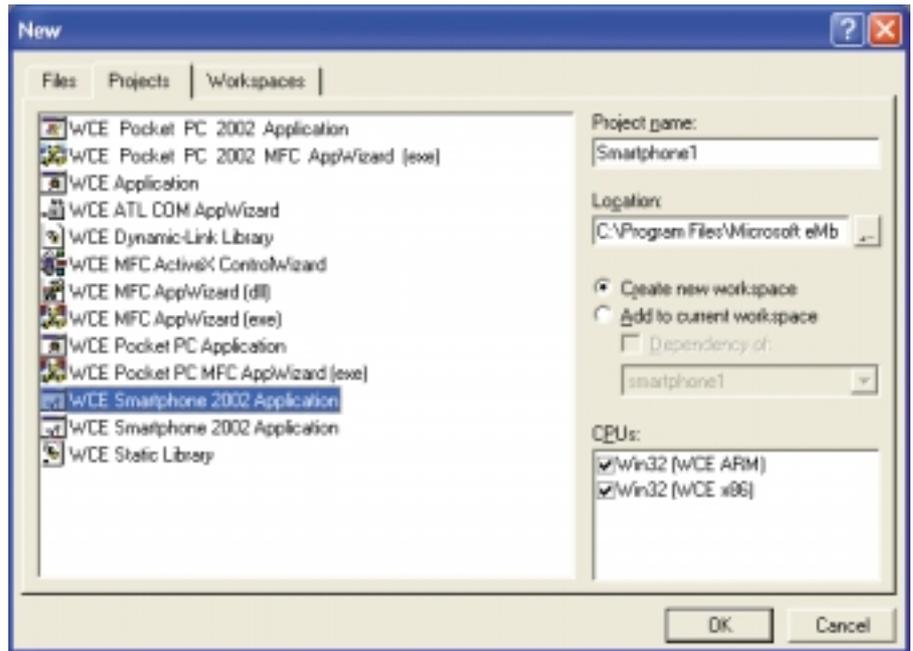


Figure 5 : Lorsque les SDK ont été installés dans eVC++, il suffit de choisir dans la liste l'appareil pour lequel on désire développer une application.

tourment sous Pocket PC 2002, mais les versions 2003 devraient être disponibles très rapidement (probablement chez Bouygues au départ). La Smartphone suit la même évolution que son frère, le Pocket PC, mais avec une génération de retard, aussi bien sur les

modèles, que sur les outils de développement. Sa carrière a débuté en France début 2002, avec le SPV (Son Photo Video) d'Orange. Il y a eu ensuite le modèle e100 pratiquement identique, puis le e200. On retrouve donc la même évolution avec les SDK correspondants. Pour le moment, les développements pour Smartphone ne peuvent s'effectuer qu'en C++, dans les environnements eVC++ 3.0 ou eVC++ 4.0, avec bien sûr les SDK correspondants et le "Petzold" à portée de main !

Si les développements sur le Smartphone ne posent pas de problèmes particuliers par rapport au Pocket PC, il est en revanche important de bien prendre en compte ses spécificités. La première d'entre elles est que ce n'est pas un appareil de saisie d'informations. Il est pratiquement impossible à l'utilisateur de saisir un texte de plus d'une trentaine de caractères, un SMS par exemple. Il faut donc en tenir compte. Deuxièmement, l'écran n'est pas tactile comme sur le Pocket PC. Par conséquent, le cheminement à l'intérieur de l'application doit être pensé différemment. Le Smartphone ne s'utilise qu'avec une seule main, et c'est en général le pouce qui est utilisé. Toutes les applications doivent être plein écran et les menus ne peuvent être composés que de deux entrées (l'entrée droite et l'entrée gauche). Enfin, l'entrée gauche ne devrait en principe ne contenir qu'une seule entrée.

■ Alain Coupel

4 sites incontournables :

Télécharger les outils de développement et les SDK

Voici l'adresse pour télécharger tous les outils et SDK Microsoft cités dans ce dossier msdn.microsoft.com/vstudio/device

Développer pour Pocket PC

Aucun développeur pour Pocket PC ne peut ignorer le site CodePPC. Celui-ci est une véritable mine d'informations qui fournit des initiations à la programmation en C++, VB, NS Basic, Java et .NET, des exemples de programmes avec les codes sources, des astuces de programmation, ainsi que des conseils pour améliorer vos applications. Celui qui ne dispose pas de liaison Internet à haut débit pourra même passer par ce site pour obtenir les CD des outils, des SDK, ou des runtimes. www.codeppc.com.

Développer pour Smartphone

Orange met à la disposition des développeurs un site très complet baptisé OrangeLab, sur la programmation d'application. Il suffit de s'inscrire (c'est gratuit), pour accéder à des dossiers et des actus. Ce site traite également des Smartphone non Microsoft tels que le Treo sous Palm OS ou les machines sous Symbian (Nokia, Sony, Ericsson). Vous pouvez également proposer vos propres applications. OrangeLab est en effet à la recherche d'applications ainsi que de contenus. www.orangelab.biz/fr/biz

Désactiver le Smartphone

developer.orangews.com/orgspv/comdefqfr.aspx

L'ergonomie est la clé de la réussite

Gildas Letort est directeur technique chez WEBTISS Technologies. Il développe des applications pour les PDA et le Pocket PC en particulier. Voici les conseils qu'il donne aux futurs développeurs.

WEBTISS Technologies est un intégrateur de solutions Internet/intranet mobiles. La société est spécialisée dans les solutions pour l'entreprise, avec cependant des réalisations B to C telles que le portail d'accès par mobile aux informations bancaires de la Société Générale, le Portail d'accès aux données sportives d'Eurosport, ou encore le programme de gestion des cartes de fidélité et de réservation de la compagnie aérienne El Al. Pour les entreprises, WEBTISS propose des solutions destinées par exemple aux chauffeurs livreurs ou au technico-commercial, leur permettant de programmer leur tournée, de saisir les informations sur leur PDA et de les transférer aussitôt vers le système de l'entreprise. Un tel système valorise le travail de ces employés et accélère largement les temps de traitement. Les applications de WEBTISS, peuvent être développées aussi bien pour Palm OS et Symbian que pour Pocket PC, mais pour Gildas Letort, la tendance actuelle des clients est plutôt orientée vers le Pocket PC.

Un développement par couches

Les développements sont effectués à partir de briques logicielles, ce qui permet de modifier plus facilement l'application pour la porter sur une autre plateforme, telle que le Palm OS, ou Tablet PC. L'application est ainsi divisée en 4 couches. La première est la base de données. " Il est inutile de réinventer la roue, aussi il est préférable d'utiliser les bases disponibles sur le marché " nous confie Gildas Letort. WEBTISS a choisi iAnywhere de Sybase, mais d'autres choix sont bien entendu possibles. La couche suivante est celle d'accès aux données. C'est le standard SQL qui prédomine. La troisième couche est la couche métier. Celle-ci est

développée en langage-objet qui peut être du C++, du C#, ou encore du Java. Le choix est effectué en fonction des désirs des clients. Certains d'entre eux souhaitent en effet reprendre la maintenance de l'application par la suite. Ils veulent par conséquent que le langage de développement utilisé soit celui pour lequel ils possèdent des compétences en interne. La quatrième couche enfin est la couche de présentation.

" Le développement pour les Pocket PC a atteint l'âge de la maturité. Il doit être effectué

par exemple avec iAnywhere) ce qui permet, côté serveur, d'attaquer diverses bases telles que Oracle, DB2, SQL Server ... Pour les applications plus simples, c'est-à-dire qui ne contiennent pas un très grand nombre de tables, il est possible d'utiliser .NET et des protocoles basés sur les Web Services gérés par ADO .NET.

Ce qu'il faut faire et ne pas faire

Il ne faut pas répliquer la base sur le Pocket PC. Il suffit de charger le sous-ensemble dont

l'utilisateur aura besoin pendant la journée. Cela permettra d'économiser de la place en mémoire et d'optimiser les temps de réponse et la bande passante. Il faut développer en pensant à l'ouverture du logiciel vers d'autres plates-formes, ou même d'autres serveurs. Le Pocket PC est avant tout un instrument de saisie. Par conséquent, les traitements doivent être délégués vers les serveurs. Enfin, aucune application ne peut être crédible si elle ne prend pas en compte la sécurité des données. Le Pocket PC n'est pas à l'abri d'une perte ou d'un vol. Les données doivent être cryptées et leur accès doit être complexe. Il faut également s'assurer de leur intégrité et également penser à leur persistance. En effet, un Pocket PC " oublie " tout lorsque la batterie est à plat. Des utilitaires de backup en



L'application Geofleet de WEBTISS permet de localiser sur une carte tous les appareils GPS/GPRS d'une flotte.

avec la même stratégie et la même rigueur de développement que pour les applications serveur " déclare Gildas Letort. Ainsi, par exemple, les développements avec le Compact Framework .NET sont en tous points semblables à ceux effectués sous le Framework .NET. Il ne faut pas oublier qu'un Pocket PC communique toujours avec serveur. Pour des projets fonctionnels élaborés, Gildas Letort préconise d'utiliser un middleware (MobiLink

mémoire flash ou sur des cartes de stockage externe type Secure Digital ou Compact Flash doivent donc être prévus dans le programme. Enfin pour Gildas Letort, la clé de la réussite est l'ergonomie. L'écran du Pocket PC est petit et il faut en tenir compte. L'utilisateur doit être pris par la main. Il faut éviter les options sur la page d'accueil et fournir un chemin unique pour arriver. Il vaut mieux multiplier le nombre d'écrans et bien canaliser l'utilisateur. ■

OPEN-SOURCE.COM

Linux - Open Source - Logiciels Libres

2004 sera une année charnière pour Linux et l'Open Source. Les nouvelles solutions "Entreprise" annoncées doivent démontrer leur capacité à réellement concurrencer les outils propriétaires, notamment dans les SGBD. On verra également si Linux est capable de s'imposer aussi sur le poste de travail et de concurrencer frontalement Windows (en entreprise).

Sur ce sujet souvent passionnel, nous avons tenté de faire le pari de l'objectivité, et de poser les vraies questions.

Oui, Linux et l'Open Source offrent des réponses crédibles alternatives aux outils propriétaires.

Aujourd'hui, opposer Libre/Propriétaire n'a plus de sens, excepté dans certains cas précis. Il est même souvent intéressant de les mixer.

Non, le coût global (caché et non caché) n'est pas forcément plus avantageux que des licences commerciales, car le coût d'un logiciel, d'une technologie, d'un système, ne se résume pas au prix d'achat. Et dans l'éternel match, Linux est-il moins cher que Windows ? A notre avis, aucune étude précise, neutre et exhaustive n'a pu fournir une réponse indiscutable.

Il ne faut surtout pas d'obscurantisme technologique. Chaque entreprise doit évaluer elle-même la question.

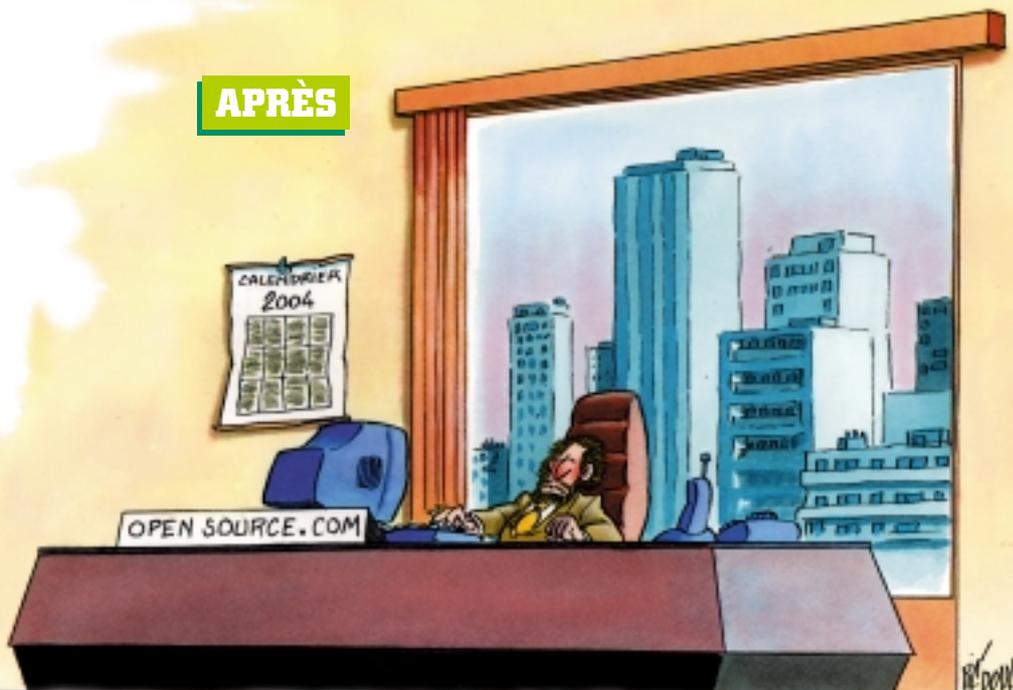
Etant donné que Open Source / Logiciel Libre ne rime pas uniquement avec Linux, ce dossier est divisé en deux grandes parties.

François Tonic

AVANT



APRÈS



Le modèle économique éclate

2003 fut marquée par de vastes mouvements autour de solutions Open Source et du système Linux. Il se dessine désormais très nettement, et plus que précédemment deux sections : un monde commercial, un monde " libre " non commercial. Voyons de quelle manière se structure l'offre.

Deux annonces fortes ont marqué 2003 : Novell reprenant SuSe après Ximian, Red Hat lâchant son offre gratuite et " grand public ". Ces deux annonces tendent vers une même logique : Linux d'entreprise. Pour le moment, Linux pour le grand public n'est pas une réalité (ou une priorité des éditeurs), hormis un nombre limité d'amateurs. Linux en entreprise est une réalité, de plus en plus. Et ce, à deux niveaux : le serveur, mais aussi le poste de travail, même si cela est encore lent à se déployer massivement. Le véritable marché, le véritable défi reste donc l'entreprise, avec une certaine priorité pour le serveur et le renforcement des offres et services sur le poste de travail.

Les distributions

Comme énoncé ci-dessus, deux offres de système s'affrontent : le Linux "commercial" et payant, et le Linux "traditionnel". On assiste au renforcement du premier type de Linux. Red Hat se concentre sur son offre entreprise payante. Novell reprend SuSe pour remplacer son propre système Netware. C'est là aussi une offre payante. Ce mouvement n'est pas nouveau. À ce tableau, on ajoutera aussi UnitedLinux (maintenant mort ou quasi) ou encore Mandrake.

On assiste sans aucun doute à une sorte de concentration des éditeurs de distributions. Aujourd'hui, il y a deux situations : une poignée de distributions reconnues faisant référence, et une kyrielle de distributions anonymes plus ou moins connues par certains



milieux Linux. L'émergence d'éditeurs est un facteur important. Une entreprise ne peut pas se permettre d'utiliser un système n'ayant pas de supports, de garanties minimales, d'évolution dans le temps. La présence, aujourd'hui, d'un nombre réduit de distributions résulte de cette demande. Bien entendu, on peut installer une distribution Linux inconnue, conçue pour un petit groupe de personnes, mais quels sont les services annexes ? Comment évoluera le projet ? Quelle est la pérennité ? À toutes ces questions, l'éditeur fournit un minimum de réponses. Plus que jamais dans l'avenir, les éditeurs de distributions auront une place de plus en plus primordiale, surtout, si Linux s'installe dans le grand public.

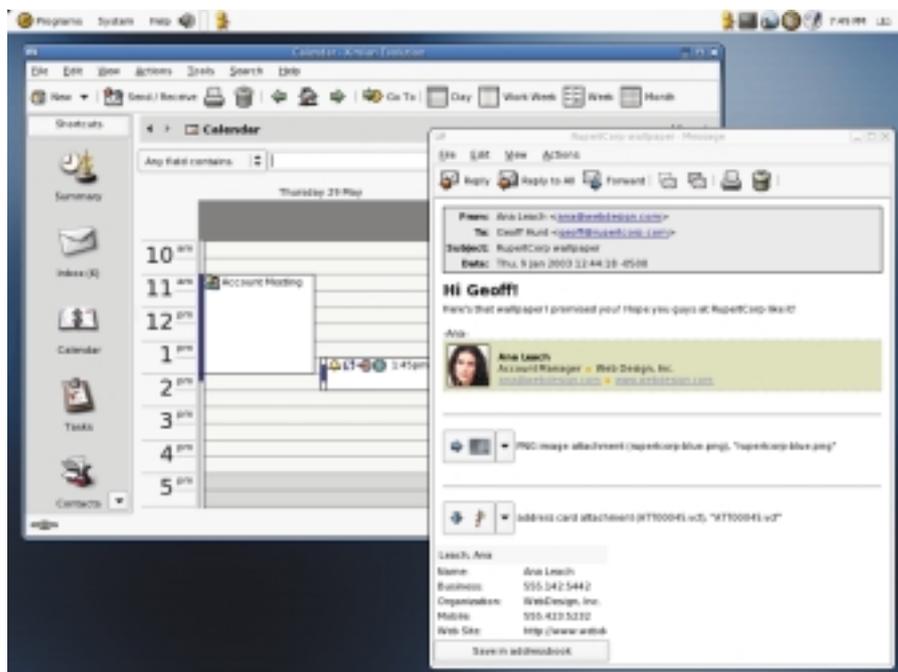
Cependant, à cela, on peut rajouter un niveau supplémentaire : les constructeurs. Là, la plupart des constructeurs qui estampillent du logo Linux sur leurs machines supportent, en général, plusieurs distributions, les plus connues, les plus utilisées. Donc, là encore, une sélection est faite, tendant à réduire de facto, le marché à quelques noms, 3 ou 4, le plus souvent. Une des différences entre une

distribution d'un éditeur estampillé Serveur ou Entreprise et une distribution " indépendante " concernera peu les outils (ils sont en général identiques, même si les éditeurs rajoutent parfois des outils spécifiques), mais plutôt les services et le support. Et cela à un prix. Parfois, les tarifs Red Hat sont très éloquentes.

Les logiciels aussi ?

Dans les logiciels, on retrouve quelque peu la situation des distributions Linux, mais avec plus de diversité. Nous avons en vrac : des consortiums, des éditeurs commerciaux, des projets Open Source de grande envergure, ayant ou non des services payants et des SSII faisant aussi des logiciels. Notons tout de même que la plupart des projets Open Source sont indépendants, de taille souvent réduite. Dans l'Open Source, la pérennité d'un projet passe par la grosseur de sa communauté et le nombre d'utilisateurs.

Commençons par les éditeurs Open Source. Ils ne vendent pas, en général, les logiciels. Ils demeurent Open Source et donc en libre téléchargement. Mais comme il faut bien un modèle économique pour vivre et faire évoluer le logiciel, la partie commerciale concerne les ser-

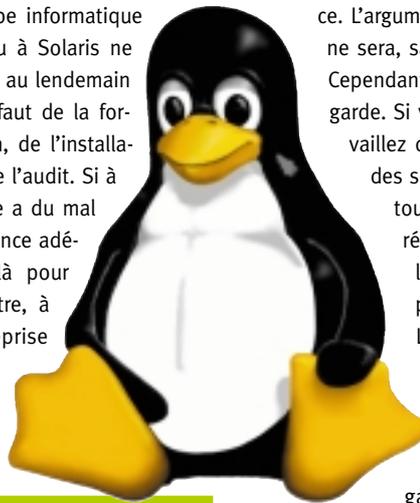


vices. Ainsi, si on regarde les services proposés par JBoss, on peut remarquer que les prix sont dignes d'un éditeur propriétaire. MySQL relève exactement de la même logique, mais va encore plus loin en prenant réellement l'aspect d'un éditeur "normal". Si on peut toujours télécharger MySQL librement, son usage en est toutefois limité. Il est interdit de distribuer le SGBD avec une application dont le code n'est pas ouvert (dans la pratique, tant que le code est ouvert, l'achat d'une licence n'est pas nécessaire). De plus, la version téléchargée n'a aucune garantie. Bref, si une SSII souhaite "bétonner" une offre MySQL, elle est obligée de souscrire des services et des versions "pros" de MySQL. Heureusement, selon la quantité, les prix baissent... PostgreSQL est bien plus flexible à ce niveau là. Mais les ambitions ne sont peut-être pas identiques. Sur le marché Linux, les grands éditeurs investissent beaucoup. Cependant, cette présence reste limitée à certains domaines (entreprise, serveur, développeur). Hormis Microsoft, tous les

poids lourds de la base de données supportent Linux sans pour autant offrir des prix au rabais. D'autres rachètent purement et simplement des éditeurs Open Source. C'est le cas de Novell qui passe d'un modèle Netware (son système) au modèle Linux / Open Source en rachetant SuSe et Ximian. Aujourd'hui, une partie du marché Linux / Libre est dominée par quelques grands éditeurs. La dernière offensive est due à Sun qui propose une distribution Linux complète pour le poste de travail, vendue à une centaine de dollars US par utilisateur.

SSII et formateurs

Jusqu'à présent, les SSII spécialisées dans l'Open Source / Linux / Logiciels Libres accaparaient la majorité du marché avec les services des grands constructeurs. Des sociétés comme IdealX ou Alcôve font du Libre depuis plusieurs années et ont réussi à survivre. Ce type de société ne vend pas spécifiquement du logiciel, mais avant tout du service, dont le prix ne diffère en rien des services propriétaires. De temps en temps, une SSII Open Source propose aussi une activité éditeur. Le service est en principe plus rentable économiquement pour une SSII. Avec l'essor du Libre et de Linux en entreprise, les besoins sont conséquents. Une équipe informatique habituée à Windows ou à Solaris ne peut pas passer du jour au lendemain à Linux et au Libre. Il faut de la formation, de la migration, de l'installation, et bien entendu de l'audit. Si à court terme, l'entreprise a du mal à trouver de la compétence adéquate, les SSII sont là pour apporter cela. Par contre, à moyen terme, l'entreprise devrait avoir plus de facilité à trouver les compétences nécessaires, soit via de la



Linux a rapporté 2,5 milliards \$ à HP

C'est très officiellement que HP a annoncé mi-janvier 2004 que les revenus générés autour de Linux (support, logiciels, services, matériels) étaient de 2,5 milliards de dollars en 2003. C'est la preuve que des constructeurs savent profiter de Linux, essentiellement en entreprise. HP mise beaucoup sur les services autour de Linux. Les prévisions pour 2004 sont de dépasser ce chiffre. Sans surprise, les serveurs HP sous Linux ont été des valeurs sûres.

formation, soit via l'embauche. Encore faudrait-il trouver de bons informaticiens Open Source / Linux, compétents en la matière. Une certification sera sans doute nécessaire. À court terme, le mouvement des SSII classiques vers l'Open Source et le Libre se fera sans doute massivement. Un marché s'ouvre pour elles et comme l'entreprise aura besoin de ce type de compétence et d'outils, la SSII ne peut que suivre. Les SSII spécialisées auront donc, demain, une nouvelle concurrence. L'argument de l'ancienneté en Libre ne sera, sans doute, pas suffisant.

Cependant, l'entreprise doit prendre garde. Si votre SSII avec qui vous travaillez depuis des années, propose des services Linux / Libre, vérifiez tout de même sa compétence réelle. Il ne faudrait pas que le budget informatique dérape à cause des Logiciels Libres. Mais encore une fois, le coût caché de Linux et du Libre est identique, peu ou prou, aux outils commerciaux. Vous

gagnerez effectivement en indépendance (et encore faut-il nuancer sur quelques points) et presque toujours sur la licence, mais les coûts sont, au final, peu différents de ceux des logiciels propriétaires. Les SSII qui n'ont pas encore lorgné sur l'Open Source auront tout intérêt à s'y intéresser, et vite. Mais, on peut sans doute s'attendre à voir émerger une forte sous-traitance par des SSII pour des projets Open Source, surtout si elles ne possèdent pas de compétences internes.

■ François Tonic

Les éditeurs propriétaires aiment Linux...

Avec la vague Linux en entreprise, les éditeurs investissent de plus en plus sur ce système, il n'y a qu'à voir la multitude d'annonces de Novell depuis quelques mois. Oracle a aussi, depuis longtemps, pris en compte Linux. Oracle 10g le supporte et l'éditeur a même sorti son offre E-Business Suite Special Editions uniquement sous Linux, ainsi que sa solution d'outsourcing compatible avec le Tux. Même réflexe chez Sybase. L'éditeur a porté sa gamme de base de données dès 1999 (Adaptive Server Enterprise, Enterprise Application Server, SQL Anywhere Studio).

Vers une recomposition des SSII libres ?

Deux mouvements ont marqué 2003 ; le rachat d'éditeurs Linux et des rachats de SSII spécialisées. Il ne s'agit pas de concentration, mais d'atteindre une masse critique sur un marché prometteur. On remarquera que tout cela se passe uniquement dans le secteur entreprise. Dernièrement, c'est OpenWide qui a racheté Accelance MSP, une société spécialisée dans l'hébergement et l'infogérance Linux / Open Source. Il y a eu aussi le rachat d'Aurora par Business & Decision, ou encore, courant 2002, le rachat d'Alcôve. Côté éditeur Linux, Mandrake a réussi à redresser quelque peu la situation, mais l'éditeur reste assez fragile et sa stratégie entreprise n'est toujours pas assez agressive.

IBM : une stratégie multiple sur Linux et l'Open Source

Depuis plus de trois ans, IBM investit massivement dans l'Open Source et Linux. S'il fut en son temps le premier grand constructeur à croire en Linux, aujourd'hui, les concurrents l'ont imité. HP a su profiter de la dynamique Linux et dans une moindre mesure, Sun qui a longtemps cherché à définir sa stratégie. Analyse et commentaires de Michel Granger et Remy Baranger d'IBM.

La stratégie d'IBM dans le monde Libre et Linux est multiple. On peut définir plusieurs domaines d'actions :

- matériel : compatibilité des gammes serveurs avec les principales distributions du marché
- services autour de Linux et l'intégration des logiciels / applications
- portage des applications IBM et de ses filiales sous Linux
- Outil de développement : Eclipse pour la fondation.

Serveurs, Open Standards : IBM mise sur tout. Au-delà de cela, IBM se plaît à préciser un point important. L'entreprise ne fait pas de

l'Open Source mais de l'Open Standard. Cela signifie tout simplement qu'IBM suit et supporte les standards ouverts. IBM reste avant tout un éditeur commercial. La prise en compte des standards ouverts a



Michel Granger

l'avantage de suivre les mouvements du marché, tout en évitant les critiques et d'enfermer son application dans du propriétaire. Le standard ouvert est une tendance lourde (et un argument) depuis plusieurs années. Il n'y a qu'à voir le gros travail d'IBM dans les Web Services, notamment avec Microsoft.

Si la grande entreprise est déjà souvent convaincue de l'intérêt de Linux et de l'Open Source au moins côté serveur, qu'en est-il pour les petites sociétés ? " Linux n'est pas cher. Sur les PME/PMI, c'est là que l'on aura la plus forte croissance " précise Michel Granger. C'est dans cette optique, que depuis peu, IBM (comme d'autres éditeurs) propose des solutions complètes repackagées, dédiées aux PME/PMI.

" Linux est généralisé sur les serveurs, c'est un acquis. En dehors de cela, aujourd'hui, ce n'est pas le cas. Quelques grandes sociétés ont déjà sauté le pas et montrent ainsi l'exemple. Les éditeurs vont aussi y aller. Avec Linux, on peut répondre à 95 % des besoins. Finalement,

Linux est un faux problème. L'important, ce sont les applications, il ne faut pas de rupture. Sur les serveurs, Linux n'est pas un problème. Sur le poste de travail, il y aura un leader." poursuit Michel Granger. Cependant, le monde Linux garde toujours une multiplicité des distributions, ce qui n'aide pas les développeurs ni les entreprises. Michel Granger ne cache pas la situation : " Oui, c'est un frein surtout quand on utilise l'interface. Il faut un standard de fait. ". Cela se vérifie largement dans le Linux d'entreprise, le marché étant aujourd'hui dominé par Red Hat, même si Novell mise beaucoup avec le rachat de SuSe.

IBM & Novell

Suite à l'annonce du rachat de SuSe par Novell, pas mal de questions se sont fait jour au sujet de l'investissement d'IBM de 50 millions de dollars US dans Novell. La raison en est finalement assez simple, comme nous le précise Michel Granger : " Nous avons un accord avec SuSe. Il n'y a pas de rupture d'accord (suite à son rachat par Novell). On voulait continuer à supporter les principales plates-formes (Linux). "

Le Grand Bleu mise aussi beaucoup sur le service, notamment pour aider les entreprises à

migrer vers Linux, ou à intégrer les applications existantes dans un nouvel environnement. Remy Baranger expose les dernières avancées d'IBM dans ce domaine : " IBM a mis en place un nouveau programme afin de fournir les ressources nécessaires, notamment des cours en ligne, pour migrer de Windows NT 4 à Linux. Nous disposons aussi en France, d'un centre de portage situé à Corbeil-Essonnes. "

Eclipse : le fer de lance

Depuis quelque temps, IBM tente de ne plus apparaître trop ouvertement dans le consortium Eclipse. Mais en à peine deux ans, cet outil a su s'imposer comme un adversaire de taille pour les autres éditeurs du secteur comme Borland. Et il est en passe de devenir incontournable.

" Eclipse est utilisé par deux types d'utilisateurs : les développeurs indépendants et les éditeurs. Eclipse, dont la version 3 sortira dans quelques mois, dépasse le simple cadre de Java. D'autres langages sont acceptés (ndlr : C# par exemple). Dans Eclipse, on y trouve une fondation et des extensions. L'outil demeure rudimentaire. Il ne dispose pas d'assistants. Si on souhaite changer de dimension avec Eclipse, on doit alors utiliser des outils de productivité, par exemple ceux de Rational. " précise Michel Granger. La dernière initiative de Sun pour harmoniser les outils de développement ne semble pas, pour le moment, émouvoir IBM. Remy Baranger apporte un petit commentaire : " On la regarde juste comme un groupe de travail au JCP. "

■ F.T.

ACCOMPAGNER LA CROISSANCE de LINUX en ENTREPRISE

" Nos clients sont demandeurs de Linux. Notre réponse : adapter toutes nos plates-formes à Linux et aider ce marché à devenir mature et professionnel " déclare Serge LESAGE, le nouveau directeur Business development & Linux, chez IBM.

Les grands comptes passent aussi à Linux. " Le changement a été net au 2ème semestre 2003. Par exemple, 17% des ventes de Z-series étaient sous Linux, alors que le CA en était insignifiant l'année précédente ", S. Lesage voit dans cette migration (passer d'OS 370 à Linux), une preuve de la robustesse et de la maturité de ce dernier.

L'investissement de Big Blue dans Linux est stratégique et global, il concerne aussi bien le matériel, le logiciel que les services : " Il se chiffre en milliards de dollars dans le monde ; c'est pour nous un signe fort porté à ce marché ". Il se traduit notamment par l'accueil et l'assistance aux éditeurs applicatifs, pour les aider à porter leurs logiciels sous Linux.

Ainsi, IBM proposera bientôt un package de gestion pour PME sous cet OS. De plus, sur le site de Montpellier, les grands comptes peuvent venir tester leurs applications sur gros systèmes, migrées sous Linux.

JK

Mandrake se réveille



Après une passe très difficile, l'éditeur Mandrake commence à redresser la tête, mais tout n'est pas encore réglé. Le Français a toujours du mal à intégrer une offre forte pour les entreprises. Entretien avec Gaël Duval, cofondateur de l'éditeur.

Programmez ! : Comment se porte aujourd'hui la société Mandrake ? L'avenir est-il plus serein qu'en 2002 - 2003 ?

Gaël Duval : La situation s'est encore améliorée en 2003 : nous sommes en flux de trésorerie positif depuis mars 2003 et nous avons connu notre premier trimestre bénéficiaire en octobre-décembre dernier, ce n'était plus arrivé depuis notre premier exercice en 98-99.

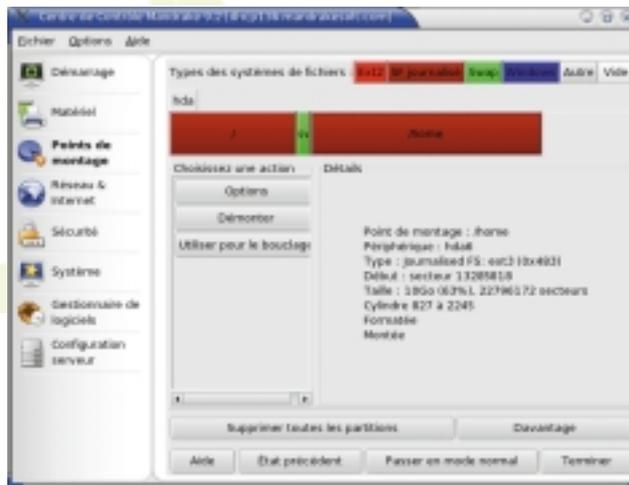
P ! : Quelle stratégie avez-vous pour 2004 et les années à venir ?

GD : Nous allons continuer à développer Mandrake avec une approche la plus globale possible (c'est à dire potentiellement mondiale), avec comme lignes directrices la convivialité et la puissance. Parallèlement, nous allons renforcer notre offre petites et moyennes entreprises. Pour le cas particulier de la France, nous allons continuer de progresser sur le marché des administrations et des grands comptes.

P ! : Vous avez lancé MandrakeMove. Quel est son positionnement et son intérêt ?

GD : MandrakeMove est un concept original qui permet d'avoir un système Linux complet dans sa poche et de pouvoir l'utiliser n'importe où. Outre l'intérêt du "LiveCD" qui permet d'utiliser le système directement sans l'installer, MandrakeMove utilise une clé USB pour stocker automatiquement toutes les données utilisateur (par exemple une configuration de

graphique, utilitaires graphiques de mises à jour en ligne...) ont été introduites par Mandrake, puis en partie copiées par la suite. Nous allons continuer à innover rapidement. Il faut aussi se souvenir que lorsque MandrakeSoft a débuté il y a 5 ans, les deux acteurs dont vous parlez étaient déjà les deux versions commerciales de Linux les plus connues à l'époque, et MandrakeSoft s'est hissé à un niveau comparable en terme de base utilisateur et de reconnaissance, alors que des centaines d'autres distributions Linux n'ont pas eu cette chance pendant le même laps de temps. Il faut donc nous laisser le temps de nous développer commercialement, tout en suivant notre voie, qui est beaucoup plus novatrice en



terme de "business model" que nos concurrents et avec un potentiel supérieur à long terme en matière de réussite d'entreprise.

P ! : Quels sont encore les limites de Linux par rapport à d'autres Unix, Windows Server 2003 ou MacOS X ?

GD : Au niveau technologique, il me semble qu'on pourrait inverser la question. Linux souffre encore d'un manque important d'adhésion des éditeurs de logiciels commerciaux à cette plate-forme, ce qui fait qu'on est trop souvent encore obligé d'utiliser Windows pour pouvoir se servir de tel ou tel logiciel. Mais il n'y a pas de raison que les choses ne changent pas d'ici quelques années, la vague du Logiciel Libre tenant à la fois du rouleau compresseur et de la révolution culturelle.

■ F.T.

P ! : Comment voyez-vous, aujourd'hui, le marché des systèmes Linux en France ? Sort-il réellement du serveur pour investir le poste de travail ? Si oui, dans quelles conditions ?

GD : Oui, mais pas encore au niveau du grand public. La

demande se fait de plus en plus pressante au niveau du bureau d'entreprise.

P ! : Pensez-vous que l'entreprise reste le premier marché de Linux et des logiciels libres ?

GD : En ce qui concerne MandrakeSoft, le marché de l'entreprise est encore sous-développé, cela nous donne d'ailleurs une bonne marge de progression dans ce secteur, car effectivement c'est encore le premier marché Linux et de loin.

connexion internet, des emails, des documents...). Son positionnement est au départ l'utilisateur d'informatique qui se déplace beaucoup, mais MandrakeMove va vraisemblablement devenir une plate-forme sur laquelle vont se greffer de nouveaux développements particuliers pour, entre autres, le monde de l'entreprise.

P ! : Mandrake s'oriente visiblement de plus en plus vers l'entreprise. Comment se positionner face aux poids lourds tels que Red Hat et tout nouvellement Novell, avec le rachat de SuSe ?

GD : Nos produits sont extrêmement compétitifs et notre structure plus légère, plus réactive. Beaucoup oublient souvent qu'un grand nombre d'innovations qui ont rendu les distributions Linux plus simples à utiliser (installer

Linux se limite-t-il encore aux serveurs ?

L'arrivée de Linux dans les serveurs a " révolutionné " la donne à plus d'un titre. D'abord, en empiétant fortement sur les parts de marché des Unix propriétaires (AIX, Solaris, HP-UX...), ensuite en montrant qu'il existait sur les serveurs une alternative crédible à Windows. Un tel scénario est-il envisageable du côté du poste client ?

Les atouts de Linux

Linux n'est pas dépourvu de qualités. Par exemple, la stabilité de son noyau n'est plus à démontrer. De nombreux éditeurs et fabricants d'ordinateurs le distribuent, à commencer par le premier d'entre eux (IBM) qui en a même fait un des piliers de sa stratégie de développement. Son faible coût de revient, du pour une bonne partie à la gratuité des licences d'utilisation et à la disponibilité des compétences pour le mettre en œuvre sont également des points très importants. Depuis quelques années, des interfaces graphiques (KDE, Gnome ...) qui n'ont plus grand-chose à envier à Windows ou Mac OS X ont fait leur apparition, rendant ainsi presque inutile la ligne de commande. Du côté de la bureautique, des produits tels qu'OpenOffice offrent les mêmes fonctionnalités que Word ou Excel pour un coût de revient extrêmement faible. Alors pourquoi, malgré toutes ces qualités Linux ne représente aujourd'hui que 1 ou 2 % des postes clients installés (chiffres IDC) ?

Les faiblesses de Linux

Linux est un système d'exploitation conçu par des ingénieurs pour des ingénieurs, contrairement à des systèmes d'exploitation tels que Windows XP ou MacOS X qui ont été conçus dès le départ pour des utilisateurs non informaticiens.

Il en résulte une très forte adaptation aux besoins " techniques " des applications serveur (Sendmail, Apache...) mais aussi de grosses lacunes dans la prise en charge des besoins des utilisateurs finaux. On peut citer par exemple, les déploiements qui sont très complexes, la très faible intégration des différents composants (bien souvent les distribu-



tions Linux ne sont que des superpositions de produits indépendants), ou encore des drivers matériels pas toujours immédiatement disponibles.

Si l'on ajoute à cela, les tarifs prohibitifs pratiqués par des sociétés comme RedHat pour le support technique, Linux n'apparaît plus nécessairement comme une solution intéressante.

Mais au final, la principale faiblesse de Linux réside sans doute dans son incompatibilité avec les applications Windows. Les utilisateurs ont toujours été très sensibles à la compatibilité des systèmes d'exploitation avec leurs logiciels.

Linux versus Windows ?

Microsoft a clairement fait le choix de séparer les systèmes d'exploitation serveurs (gamme Windows 2003) de ses systèmes d'exploitation clients (gamme Windows XP) pour mieux coller aux besoins des utilisateurs. Le rythme des évolutions annoncées pour ces deux plates-formes est différent puisque les OS clients vont évoluer plus vite (longhorn client est annoncé pour 2005) que les OS serveurs (Longhorn serveur est annoncé pour 2006). Cette séparation des plates-formes et l'évolu-

tion plus rapide de la partie cliente chez Microsoft montre clairement, d'une part que les attentes sont différentes et d'autre part que le poste client est appelé à évoluer plus rapidement à l'avenir.

De plus, Microsoft joue à fond la carte de l'intégration entre ces différents produits (Windows, Internet Explorer, Media Player, Office ...) pour offrir toujours plus d'intégration et d'interopérabilité.

Du côté de Linux, l'interface graphique n'est toujours pas définitivement standardisée, il subsiste encore aujourd'hui plusieurs plates-formes concurrentes et les applications manquent souvent de cohérence dans certaines distributions. Il ne suffit pas de mettre côte à côte un navigateur, un lecteur multimédia, un traitement de texte et un tableur pour obtenir un poste de travail correctement intégré !

Conclusion

Après une entrée fracassante dans le monde des systèmes d'exploitation pour serveurs, Linux a de sérieuses difficultés à gagner des parts de marché sur les postes clients.

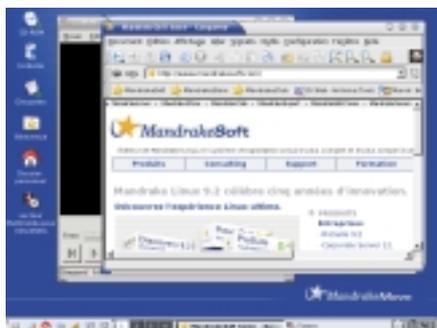
Interfaces graphiques moyennement ergonomiques, drivers pas toujours disponibles, distributions peu intégrées sont autant d'inconvénients au passage à Linux sur le poste client. Si l'on y ajoute les difficultés liées à l'incompatibilité avec les applications Windows existantes et la nécessité de reformer les utilisateurs, Linux perd beaucoup de ses attraits. Il existe néanmoins deux segments de marché susceptibles d'être conquis par Linux sur le poste client. Il s'agit des postes clients ultras légers pour lesquels la seule application importante est le navigateur web (Mozilla) et des postes bureautiques de base dans les organisations recherchant des solutions low-cost. Pour les autres, Windows restera sans doute encore pour longtemps la plate-forme cliente de référence.

■ **Médéric Morel**

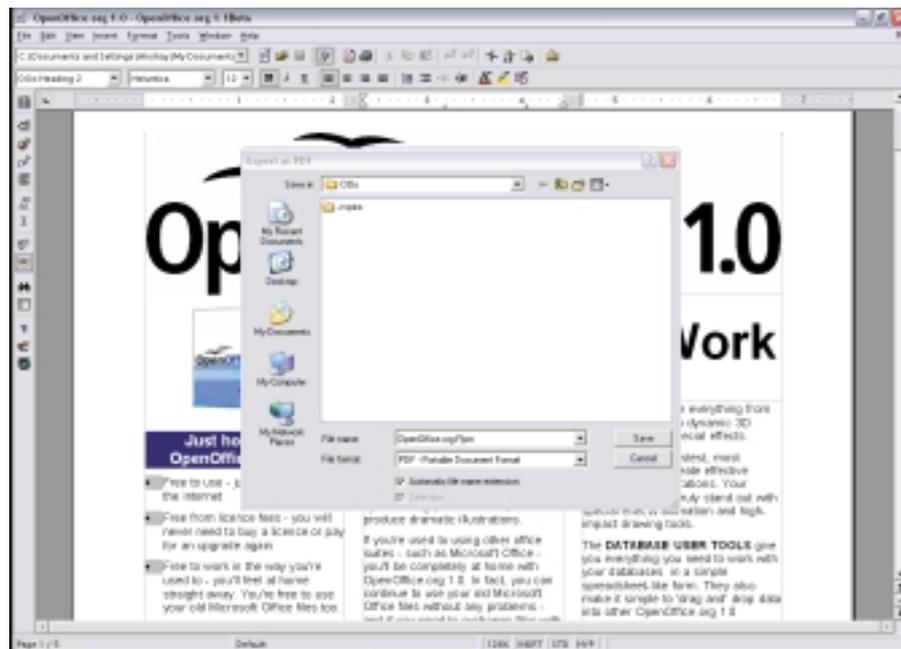
Directeur technique - Neoxia
mederic.morel@neoxia.com

Grand public : pari impossible ?

Linux ne concerne pas uniquement l'entreprise, mais aussi un public bien plus large, nous tous. Cependant, force est de constater que Linux reste réservé à une portion avertie des utilisateurs. La faute en incombe tout d'abord aux distributions qui ne sont pas encore au niveau d'un Windows ou d'un MacOS X. Même si certaines distributions ont une orientation très grand public, il n'en demeure pas moins que Linux reste un système difficile à maîtriser. Il faut toujours rassurer l'utilisateur, surtout quand celui-ci est un utilisateur lambda. Mandrake a très tôt misé sur le grand public, mais cette stratégie a failli lui coûter son existence, car le marché n'est pas encore prêt ou mature. Surtout, les distributions ne sont pas assez adaptées. Or, seul un éditeur de distributions peut réellement offrir un Linux, afin de sécuriser l'utilisateur et lui offrir toutes les prestations qu'il attend. Mais, un écueil demeure au niveau du grand public. Même si Linux est livré avec moult logiciels, l'utilisateur retrouve-t-il son environnement ? Ses habitudes ? Ses logiciels ? Il y a un effet psychologique à ce que les grands éditeurs



portent leurs logiciels sous Linux. Tant que cela ne sera pas généralisé, Linux auprès du grand public aura du mal à convaincre. De plus, il faudra avant toute chose régler le problème du support matériel par les constructeurs. Ils doivent impérativement supporter Linux, afin d'éviter que des projets non officiels



ciels ne le fassent. Si cela n'a rien de choquant, le projet non officiel n'offre pas les mêmes garanties qu'un pilote constructeur. Si la situation tend à s'améliorer, elle est loin d'être optimale, surtout dans les périphériques de tous les jours. Dans l'idéal, il faudrait que systématiquement, les constructeurs de périphériques incluent directement les cartons, les pilotes et applications Linux, comme le ferait Windows ou MacOS X.

Cependant, il y a un autre aspect à prendre en compte, c'est le côté logiciels Open Source / Libres. N'oublions pas qu'il ne faut pas disposer d'un Linux pour pouvoir en profiter, tant s'en faut. Un certain nombre de projets Open Source fonctionne sur une multitude de systèmes. Ainsi, si vous ne voulez pas utiliser un Microsoft Office pour faire votre bureautique, vous pouvez utiliser un Open Office sous Windows ou MacOS X. Là, par exemple, j'écris ces lignes dans le train, sous Windows, avec Open Office 1.1. Et cela est une réalité. Cependant, là aussi, il y a un problème de support, d'assistance technique, de mise à jour du logiciel. Dans bien des cas, l'utilisateur est laissé seul ou presque. D'autre part, le nombre de projets Open Source en Français, dans le logiciel et la documentation, n'est pas aussi important que cela. Bien entendu, les distributions Linux font des efforts importants. Mais tous les autres projets ? Voilà un problème que

Et à Programmez ?

À la rédaction de Programmez, nous utilisons une grande diversité d'outils et de systèmes. Pour la partie Web, nous utilisons des solutions Open Source classiques : Apache, MySQL, PHP. Pour la rédaction et les tests produits, là, on est obligé d'être très hétérogène : Windows, MacOS X, Linux, que ce soit sur PC ou PowerMac. Côté bureautique, les habitudes de Microsoft Office demeurent, même si Open Office est régulièrement usité par certains de vos auteurs favoris.

l'utilisateur devra prendre en considération. Car même s'il comprend l'Anglais, il est plus pratique et agréable d'utiliser une application en Français. L'autre facteur pour opter pour un logiciel libre, est de savoir s'il répond à vos attentes fonctionnelles. Car, il est peu envisageable de changer un logiciel propriétaire par un logiciel libre si celui-ci n'est pas au moins au niveau du logiciel propriétaire. Dès aujourd'hui, vous pouvez aisément changer un certain nombre de logiciels de votre Windows ou MacOS par des outils libres. Il faudra aussi que le logiciel Open Source sache reprendre les documents existants.

À vous de sauter le pas.

■ François Tonic

Des outils de développement alternatifs

L'Open Source et dans une moindre mesure les distributions Linux, ont révolutionné une certaine manière de faire de la programmation. Il y a des cas où l'Open Source peut apporter une solution alternative intéressante, et dans d'autres, où il vaut mieux rester en outils propriétaires.

À qui s'adressent les outils de développement Open Source / Libre ? À la grande majorité des développeurs, qu'ils soient indépendants, " amateurs ", dans une SSII, ou dans un service informatique d'une entreprise. Que l'on soit sous Windows, Linux ou MacOS X, les outils Open Source / Libre offrent bien souvent une bonne alternative. Pour un développeur indépendant ou un passionné, l'Open Source / Libre offre l'avantage d'être en code ouvert et surtout librement téléchargeable. Ces développeurs n'ont pas toujours les moyens d'acquérir les outils propriétaires. Cependant, on voit chez certains éditeurs propriétaires des versions Personal fonctionnellement plus limitées, mais gratuites. Malgré tout, bien souvent, il faut connaître l'Anglais pour utiliser les outils libres / Open Source. Pour l'entreprise et la SSII, l'Open Source permet de réduire considérablement la facture des licences (attention toujours à certaines solutions qui imposent des règles). Cela permet de développer un projet en ayant un minimum de contraintes envers l'outil. Bref, vive l'Open Source dans le développement moderne !

Les langages

Il y a certains types de langages sur lesquels il n'est guère possible d'utiliser des outils Libre / Open Source, car ils sont totalement ou quasi inexistantes. Pour tout ce qui concerne VB, VB.NET, difficile de se passer réellement des IDE Microsoft. Pour faire uniquement de l'ASP.NET, vous pouvez passer pour des projets légers à l'outil gratuit Microsoft WebMatrix (figure 1). Pour C#, il existe plusieurs IDE payants ou non. Pour ce qui est de Java, C, C++, là, pas de gros soucis, l'offre Open Source est largement suffisante et diversifiée, idem pour Python, Perl, PHP.

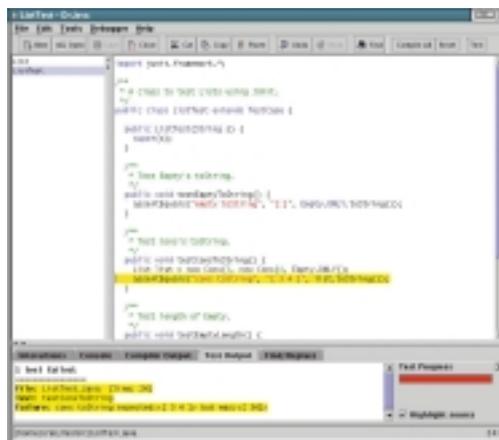
Les IDE

Certains IDE sont impossibles à changer pour de l'Open Source, ainsi, il n'y a pas d'équivalent à visual Studio .NET sauf à prendre des

IDE dans certains domaines précis comme C# avec C# Builder de Borland. Sous Java, on peut dès le départ utiliser un IDE Open Source, on pensera immédiatement à Eclipse. Par contre, il est difficile de changer d'IDE Java / J2EE quand le projet est déjà en cours de dévelop-



Figure 1 : WebMatrix



pement ou en production. Dans le cas d'une application Web, ou de composants EJB, il faut souvent réécrire le code. Pour les outils PHP / Python, changer d'outils est bien plus simple. Pour les applications mobiles, l'offre Open Source reste encore en retrait par rapport aux éditeurs, surtout dans les environnements Windows CE et .NET Compact Framework.

Les outils annexes

Le développement ne s'arrête pas aux IDE. Pour les debugs, les compilateurs, les généra-

tions de documentations, les tests unitaires, la configuration, le déploiement, etc., il existe de solides solutions Open Source : Ant, GCC, Struts, Junit, Marven... pour ne citer que ceux-ci. Il y a maintenant de plus en plus d'interaction entre les IDE propriétaires et des frameworks et outils Open Source. C'est pratique, car on travaille alors sur des standards ouverts. Privilégiez ces solutions si vous pouvez le faire.

Cependant, dans certains domaines précis et/ou pointus, l'Open Source est largement en retard ou n'offre pas d'outils crédibles. Pour la modélisation UML, mieux vaut utiliser des outils réputés du marché. Pour MDA et SOA, pareillement, on devra utiliser des outils propriétaires. Pour les Bases de données très lourdes, les Oracle, Sybase, SQL Server et autres DB2 sont toujours des solutions incontournables, même si MySQL 5 doit combler une partie du retard.

Les problèmes de l'existant

Il ne faut pas oublier le problème du code existant. S'il est possible de migrer d'un outil à un autre dans certains cas, il ne faut pas généraliser cela, car ce n'est pas la réalité. Comme dit plus haut, il y aura des problèmes dans certains projets Java. Pour VB, il faudra souvent tout réécrire et changer de langage. Pour les compilateurs, en théorie pas de gros soucis, sauf si des directives de compilations sont codées en dur, il faut vérifier la compatibilité avec le nouveau compilateur. De toute manière, quand on change de version d'IDE ou d'IDE purement et simplement, il y a dans la plupart des cas, des problèmes de compatibilités et de migrations qu'il faut évaluer et résoudre.

Donc, quel que soit le développeur, évaluez et testez le ou les outils Open Source que vous souhaitez utiliser. Dans le cas de codes existants, vérifiez la compatibilité avec le ou les nouveaux outils.

■ F.T.

MiniGuide d'équivalence Windows/Linux

Cet article n'a pas pour but d'être exhaustif (il serait en effet impossible de l'être à moins de remplir des dizaines de pages). Nous vous proposons une petite liste réaliste d'équivalence Linux de logiciels qui tournent sous Windows. Toutefois, comme logiciel libre n'est pas synonyme de "Linux", plusieurs d'entre eux tourneront indifféremment sous les deux OS.

Actualités marquantes

En ce début d'année 2004, plusieurs faits ont marqué l'actualité Open Source. D'abord, il y a les tous derniers chiffres publiés par Netcraft, cette organisation britannique qui tient à jour des statistiques détaillées au sujet de tous les services reliés à Internet. Ces chiffres montrent clairement que Microsoft a cédé beaucoup de terrain concernant les plates-formes Web. Apache, logiciel libre, atteint 67,41% de parts de marché contre seulement 21,02% pour Internet Information Server (IIS), logiciel propriétaire. Autrement dit, le quatuor Linux-Apache-MySQL-PHP (baptisé LAMP) est définitivement gagnant.

Ensuite, le logiciel libre GNU a été incorporé à la liste des "Amis des trésors du monde". Il s'agit d'une opération lancée par la fédération mondiale des clubs UNESCO, sous les auspices de l'UNESCO et de la Convention "pour la protection du Patrimoine Mondial Culturel et Naturel". En clair l'UNESCO reconnaît les valeurs véhiculées par le Logiciel Libre.

Enfin, après IBM, HP, Siemens, Novell a investi "massivement" dans Linux : après le rachat de Ximian, c'est SuSE qui est passé dans le giron de l'éditeur de logiciels de l'état des Mormons.

Mais pour les utilisateurs et développeurs Windows, la grande question reste la même : existe-t-il un équivalent de mes logiciels sous Linux ?

Logiciels libres sous Linux et sous Windows

Ne faites pas d'amalgame entre Linux et Logiciels libres. Apache tourne aussi bien sous les deux plates-formes, et ce n'est qu'un exemple. Néanmoins, il peut arriver qu'une version soit sous licence GPL pour Linux, tandis qu'elle sera sous licence commerciale pour Windows. C'est le cas du moteur de recherche mngosearch (<http://search.mnogo.ru/download.html>) qui remplacera intelligemment la fonction "Index Server" de Windows. Entre parenthèses, ce logiciel est aussi intéressant, car il peut indexer (avec "catdoc" ou "antiword") des documents word, tout comme Index server. Un développeur l'utilisera pour indexer toutes ses documentations et même ses sources s'il programme en C. En effet il pourra exécuter C2HTML afin de transformer le code source en document HTML qui à son tour sera indexé (<http://www.geocities.com/shudderfix/c2html/>).

Mais, avant de nous attaquer au développement proprement dit, réalisons d'abord un bref tour d'horizon des équivalences sous forme d'un tableau d'un environnement de travail "classique".

Windows	Linux	
Navigateurs Web (Internet Explorer)	KDE Konqueror	http://www.konqueror.org/
	GNOME Epiphany	http://www.gnome.org/projects/epiphany/
	Firebird (Linux, Windows et Mac OS X)	http://www.mozilla.org/projects/firebird/
OutLook Express	Evolution	http://ximian.com/products/evolution/
Carnet d'adresses	Rubrica	http://digilander.libero.it/nfragale/index.html
Lecteur de News	Knode	http://knode.sourceforge.net/
Logiciels d'aide au téléchargement du style GetRight sous Windows	GetLeft	http://personal1.iddeo.es/andresgarci/getleft/english/download.html
	Wget (en ligne de commande et disponible avec n'importe quelle distribution et même sous Windows)	http://www.gnu.org/software/wget/wget.html
	Khttrack	http://www.nongnu.org/khttrack/
Client FTP graphique du style CuteFTP sous Windows	Tkftp	http://tkftp.firebird.cx/
Messagerie instantanée	Gaim (MSN, ICQ, IRC, JABBER)	http://gaim.sourceforge.net/
P2P	Mldonkey (Donkey, Souseek, Fasttrack, Gnutella, Open Napster, Direct Connect, BitTorrent)	http://www.freesoftware.fsf.org/mldonkey/
Equivalent de ZoneAlarm ou ICF	Guarddog	http://www.simonzone.com/software/guarddog/
Explorateur de fichiers (équivalent de l'explorateur de Windows)	Endeavour2	http://wolfpack.twu.net/Endeavour2
Compression des données (zip)	Karchiver	http://perso.wanadoo.fr/coquelle/karchiver/index_fr.html

Ceci n'est qu'une sélection de quelques outils courants, le développeur trouvera aussi son bonheur en matière de gravure CD/DVDs, de multimédia en général (divx, MP3, etc.), de bureautique, d'antivirus, de logiciels Anti-Spam et d'utilitaires de sauvegarde.

Les éditeurs

Les programmeurs ont chacun leurs éditeurs fétiches. Certains sont ciblés spécialement pour un langage et dans le cas par exemple de pro-

grammation de scripts, un éditeur est souvent plus adapté qu'un environnement complet de développement. Xemacs a l'avantage de tourner sous Windows et Linux (<http://www.xemacs.org/>), tout comme Vim (<http://www.vim.org/>), SciTE (<http://www.scintilla.org/SciTE.html>) ou jed (<http://space.mit.edu/%7Edavis/jedsoft/jjed/>).

Quelques langages et environnements de développement

Passons en revue quelques-uns des plus connus :

Windows	Linux	
Delphi	Kylix	http://www.borland.com/
	Lazarus	http://www.lazarus.freepascal.org/
Pascal	Freepascal	http://www.freepascal.org/
	Gnu Pascal	http://www.gnu-pascal.de/
IDE Java	Jbuilder	http://www.borland.com/jbuilder/personal/index.html
	NetBeans	http://www.netbeans.org/
	Eclipse	http://www.eclipse.org/
	Forte	http://www.sun.com/software/sundev/index.html
	Vide	http://www.objectcentral.com/vide.htm
IDE Visual Basic	Phoenix	http://www.janus-software.com/
	Kbasic	http://www.kbasic.org/l/home.php3
Visual Basic, C# et surtout ASP.NET	Mono	http://www.go-mono.com/
Le C++ est gâté : il n'y a que l'embarras du choix	Dev-C++	http://www.bloodshed.net/devcpp.html
	Gbuilder	http://gbuilder.sourceforge.net/
Python	Boa-Constructor	http://boa-constructor.sourceforge.net/
Prolog	Gnu Prolog	http://pauillac.inria.fr/~diaz/gnu-prolog/
Perl et TCL	Les outils d'activestate (et komodo)	http://www.activestate.com/
Oracle	TORA (un IDE de développement pour les bases de données Oracle (procédures stockées PL/SQL))	http://www.globecom.se/tora
XBase (Visual Foxpro)	CLIP	http://www.itk.ru/index.shtml
	Le projet Harbour	http://www.harbour-project.org/
Développement de solutions de e-commerce	OS commerce	http://www.oscommerce.com/
	Jonas	http://jonas.objectweb.org/
	Jboss	http://www.jboss.org/
	Cocoon	http://cocoon.apache.org/2.0/

Notez que les environnements graphiques multi plates-formes les plus populaires à ce jour sont sans aucun doute wxWindows (<http://www.wxwindows.org/>), TK (<http://www.tcl.tk/>) et QT (<http://www.trolltech.com/>).

Quelques outils de gestion de projets

De nombreux outils Open Source écrits en Java (Windows et Linux) existent dans le rayon du Knowledge Management (comme KnowledgeTree, à l'adresse <http://www.jamwarehouse.com/products.asp>). D'autres sec-

teurs plus verticaux sont déjà dotés d'outils spécifiques à leurs métiers. En voici quelques-uns :

Windows	Linux	
Méthodologie UML Umbrello		http://uml.sourceforge.net/
	Argo UML	http://www.argouml.org/
Système CASE	Rational Rose (Commercial)	
Microsoft Project	Monsieur Project	http://mrproject.codefactory.se/
Visio	Kvio	http://www.koffice.org/
	Dia	http://www.lysator.liu.se/~alla/dia/
MatLab	EULER	http://euler.sf.net/
	Yorick	http://wuarchive.wustl.edu/languages/yorick/doc/
Autocad	Varkon	http://www.tech.oru.se/cad/varkon/
	ThanCad	http://thancad.sourceforge.net/
	PythonCad	http://www.pythoncad.org/

Ce que vous avez manqué

À la date du 20 janvier 2004, freshmeat.net recensait à l'intitulé "Software Development" un total de 7690 projets... 42 projets concernant, par exemple, le désassemblage (ce qui est en quelque sorte un non-sens pour des outils Open Source qui sont fournis la plupart du temps avec leurs codes sources). Cette rubrique "développement de logiciels" n'est pas non plus exhaustive, car tout dépend de son critère de classement. En effet, sachez par exemple que la rubrique base de données à elle seule comprend 1489 projets.

Thème	Nombre de projets
Assemblers	44
Bug Tracking	104
Build Tools	276
Code Generators	347
Compilers	305
Debuggers	182
Disassemblers	42
Documentation	173
Embedded Systems	131
Internationalization	31
Interpreters	302
Libraries	3445
Localization	22
Object Brokering	105
Pre-processors	38
Quality Assurance	181
Testing	297
User Interfaces	203
Version Control	221
Widget Sets	185
TOTAL	7690

Conclusion pour les développeurs

Avant de choisir un outil commercial mono OS, posez-vous la question de savoir si un produit Open Source tournant sous plusieurs plates-formes existe. Il est logique que les produits libres soient en général plus portables, vu qu'ils sont livrés avec leurs codes sources. Par conséquent, vous y gagnerez en portabilité, en maintenance du produit et en argent sonnante et trébuchante. ■ Xavier Leclercq

Logiciels libres : une offre logicielle en retard ?

Ces dernières années, les logiciels libres ont beaucoup progressé. Jadis limités à l'univers des serveurs d'infrastructure et des principaux services Internet (serveurs web, mail, DNS, ...), ils se sont propulsés en peu de temps jusqu'aux portes des postes clients avec des applications de type "bureautique".



Que l'offre des logiciels libres soit de plus en plus complète est un fait. Mais la profusion d'articles et d'études spécialisées les concernant prouve encore une certaine filiosité à choisir une solution de ce type pour des projets de plus en plus critiques. Cette crainte s'explique en partie par le niveau de packaging des logiciels commerciaux et les efforts marketing et commerciaux qui leur sont associés. On pourrait également citer la croyance tenace d'une absence de support liée aux logiciels libres. Il devient alors légitime de se demander comment se placent les deux offres l'une par rapport à l'autre.

Comparaison possible ?

Un premier point à souligner est qu'il est assez inapproprié de comparer tel ou tel logiciel commercial à son "équivalent" libre. En effet, deux philosophies très distinctes s'opposent :

- Un logiciel commercial évolue fréquemment par ajouts successifs de fonctionnalités connexes. Ainsi, IIS, le serveur web de Microsoft, inclut également des fonctionnalités de serveurs FTP et bien d'autres.

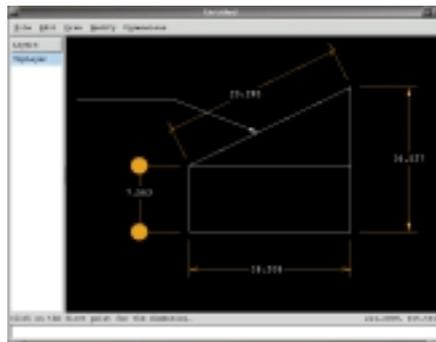
- Un logiciel libre, pour sa part, tend à se restreindre à son champ d'application et à exceller dans son domaine. Même s'il améliore son support de telle ou telle norme édictée par le W3C, Apache reste strictement un serveur web. Cette différence fondamentale rend plus difficile tout benchmark et brouille la visibilité des clients : un ISP pourra déployer une offre complète avec seulement quelques briques propriétaires, mais probablement trois fois plus de logiciels libres pour un même niveau de service.

Malgré cela, il est possible de distinguer deux familles d'offres logicielles, pour lesquelles les différences sont notables :

- L'environnement "serveur", au sens large, tel que cité plus haut, dans lesquels les logiciels libres ont les mêmes fonctionnalités que les logiciels commerciaux. Ils sont souvent la

référence, l'état de l'art dans leurs domaines respectifs, grâce à un strict respect des standards. Ceci s'explique aisément par le mode de développement de ces outils : pour une équipe de développement répartie internationalement, où les décisions sont prises essentiellement de façon collégiale (avec une prépondérance pour l'avis du "leader developer" bien sûr), les standards représentent un lien évident sur lequel tous peuvent s'accorder.

- Le monde des applications clientes et de gestion. Dans ce cas, les logiciels libres accusent



souvent un important retard vis-à-vis des offres commerciales. Cela est également dû à leur nature : ils sont initiés par des développeurs pour combler des besoins qui leur sont propres. Besoins qui ne rejoignent pas nécessairement ceux des utilisateurs finaux.

Néanmoins, l'offre libre se structure de plus en plus pour répondre à ces manques, que ce soit par le reversement d'outils propriétaires qui connaissent un essor très rapide (cas du logiciel Opengroupware.org, devenu un logiciel libre, à l'initiative de son éditeur Skyrix mi 2003, et qui est l'un des plus prometteurs outils de groupware actuels), ou par la création d'entreprises, groupes d'intérêts ou consortium qui développent une brique particulière (cas de l'environnement de bureau KDE qui atteint un haut niveau de convivialité et d'intégration).

Comment comparer ?

Quel que soit le domaine dans lequel on souhaite comparer une gamme de logiciels libres

à un ou plusieurs logiciels commerciaux, il est important de retenir les étapes suivantes :

- Définition précise des besoins des utilisateurs, administrateurs, direction, ... et rédaction d'un cahier des charges ou de spécifications aussi détaillées que possible,

- Réalisation d'un audit détaillé de l'existant, pour distinguer les offres phares correspondant à l'état de l'art à un instant donné,

- Comparaison des logiciels par observation des sites web, des forums sur les-

quels les utilisateurs donnent leur opinion/ Il convient aussi de porter une attention toute particulière à la présence d'interface d'administration, ainsi qu'à la qualité de la documentation associée,

- Mise en place d'une maquette validant les fonctionnalités et les comparant aux besoins exprimés,

- Déployer la solution en production,
- Chercher un support adapté à ses besoins,
- Réaliser des transferts de compétences opérationnels sur les solutions déployées.

Pour tout cela il est important, dans l'univers des logiciels libres, de se faire accompagner de compétences reconnues pour mener à bien toutes ces étapes, spécialement pour des projets de grande ampleur. Ceci est tout particulièrement vrai pour le support, jugé à tort comme le point faible des offres libres.

Exemple

Pour illustrer ce propos, prenons l'exemple du marché des serveurs d'applications J2EE. LINAGORA (SLL), travaille actuellement sur un projet pour une grande administration concernant la fourniture et la maintenance d'un socle J2EE pour l'ensemble des applications qui seront déployées dans les cinq prochaines années. Cette administration a le choix entre des offres éditeurs (sous la forme de licences site) ou une solution à base de logiciels libres. Un élément fort de ce choix pour les commanditaires du projet reste le respect des normes et des



standards, domaine où le logiciel libre devance ses potentiels concurrents propriétaires.

L'offre libre proposée est composée des logiciels suivants : Apache, Tomcat et JBoss qui sont complétés par un ensemble de logiciels libres pour assurer l'administration et la supervision de la plate-forme (par exemple Nagios pour le contrôle et la gestion des applications déployées). Pour assurer la maintenance de cet " agglomérat logiciel ", LINAGORA propose son offre de Tierce Maintenance Logiciels Libres (TM2L), une TMA (Tierce Maintenance Applicative) appliquée à des briques Open Source. Cette TM2L vise à maintenir dans le temps la plate-forme serveur d'applications comme le ferait un éditeur de logiciels. En effet, compte tenu du cadre juridique des Communautés qui développent ces logiciels libres, c'est LINAGORA qui prend les engagements sur le support et la maintenance.

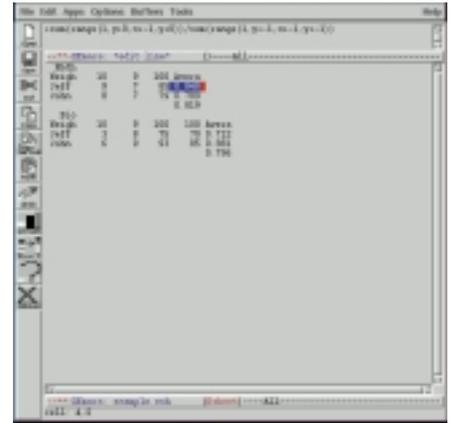
Un des enjeux de cette prestation est la communication avec les différentes Communautés en charge du développement des projets utili-

sés. Le but premier est de limiter les divergences entre le projet communautaire et le projet maintenu pour le client dans le cadre de la TM2L.

Le tableau ci-dessous résume les principaux éléments de comparaison entre les serveurs d'applications éditeur et la solution libre :

Éléments de comparaison	Offres logiciels éditeur	Offre logiciels libres
Respect des normes et des standards	4/5	5/5
Coûts d'acquisition	2/5	4/5
Coûts de maintenance	4/5	3/5
Performances	3/5	3/5
Exploitabilité	4/5	3/5

En conclusion, même si l'offre des logiciels libres est fondamentalement différente de celle des logiciels commerciaux, par leur nature, celles-ci sont très proches en terme de fonctionnalités. Pour distinguer le bon assemblage de briques libres qui répond au mieux à un besoin, il faut s'appuyer sur des professionnels qui, par leur mode de fonctionnement, leur



lien privilégié avec les communautés, leur perpétuelle activité de recherche et développement, sont les mieux placés pour réaliser un état de l'art pertinent, et en assurer le suivi au long terme.

■ **Michel Marie MAUDET,**
Directeur des opérations chez Linagora

■ **Laurent PIERRE**
Directeur de projets chez Linagora



L'intégration de Linux dans les applications commerciales

Etude de cas : MusicMagic

Chercher par titre, artiste ou album. Lancer cette sélection ou demander au système de trouver d'autres morceaux de même style et créer une compilation que vous pouvez personnaliser par genre, rythme, ou degré de variété. Acheter le CD, télécharger cette musique sur votre lecteur MP3 ou utiliser le

système comme une station de radio privée... Avec MusicMagic, les possibilités sont illimitées. Vous ne pensez pas qu'un système de gestion des préférences automatisé aussi complexe puisse avoir été développé avec le système d'exploitation Linux ? Et pourtant, c'est exactement ce que nous avons fait. Linux nous apporte une plate-forme de développement parfaitement adaptée à MusicMagic.

Linux à l'époque... la méfiance

Lorsqu'est venu le moment de lancer le développement de MusicMagic, la querelle entre Windows et Linux avait atteint son paroxysme et pouvait difficilement être ignorée. Predixis avait besoin d'un système d'exploitation qui puisse exécuter des services web de façon fiable et sans que l'on doive consacrer beaucoup de temps et d'argent à son adaptation et

à son déploiement. Le choix s'était réduit à trois types d'O.S viables. Nous pouvions sélectionner une plate-forme commerciale Unix, mais cela aurait été très cher. Microsoft proposait toute une variété de saveurs Windows, mais nous doutions qu'elles soient assez fiables dans une application commerciale de services web, et nous ne pouvions pas nous permettre de payer quelqu'un pour redémarquer un O.S Windows à chaque fois qu'il planterait. Enfin, il y avait Linux, dénigré et pas encore reconnu, mais disponible et bon marché. En tant que startup, Predixis n'était pas dans une situation financière lui permettant d'acquérir des licences propriétaires Unix. Linux n'était pas cher, mais est-ce qu'il ferait l'affaire ? La campagne de Microsoft contre Linux a été le déclic. La décision a été relativement facile à prendre : peu cher, peu reconnu et haï de Microsoft, Linux a été notre premier et notre dernier choix.



Linux dans l'architecture de MusicMagic

MusicMagic est une application distribuée multicouche qui utilise les services web pour la communication entre ses différents éléments : une base de données, un moteur de gestion des préférences et un module interface utilisateur. Son architecture est aussi moderne que les standards de développement actuels. Nous avons choisi les services web et plus précisément SOAP comme protocole de communication entre ces modules, parce que cela nous permettait d'isoler la fonction des différentes parties du système tout en ayant une architecture très simple et élégante. L'accent mis sur l'élégance et la simplicité est très important. Du fait que MusicMagic est une technologie d'avant-garde, nous avons constamment rencontré de nouveaux défis pendant sa création ; je n'exagère pas en affirmant que MusicMagic s'est révélé être un projet vivant, changeant constamment de forme pendant son développement. Conséquence, il était vital que l'architecture reste aussi simple que possible pour permettre aux différents éléments du système d'évoluer et grossir de façon indépendante. Si nous ne pouvions pas respecter cette condition, alors le système entier deviendrait trop lourd pour supporter un développement et des améliorations continuelles.



Donc, distribution et simplicité étaient des conditions nécessaires depuis le début du développement et SOAP nous a permis de les remplir aisément. Les différents modules de MusicMagic ont différentes fonctionnalités et lorsque ces modules changeaient ou évoluaient, ces modifications n'étaient visibles qu'à travers l'interface SOAP, que nous pouvions contrôler. SOAP assure une bonne communication entre les modules qui tournent sur différentes machines, à tel point que l'échange d'informations entre les modules n'est jamais devenu – et n'a jamais introduit le risque d'un goulot d'étranglement dans le système. Le véritable goulot d'étranglement était en fait les conditions nécessaires à l'analyse de la musique pour l'enrichissement de la base de données.

Puisque SOAP garantit une communication efficace entre les modules, nous avons pu nous concentrer sur la performance et les fonctionnalités du système dans son ensemble, et créer une architecture qui serait simple, efficace et facile à utiliser, sans se préoccuper de la façon dont les modules communiqueraient. Le schéma suivant représente l'architecture de base que nous avons développée. Comme mentionné plus tôt, les principaux éléments sont une base de données, un moteur de gestion des préférences et une interface utilisateur. La base de données – ou les bases de données, puisque plusieurs peuvent être exploitées par ce système – et le moteur de gestion des préférences sont compatibles avec SQL (en utilisant les API JDBC ou ODBC). Cette partie du système MusicMagic tourne sur Linux.

La base de données est une base relationnelle contenant des informations (données) extraites de morceaux de musique analysés et qui définissent ce que contiendra vraiment la DB. Lorsque l'on entre les goûts d'un utilisateur concernant certains types de musique, par l'intermédiaire du module d'interface utilisateur au moteur de gestion des préférences, celui-ci est capable de trouver dans la base, la musique qui correspond aux caractéristiques données. Ce moteur est en fait une série de serveurs Apache et Tomcat qui fonctionnent comme un serveur de services Web intégré. Le module interface utilisateur est une interface entre de multiples utilisateurs et le système MusicMagic.

Le fonctionnement du système est aussi simple que son architecture. Un utilisateur

donne à l'interface ses préférences musicales par l'intermédiaire d'un petit questionnaire, sélectionne ensuite un artiste ou une chanson. La requête utilisateur est transférée au moteur de gestion des préférences, qui lance à son tour une requête à la base de données. La base de données renvoie la demande avec une liste de titres similaires au moteur. Ensuite, le moteur livre l'information à l'utilisateur via l'interface. Les connexions entre la base de données et le moteur, puis entre le moteur et l'interface se font toutes par messages SOAP. Linux est utilisé pour faire fonctionner le cœur du système, à savoir, le tandem efficace base de données / moteur de gestion des préférences.

La connexion entre l'utilisateur et l'interface peut avoir lieu par l'intermédiaire d'un kiosque en ligne, un navigateur HTML, un PC sous Windows, un organisateur sous Linux...il n'y a pas de limites puisque SOAP assure des connexions simples, standardisées et compatibles avec le reste du système.

Linux fait fonctionner les deux types de services Web (sans fil et connexion directe) efficacement, car il ne demande que très peu de ressources système pour maintenir la base de données et le serveur de services Web actifs. C'est pour cela que les requêtes à la base de données peuvent être gérées avec un délai très court du point de vue du système. Linux rend ces systèmes très flexibles en permettant la multiplication du nombre de bases de données, clients ou vendeurs en fonction des besoins.

Finalement, ce n'est ni l'architecture du système, ni la communication entre les modules, ni la gestion des ressources du système qui représentait un goulot d'étranglement pendant le développement de MusicMagic, mais les fonctions de traitement de la musique. Pour enrichir la base de données, il a fallu en effet développer un système d'analyse acoustique en utilisant des formats de compression MP3 pour nous permettre de créer un spectre de variables tellement large que le système pourrait répondre lui-même à n'importe quelle requête intelligemment. La quantité des variables utilisées et le temps

nécessaire au remplissage de la base de données pour qu'elle soit utilisable ont constitué la plus grande partie du temps de développement de MusicMagic.

Quel avenir ? une conclusion prudente

Même si Linux est écrit par des programmeurs pour des programmeurs, son avenir dépend de sa capacité à supporter des applications commerciales de façon fiable. Un courant dangereux est apparu au sein de la communauté Linux soutenant que les applications commerciales ne sont plus ou moins que des "jouets" dont le développement est un "exercice pour étudiants". Cela peut paraître snob de la part du cœur des adeptes Linux, qui préfèrent peut-être voir le gros potentiel de Linux rester caché et inutilisé entre les murs de la commu-



nauté de développement. Une telle insularité pourrait coûter cher à Linux dans sa course pour rester compétitif. Linux doit continuer à évoluer sur le terrain des applications commerciales et permettre à d'autres applications, comme MusicMagic de fonctionner. Enfin, c'est seulement lorsque Linux aura prouvé son potentiel dans le commerce électronique qu'il sera devenu une vraie force dans la communauté IT.

■ Dr Adam Kolowa
Fondateur de Parasoft



Licence : vrai problème ou fausse idée ?

Dans les critiques ou les points négatifs pour l'utilisation de l'Open Source ou du Libre, la licence et même plutôt les licences arrivent souvent en tête. Il faut bien reconnaître que tout n'a pas été résolu. Mais, la situation n'est pas aussi problématique que cela...

Comme un bonheur ne vient jamais seul dans l'Open Source, nous avons à faire à un troupeau de licences, une bonne vingtaine, mais dont seulement une poignée sort du lot. Ce grand nombre de licences comporte donc des clauses différentes. Surtout, les licences ne sont pas toujours compatibles entre elles. Chaque licence comporte donc des clauses, comme pour les licences de logiciels propriétaires. La plus connue, et la plus usitée, est la licence GPL. Prenons le cas de la GPL. Par définition, si on utilise un code GPL dans son application et que l'on souhaite la diffusion, il y a nécessité d'utiliser la licence GPL pour l'ensemble du code source ! Cela interdit donc de vendre des logiciels utilisant du code GPL à l'intérieur. Cependant, si le code GPL est utilisé dans un projet uniquement interne à l'entreprise, là, il n'y a aucun souci.

Diversité garantie

Pour pallier à ce problème de licencier le code entièrement en GPL, il existe une autre licence plus souple, la LGPL. Cette licence permet d'utiliser des bibliothèques Open Source. Si une application commerciale utilise une bibliothèque LGPL (sans modifier le code source), le code de l'application demeure propriétaire, on ne fait qu'appeler une bibliothèque Open Source. Donc, vous l'aurez compris, la compatibilité entre les licences (et les différences des clauses) est un élément important à bien maîtriser. Les non-experts risquent de s'y perdre très rapidement. Il est facile de violer une clause de la licence.

GPL organise en quelque sorte une certaine idée de l'Open Source. D'autres licences sont



bien plus " libérales ". La plus connue est sans aucun doute la licence BSD. Elle autorise à peu près tout : modifier le code, mettre du code propriétaire, restreindre la licence du logiciel final. Les contraintes sont souvent peu nombreuses. Dans la même idée, il y a aussi la licence Apache. Elle a la particularité d'avoir une clause de publicité. Ainsi, si je crée un logiciel incluant un projet Apache, je dois citer dans la documentation, " Ce logiciel inclut un logiciel développé par la Apache Software Foundation ". La licence Apache est supposée être incompatible avec la licence GPL.

Légale ou pas légale la licence ?

En Open Source, la licence mentionne l'absence de garantie sur le logiciel, la qualité et les performances. Bref, l'utilisateur utilise un code Open Source à ses risques et périls. Une entreprise implémentant un outil Open Source dans son environnement prend donc la responsabilité de cette utilisation. On lit parfois que la licence GPL, à cause de sa non-responsabilité, n'est pas valable en France, donc illégale. Cependant, la licence GPL n'est pas incompatible avec le droit français. Elle est parfaitement utilisable. Malgré tout, nous ne saurons que vous conseiller d'étudier la jurisprudence concernant les clauses dites abusives. Mais les pro-

blèmes de responsabilité (et donc de garantie) concernent aussi bien les logiciels propriétaires qu'Open Source. Ce qui peut gêner, c'est l'absence d'un interlocuteur en face de soi. Même si avoir un éditeur ou une SSII Libre n'offre pas plus de garanties ou de responsabilités, cela rassure d'avoir une entité palpable. C'est pour cela que les entreprises préfèrent souvent utiliser de l'Open Source proposé / vendu pour des sociétés. Cependant, il semble que la polémique ne soit pas près de s'éteindre sur le problème de conformité de la licence GPL.

Après ce rapide tour d'horizon, que peut-on en conclure ? L'Open Source n'est pas incompatible avec un projet d'entreprise interne. Pour ce qui est des outils commerciaux utilisant de l'Open Source, là, il est vrai, il faut d'abord procéder à une évaluation des licences des codes libres utilisés. Les problèmes de responsabilité et de garantie sont identiques aux logiciels ouverts et fermés. Bref, le problème de licence, lorsqu'il est bien compris et mesuré, n'est pas un obstacle. Dans les mois à venir, il faudra tout de même suivre l'évolution de l'affaire SCO dont la menace, même si elle est virtuelle, sur Linux n'en est pas moins une réalité. À cela, s'ajoutera le brevet logiciel qui n'est toujours pas réglé au niveau européen.

Pour aller plus loin :

Garanties et responsabilités dans les logiciels libres de Maître Valérie Sédaillan, www.internet-juridique.net

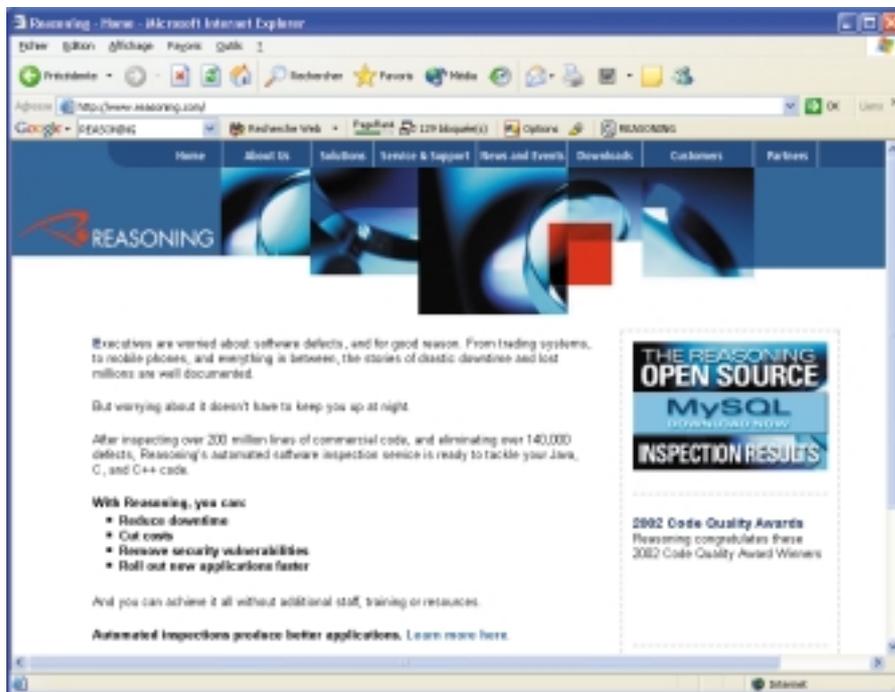
■ François Tonic



Logiciels Open Source : champions en audit du code source

Qui a dit qu'Open Source ne rimait pas avec Qualité du Code ? En tout cas, la société Reasoning qui est prestataire de service en contrôle automatisé de qualité du logiciel (ASI : Automated Software Inspection) a jeté un pavé dans la mare à ce sujet.

En résumé le code de MySQL 4.0.16 a été passé à la moulinette de leur contrôle qualité, ce qui au final a débouché sur la découverte de 21 lignes de codes "défectueuses" sur un total de 236.000 lignes. La densité d'erreur est par conséquent de 0,09 défauts pour mille lignes. Par contre, la densité d'erreur émanant de logiciels commerciaux est en moyenne de 0,57 défauts pour mille lignes. Ce chiffre est basé sur le contrôle de 35 millions de lignes de codes. En généralisant, on peut tirer la conclusion qu'un code de MySQL est infiniment plus robuste qu'un code commercial. Mieux, David Axmark, cofondateur et Vice-président de MySQL, contacté à ce sujet par Reasoning a déclaré que les erreurs trouvées seraient corrigées dans la semaine (ce qui a été fait : l'étude de Reasoning datant de mi-décembre 2003). En fait, c'est logique. Confronté à un logiciel propriétaire non livré avec son code source, l'utilisateur final ne peut que se contenter de signaler une erreur, et de prier pour que celle-ci soit rapidement corrigée (...). Avec un code ouvert, l'utilisateur peut aller plus loin et "descendre voir" dans le code où se situe l'erreur, la corriger pour son propre compte et par la suite en faire bénéficier l'ensemble des utilisateurs. Il endosse la double casquette de testeur final et de développeur occasionnel. Si l'erreur est bloquante et particulièrement gênante pour lui, il essaiera par tous les moyens de la corriger. En outre, avec un code Open Source, si une amélioration, une extension, ou une correction d'erreur est réalisée c'est très souvent par un spécialiste du domaine. Par contre, il est courant que lors de l'élaboration initiale d'un logiciel commercial le chef de projet fasse appel à un expert externe. Celui-ci n'aura pas la même disponibilité tout au long de la vie de l'application. Le développeur



commercial à qui sera signalée l'erreur à un moment donné ne sera pas donc forcément compétent dans le domaine où il doit intervenir. Pire, le développeur Open Source n'est pas nécessairement tenu par des contraintes budgétaires, de temps, ou par des règles rigides. Il prendra le temps, sera innovant et "soignera" sa rustine. Tandis que le développeur commercial est souvent payé pour faire au mieux et au plus vite.

Tout ceci est un peu caricatural. Il ne faut pas non plus généraliser, car de très bons pro-

grammeurs on en retrouve évidemment dans les deux mondes. Mais un fait est tenu pour certain : si vous disposez du code source, vous pouvez l'auditer. Si vous n'en bénéficiez pas, vous ne pouvez pas juger de sa qualité intrinsèque. C'est malheureusement aussi simple que cela.

Des logiciels d'audits à la rescousse

Un développeur C peut toujours détecter dans son code des appels aux fonctions `gets()`,

Quelques liens supplémentaires d'utilitaires de type 'audit' du code source :

Flawfinder : <http://www.dwheeler.com/flawfinder/>

RATS : http://www.securesw.com/download_form_rats.htm

ITS4 : <http://www.danielnaber.de/its4/>

PScan : <http://www.striker.ottawa.on.ca/~aland/pscan/>

BOON Buffer Overrun detection : <http://www.cs.berkeley.edu/~daw/boon/>

MOPS (MOdelchecking Programs for Security properties) : <http://www.cs.berkeley.edu/~daw/mops/>

Cqual : <http://www.cs.umd.edu/~jfoster/cqual/>

ESC/Java (Extended Static Checking for Java) : <http://www.cs.kun.nl/ita/research/projects/sos/projects/escjava.html>

JCAVE (JavaCard Applet Verification Environment) : <http://www.sics.se/fdt/projects/vericode/jcave.html>

checker ver	file	fn	startl	endl	srcfn	reason	type
NULL 2.4.1	drivers/acpi/interpreter/amutils.c	acpi_aml_build_copy_internal_package_object	468	472	_cm_create_internal_object	Using unknown ptr "this_dest_obj" illegally! set by '_cm_create_internal_object'	BUG
NULL 2.4.1	drivers/ieee1394/video1394.c	alloc_dma_iso_ctx	304	305	kmalloc	Passing unknown ptr "d" as arg 0 to call "memset" set by 'kmalloc'	BUG
NULL 2.4.1	drivers/scsi/AM53C974.c	AM53C974_init	681	683	scsi_register	Using unknown ptr "instance" illegally! set by 'scsi_register'	BUG

`scanf()`, `sprintf()`, `strcat()`, `strcpy()` qui sont susceptibles de déboucher sur un dépassement de tampon. De même, les fonctions `printf()`, `fprintf()`, `vprintf()`, `snprintf()`, `vsnprintf()` présentent des pièges dans le formatage des chaînes. Et au point de vue des failles de sécurité s'ajoutent les fonctions `exec()`, `popen()`, `system()`, `ainsi qu'access()`, `chown()`, `chgrp()`, `chmod()`, `mktemp()`, `tempnam()`, `tmpfile()`, `tmpnam()` (ce que l'on nomme en anglais les "Race conditions"). L'idéal est de bien coder, mais le développeur est un être humain et n'est pas infallible, d'où le recours à un logiciel d'audit externe.

Citons en vrac : le projet MC, le projet SLAM et une série d'utilitaires comme pscan, Flawfinder, RATS, etc.

Un exemple classique : SPLINT

Il y a plus de 30 ans (...) S.C. Johnson a conçu un outil de débogage baptisé Lint. C'était avant l'avènement du C ANSI et sous UNIX. John Guttag et Jim Horning ont adapté l'outil qui a été nommé LCLint ; dont l'évolution actuelle se nomme Splint (Secure Programming Lint) disponible à l'adresse <http://www.splint.org/>.

Ce qui donne sous debian :

```
#apt-get install splint
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
 splint
0 upgraded, 1 newly installed, 0 to remove
and 0 not upgraded.
Need to get 808kB of archives.
After unpacking 2728kB of additional disk
space will be used.
Get:1 http://ftp.easynet.be unstable/main
splint 3.1.1-3 [808kB]
Fetched 808kB in 7s (111kB/s)
Selecting previously deselected package splint.
(Reading database ... 48086 files and directories
currently installed.)
Unpacking splint (from .../splint_3.1.1-3_i386.deb) ...
```

Setting up splint (3.1.1-3) ...

La dernière version est la 3.1.1.

→ **ATTENTION** : Des binaires pour Windows sont aussi disponibles sur le site.

Splint détecte des erreurs non identifiables a priori avec GCC. De nombreux programmeurs se contentent des messages d'erreurs générés par GCC et c'est un tort.

Prenons le cas de ce code source C suivant ("test.c") :

```
#include <stdlib.h>
#include <stdio.h>
int main()
{
    int *p = malloc(5*sizeof(int));
    *p = 4;
    free(p);
    return 0;
}
```

Le problème se situe au niveau de l'allocation-mémoire : si jamais le pointeur retourné est null, c'est une catastrophe pour la ligne suivante d'assignation !

Splint mettra le programmeur en garde contre cette pratique :

```
#splint test.c
Splint 3.1.1 - 17 Sep 2003

test.c: (in function main)
test.c:6:10: Dereference of possibly null pointer
p: *p
A possibly null pointer is dereferenced. Value
is either the result of a function which may
return null (in which case, code should check
it is not null), or a global, parameter or structure
field declared with the null qualifier. (Use
-nulldef to inhibit warning)
test.c:5:18: Storage p may become null
```

Finished checking - 1 code warning

Pour que splint accepte le code il faudra tester le retour de fonction :

```
test2.c :
#include <stdlib.h>
#include <stdio.h>
int main()
{
    int *p = malloc(5*sizeof(int));
    if (p == NULL) {
        fprintf(stderr, "erreur d'allocation
mémoire");
        exit(EXIT_FAILURE);
    } else *p = 4;
    free(p);
    return 0;
}
```

Ce qui donne :

```
#splint test2.c
Splint 3.1.1 - 17 Sep 2003

Finished checking - no warnings
```

Le projet MC Open Source

MC signifie "Meta-niveau de compilation". Il a

vu le jour à l'université de Stanford, dans le but de repérer plus facilement des erreurs dans les systèmes logiciels. Il consiste en un compilateur xgcc qui autorise l'insertion des tests d'analyses du code. Pour l'instant, ce projet est toujours en développement, mais il a été mis à la disposition du public une interface Web. Celle-ci interroge une base de données contenant les erreurs découvertes à ce jour par le compilateur xgcc sur le code source de différentes versions du noyau. Vous pouvez découvrir ceci en surfant à l'adresse : <http://meta-comp.stanford.edu/linux/list.php3>.



Sriram K. Rajamani est le développeur principal chez Microsoft du projet SLAM



Par exemple avec MC le noyau 2.4.21 comprend 124 erreurs.

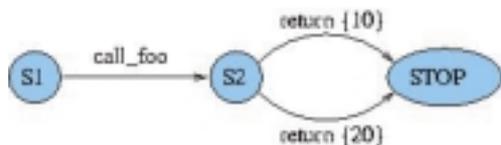
By type :	By version :	By checker :	Total :
type	ver	checker	total
count	count	count	Count
BUG	2.4.1	NULL	total
124	124	124	124

Dont voici les trois premières :

Ce système emploie des "checkers", c'est-à-dire des extensions du compilateur pour tester certains types d'erreurs. Le "checker NULL" par exemple recherche une erreur potentielle de passage de pointeur NULL. "FN" détecte les fonctions dépassant les capacités de la pile du kernel. "BLOCK" vérifie qu'une fonction n'est pas en situation de "deadlock"... etc.

Le projet SLAM de Microsoft

Le projet SLAM est une initiative du très fructueux centre de recherche de Microsoft (<http://research.microsoft.com/slam>). Il utilise une méthode sophistiquée d'analyse statis-



tique pour déterminer les points faibles d'un code. Le projet a été testé avec succès pour déterminer les failles de codes des pilotes de Windows XP (test des appels API avec l'outil SDV : Static Driver Verifier).

Moped (Model-Checker for Pushdown Systems) est un logiciel d'audit qui a été écrit par Stefan Schwoon de l'université de Stuttgart (<http://www.fmi.uni-stuttgart.de/szs/tools/moped/>). Il fait appel à SLAM pour réaliser son étude statistique du code source. Ce logiciel fonctionne aussi bien sous Windows (via cygwin) que sous Linux. D'autres logiciels se sont basés sur le SLAM Toolkit. C'est le cas de BLAST (Berkeley Lazy Abstraction Software Verification Tool) de l'Université de Berkeley (<http://www-cad.eecs.berkeley.edu/~rupak/blast/>) mais aussi de MAGIC (Modular Analysis of proGRams In C) dont nous allons discuter plus en détail (<http://www->

2.cs.cmu.edu/~chaki/magic/).

Comment cela fonctionne-t-il ? Nous partons du code source de la fonction suivante :

```

(my_proc.pp) :
int my_proc(int x)
{
    int y;
    if(x == 0) {
        y = foo();
        if(y > 0) return 10;
        else return 20;
    } else {
        y = bar();
        if(y < 0) return 30;
        else return 40;
    }
}
  
```

Puis nous établissons une série de règles d'états :

Si le premier argument à my_proc est égal à zéro...

... alors my_proc appelle "foo".

... Selon ce que retournera "foo", une valeur plus élevée que zéro ou non, my_proc renvoie 10 ou 20.

Sinon si le premier argument n'est pas égal à zéro...

... alors my_proc appelle "bar".

... Selon ce que retournera "bar", une valeur plus élevée que zéro ou non, my_proc renvoie 30 ou 40.

Ce "pseudo code" est alors traduit en spécifications :

```

my_spec.spec:
cprog my_prog = my_proc {
    abstract abs_5, {($1 == 0)}, S1;
    abstract abs_6, {($1 != 0)}, S3;
    abstract abs_7, {($1 == 0)}, S3;
    abstract abs_8, {($1 != 0)}, S1;
}

cproc my_proc {
    abstract { abs_1, ($1 == 0), S1 };
    abstract { abs_2, ($1 != 0), S3 };
}
  
```

S1 = (call_foo -> S2),
 S2 = (return {10} -> STOP | return {20} -> STOP).

S3 = (call_bar -> S4),
 S4 = (return {30} -> STOP | return {40} -> STOP).

```

cproc foo {
    abstract { abs_3, (1), FOO };
}
  
```

```

cproc bar {
    abstract { abs_4, (1), BAR };
}
  
```

```

FOO = ( call_foo -> return {-1} -> STOP ).
BAR = ( call_bar -> return {50} -> STOP ).
  
```

Si tout va bien, Magic devrait pouvoir vérifier avec succès l'abstraction abs_5 (si le premier argument est égal à zéro on appelle foo) et produire un résultat comme ceci :

```

#magic -abstraction abs_5 my_proc.pp my_
spec.spec -optPred
conformance relation exists !!
abstraction abs_5 is valid ...
Simplify process destroyed ...
terminating normally ...
  
```

De même, vous pouvez essayer de vérifier que l'abstraction abs_7 est inadmissible (si le premier argument est égal à zéro on appelle bar) :

```

#magic -abstraction abs_7 my_proc.pp my_
spec.spec -optPred

*****
branch ( x == 0 ) : TRUE
#### epsilon_0 ####
y = foo ( )
#### call_foo ####
y = foo ( )
#### { y = -1 } ####
branch ( y > 0 ) : FALSE
#### epsilon_0 ####
return ( 20 )
#### return { ! } ####
*****

conformance relation does not exist !!
abstraction abs_7 is invalid ...
  
```

Conclusion

L'audit logiciel du code source est une activité d'avenir en pleine ébullition. Cette dernière approche par spécifications est intéressante, même si elle réclame pas mal de travail. Mais dans le cas, par exemple, d'un "logiciel embarqué" à bord d'une navette ou d'un avion, ce genre de spécifications n'est pas un luxe...

■ Xavier Leclercq

Le logiciel libre et la sécurité

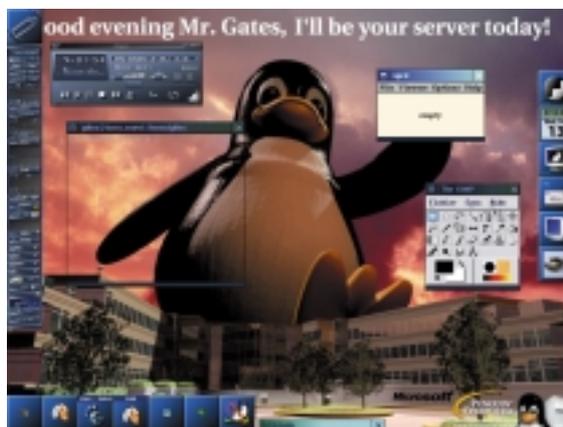
Problématique essentielle des architectures informatiques, aujourd'hui, la sécurité reste encore un domaine assez nébuleux. Pourtant, la sécurité offerte par les logiciels libres est souvent mise en cause lors des processus de décision. Ce point de vue est-il justifié et raisonnable, ou découle-t-il d'une méconnaissance du problème ?

A une question aussi délicate, il n'est pas réaliste d'apporter une réponse sans nuance, validant ou rejetant en bloc la défiance parfois ressentie envers les logiciels libres dans les entreprises. Avant de tenter cependant de répondre à cette question, faisons le point sur les critères à retenir lorsque l'on prétend juger la "sécurité" d'une application. Ce terme doit en effet être pris dans son acception la plus généraliste, a fortiori s'il est question d'un élément central d'un système d'information, pour que la problématique soit entière. Les axes principaux seront donc la sécurité au niveau du réseau (procédures d'identification, gestion des droits d'accès, etc.) et la sécurité au niveau exécution (intégrité des données manipulées, résistance aux montées en charge, etc.).

Pour ce qui concerne le premier critère, la sécurité au niveau réseau, les logiciels libres ont abondamment recours à des mécanismes standardisés, sous forme de bibliothèques, ainsi qu'à des algorithmes de cryptage connus de tous. Les sources étant accessibles à tout un chacun, la tâche consiste à mettre au point des systèmes qui ne reposent pas sur le secret de leur fonctionnement interne, mais sur un schéma sécuritaire ne comportant pas de faille. Ce qui est un gage d'efficacité incontestable, étant entendu, qu'un ordinateur, ainsi que chacun sait, est incapable de garder un secret bien longtemps ! Le partage des sources, et par conséquent des techniques de sécurisation, a permis avec le temps la mise au point d'outils éprouvés et reconnus pour leur efficacité. Et cette mise en commun des techniques employées permet à un plus grand nombre d'en surveiller l'évolution, ou plus justement l'évolution des attaques qui peuvent être déployées face à elles. Cette notion de

masse d'utilisateurs et développeurs est capitale dans le problème de la sécurité, comme nous allons le confirmer en nous intéressant au second critère de sécurité des logiciels : la fiabilité à l'exécution.

Une faille de sécurité sur un réseau peut être grave, à condition qu'elle soit exploitée. En revanche, un bug conduisant à la perte de données sera aussi inévitable que ravageur. C'est pourquoi la fiabilité intrinsèque d'un logiciel est à prendre en compte avec au moins autant d'intérêt que son inexpugnabilité à



l'échelle du réseau. Or un programme, qu'il soit libre ou propriétaire, ne peut que rarement prévoir tous les cas de figures, et de fait, peut souvent contenir des bugs extrêmement difficiles à découvrir. Ce n'est souvent qu'à l'usage en production que certains bugs critiques finissent par être découverts. Dans ce cas, que doit-on attendre du fournisseur du logiciel ? Tout simplement la mise à disposition d'un correctif dans les meilleurs délais ! C'est précisément là qu'intervient la notion de "parc d'utilisateurs". Le logiciel libre permet à chacun, selon ses compétences naturellement,

d'intervenir immédiatement sur un bug fraîchement détecté. La proportion de développeurs parmi les utilisateurs de logiciels libres fait qu'il n'est pas rare de voir apparaître des correctifs, quelques heures seulement après qu'un bug ait été répertorié. Et ceci est naturellement aussi vrai en ce qui concerne les failles de sécurité. Cette réactivité n'est que très rarement atteinte par des éditeurs commerciaux, qui ne disposent pas toujours d'équipes de développeurs prêtes à réagir dès qu'un problème leur est signalé sur un produit donné, ce qui peut provoquer une inertie très importante, retardant d'autant la sortie d'un correctif. Sans compter que de tels correctifs peuvent parfois représenter une fort mauvaise publicité...

Est-ce à dire, donc, que les logiciels libres sont nécessairement, voire mathématiquement, plus sûrs que les logiciels commerciaux ? Non, naturellement non. La réactivité de la communauté du logiciel libre est globalement proportionnelle au nombre d'utilisateurs d'un logiciel. S'il est évident que des logiciels, comme le noyau même des systèmes GNU/Linux, ou bien encore Apache ou OpenOffice disposent d'une organisation suffisamment forte pour répondre en urgence aux problèmes de sécurité, alors que cela ne sera pas le cas d'un petit logiciel méconnu, utilisé sur quelques centaines de postes. Il est donc légitime de s'interroger sur la fiabilité de certains produits, mais pas de douter du libre parce qu'il s'agit de libre !

En conclusion, nous rappellerons que quelle que soit la solution et la plate-forme qui retiendra votre préférence, il restera toujours plus sage d'investir dans un excellent système de sauvegarde !

■ Gauthier Delamarre

www.programmez.com

La source de vos sources

Open Source : maintenir et migrer

Bien entendu, le support d'un outil Open / Libre concerne tous les utilisateurs. Pour l'entreprise, c'est crucial. Car, même, s'il y a compétence interne, il est impossible de tout faire. Il y a aussi un problème de maintenance et de mise à jour. Une veille est-elle nécessaire ?

La veille nécessite un travail en interne afin de suivre l'actualité. Les projets Open / Libre évoluent souvent très rapidement. Comme dans le propriétaire, il faut s'assurer de la bonne qualité d'une mise à jour et de son utilité. Le test avant déploiement constitue toujours un travail assez long. Pour les distributions, le problème est le même. Elles sont mises à jours plusieurs fois par an.

Ne parlons même pas du noyau Linux. Si vous n'avez pas de compétence en interne, ou que vous ne savez pas le faire, n'essayez surtout pas de changer de noyau. Attendez la mise à jour de votre distribution. Cependant, il n'est pas inutile de suivre les évolutions fonctionnelles du noyau.

Dans le pire des scénarios, un projet Open Source utilisé peut s'arrêter du jour au lendemain. Là, il n'y a pas beaucoup de solutions. Soit un projet concurrent peut le remplacer et récupérer les données, soit il y a un autre projet, mais les données existantes ne sont pas reconnues, ou alors vous décidez de garder quand même le projet mort, car il est assez stable et fonctionnel pour le moment. Solution de court terme. C'est pour cette raison, qu'il ne faut pas (ou tout du moins éviter) déployer des projets peu connus (ayant donc une faible communauté), surtout si l'application doit s'occuper d'un domaine sensible. Privilégiez le projet reconnu.

Bref, on peut dire que maintenir les logiciels Open / Libre et/ou son système Linux, nécessite un travail régulier pour suivre les évolutions, évaluer, tester puis déployer. Sur ce point, on ne diffère pas du modèle propriétaire.

Pour la migration

Il faut procéder par étape. Tout d'abord, si vous êtes une grande entreprise, ou une petite, réaliser un audit (en interne ou via un prestataire externe) afin d'évaluer les besoins fonctionnels (et applicatifs) de chaque département, entité, utilisateur. Il faut faire corres-

pondre la solution Open Source à l'ancienne solution propriétaire. Second point, évaluer le ou les produits Open Source que vous avez choisi(s) afin de voir si les besoins fonctionnels sont remplis. Si ce n'est pas le cas, voyez si d'autres projets sont possibles. Dans tous les cas, il faut vérifier la pérennité du dit projet et voir s'il n'est pas déjà mort. Après la phase de test, viendra la phase de déploiement. Là aussi, ne procédez pas à un déploiement massif. Faites l'expérience sur quelques postes. Dans le cas d'une utilisation mixte (Open Source – Propriétaire), vérifiez la bonne compatibilité entre les deux mondes. Il faut surtout dès le départ impliquer les utilisateurs, comme toujours, il ne faut pas les prendre par surprise. Sans eux, une telle migration est

environnement applicatif Open Source. Car il ne faut pas oublier que les applications Open Source doivent la plupart du temps prendre en compte un existant. La chaîne ne doit pas être rompue ou corrompue. L'évaluation fonctionnelle doit valider ou infirmer cette partie.

Pour un utilisateur indépendant, un utilisateur lambda comme vous et moi, la démarche n'est guère différente, on change juste d'échelle. Nous avons des habitudes avec les logiciels propriétaires. Le logiciel Open Source ne doit donc pas nous contrarier. À vous d'évaluer les besoins et le meilleur outil Open Source. Pour les logiciels les plus courants, de type suite bureautique, cela ne posera pas ou peu de problèmes, hormis si vous utilisez une version très récente de Microsoft Office ou des fonctions très précises, ex. : InfoPath ou la gestion des droits numériques. Dans ce cas, pas de solutions alternatives. Sur le format spécifique WordML ou ExcelML, les schémas ont été ouverts. Donc, on peut s'attendre très rapidement à une compatibilité dans les suites Open Source.

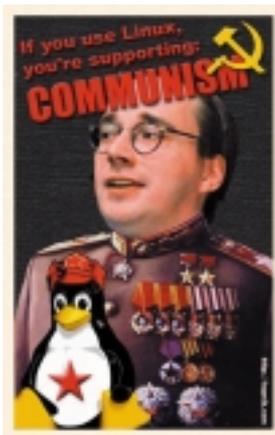


voquée à l'échec. Formations, réunions sont nécessaires à la bonne compréhension. Changer les habitudes de travail est toujours déstabilisant. Heurter un utilisateur, c'est fournir un prétexte à l'échec.

Une fois la phase décision prise, il faut maintenant vérifier que les données existantes sont migrables et implémentables dans le nouvel

Si vous n'êtes pas un utilisateur averti en Open Source, dans un premier temps préférez les grands classiques. Si vous souhaitez garder Windows ou MacOS X, pas de problème. Encore une fois, vous n'êtes pas obligé de passer à Linux pour profiter de l'Open Source.

■ F.T.



Linux et l'Open source dans tous leurs ETATS

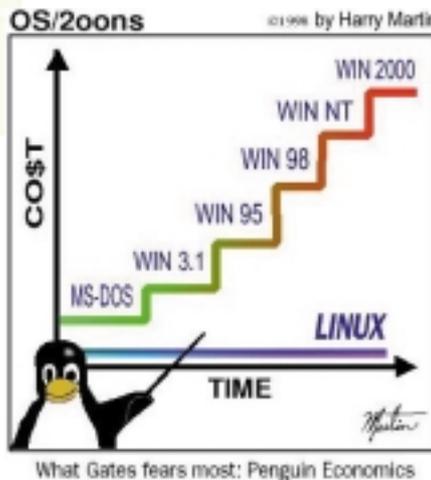
Depuis 13 ans qu'il est né de l'idée d'un seul individu, Linux a prouvé, par sa propagation même et ses développements, la pertinence et la viabilité du concept de l'Open Source. On est ainsi passé de simples individus passionnés de code à un marché professionnel, entreprises développant ou utilisant des applications sous OS libre. Depuis le milieu 2003, une nouvelle étape est en train d'être franchie : le ralliement affirmé de la plupart des gouvernements et administrations. Linux entre dans tous ses Etats !

"L'administration du 3e millénaire sera électronique, ou ne sera pas !" Vœu pieu ? Non : malgré les réticences et les lenteurs, l'E-gouvernement n'est déjà plus aujourd'hui un fantôme, mais un état de fait, validé récemment en France par la dernière directive du gouvernement (octobre 2003) sur le développement nécessaire et immédiat de l'administration électronique. On notera que cette évolution est aussi celle des "clients" : une étude a révélé au même moment que l'internaute français arrivait en tête en taux de consultation des sites publics (Education National et ANPE surtout) devant ses homologues allemands et britanniques. Pourtant, cet appétit n'est pas encore pleinement satisfait par les services publics "fournisseurs" : une autre étude, réalisée cette fois par Cap Gemini pour la Commission Européenne, a évalué la disponibilité en ligne des vingt services administratifs de base, pour les citoyens (emploi, impôts, permis de construire, etc.) et pour les entreprises (impôts sur les sociétés, douanes, TVA, etc.). La moyenne des pays européens se situe fin 2003 à 67% de disponibilité, pour 60% en 2002. La France n'arrive qu'au 7e rang de ce hit-parade, avec cependant une bonne progression de 10% en un an.

Migration : le visa de l'industrie

Bref, il reste du chemin à faire. Et il semble bien, aujourd'hui, devoir passer par l'Open Source, et Linux au premier plan. Les motivations de l'Etat ne sont pas différentes de celles des entreprises privées : soucis d'économie, de fiabilité, et d'indépendance par rapport aux langages et applications propriétaires. Or l'Open Source répond à ces attentes, et le contexte lui est aujourd'hui totalement favo-

nable. D'abord, le logiciel libre vient de gagner en respectabilité à cause du ralliement massif des grands acteurs du marché. Novell a effectué en novembre dernier un spectaculaire virage "libre" : rachat du distributeur allemand Suse Linux, rachat de Ximian (créateur de Gnome), et développement d'une plate-forme de services complète en OS libre. En même temps, IBM a entrepris de livrer la version 3 de Red Hat Enterprise Linux avec l'intégralité de ses serveurs, qu'ils soient à base d'Intel, de



RISC, ou sur les mainframes. Sun, Oracle ou HP ont aussi pleinement intégré Linux à leur offre, garanties d'expérience et de support. Voilà de quoi rassurer le plus frileux des décideurs informatiques de l'administration publique !

L'attitude de Microsoft participe aussi largement à ce contexte de glissement vers le "libre". Directement dans la ligne de mire, l'image globalisante de Windows et de Bill Gates attire toutes les rancoeurs ainsi que toutes les condamnations pour "position

dominante". Concrètement, la récente transition de Windows NT à Windows XP, presque imposée en administration-serveur, a mis en avant les problèmes de dépendance et de coûts dûs aux systèmes d'exploitation propriétaires. Enfin, ce n'est pas la vaste campagne publicitaire anti-Linux entamée début 2004 qui devrait arranger les choses... Elle montre surtout la réalité de la peur du géant !

De l'Allemagne à l'Australie

On assiste donc, en Europe comme dans le monde, à une volonté logique des gouvernements de migrer leurs administrations électroniques vers l'Open Source en général, et vers Linux en particulier. Un des cas les plus spectaculaires est celui de la ville de Munich qui a décidé de faire passer de Windows à Linux les 14000 ordinateurs d'administration de sa communauté urbaine, pour un coût de près de 30 millions d'euros. Et la visite en catastrophe de Steve Ballmer, gros bonnet de Microsoft, n'a pas pu convaincre le maire de la ville... La transition, encore en cours, doit même désormais s'accompagner d'une migration depuis l'Office de Microsoft vers le logiciel libre OpenOffice ! Même volonté en Angleterre : depuis octobre



dernier, neuf programmes d'évaluation de Linux sont en cours de test avec IBM. Objectif : évaluer les économies que pourraient réaliser les services ministériels,

dont celui du Bureau du premier ministre, en passant à l'Open Source. Un programme similaire se développe simultanément à Moscou, en collaboration avec le gouvernement russe et les universités locales, qui combine cette

fois évaluation matérielle et formation des administratifs.

Enfin, la palme de la clarté revient à l'Australie et à l'organisme d'état ACT (Australian Capital Territory) qui a publié en décembre dernier un décret demandant aux organismes publics de " ne pas utiliser de logiciels qui ne répondent pas aux standards Open Source et ISO ". L'auteur du décret, la démocrate Ms Dundas, ajoute d'ailleurs : " Depuis juin de cette année, notre administration a payé près de 15 millions de \$ à Microsoft en licences logicielles. Pourtant, nous avons les outils équivalents en Open Source, de quoi faire des économies et créer dans la foulée des emplois. "

Liberté, Egalité, Fraternité

La France n'est pas en retrait. La Poste, France Telecom, la Macif, le CEA, certaines branches de l'Education Nationale, de multiples conseils régionaux, collectivités locales et services administratifs ont opté pour le logiciel libre, au-delà même de Linux. Le CEA (Commissariat à l'Energie Atomique) a ainsi choisi la plateforme Open Source Zope pour gérer la gestion et la collaboration des informations entre ses équipes – un choix de gestion documentaire qui a résulté d'une longue étude comparative. Pour son site Web, le Conseil du Médicament, organisme directement dépendant du ministère de la Santé et des Services Sociaux, vient aussi de se rallier aux outils logiciels libres (entre autres PhpWebSite et MySQL) pour assurer la mise à jour des listes de médicaments. Les Journaux Officiels (JO) du gouvernement entament de leur côté une pleine migration vers le XML, langage libre de gestion de données dont seul le W3C se contente de régir, comme pour le HTML, la conformité. Bref, au-delà même de Linux et du système d'exploitation, la migration des administrations françaises se dirige vers toutes les applications du libre.

L'état français ne cache d'ailleurs pas ses préférences, en particulier à travers les prises de position de l'ADAE, son Agence pour le Développement de l'Administration Electronique. L'ADAE a tenu, ce 4 février, une conférence sans ambiguïté au Salon Solutions Linux de Paris, insistant sur la nécessité d'une administration électronique de qualité. Sa ligne de conduite principale veut assurer la rationalisation des dépenses informatiques et une mutualisation la plus large possible : " A ce titre, les logiciels libres constituent un socle

technologique de biens communs structurants. Ils offrent toutes les qualités requises de transparence et de fiabilité. Ils permettent de maîtriser le rapport coût total / adéquation aux besoins par le renforcement de la concurrence, ils permettent de maîtriser les logiciels eux-mêmes et de pouvoir s'assurer de leur pérennité. Enfin, ils sont parfaitement adaptés à l'effort de mutualisation d'ores et déjà engagé. " Peut-on être plus clair ?

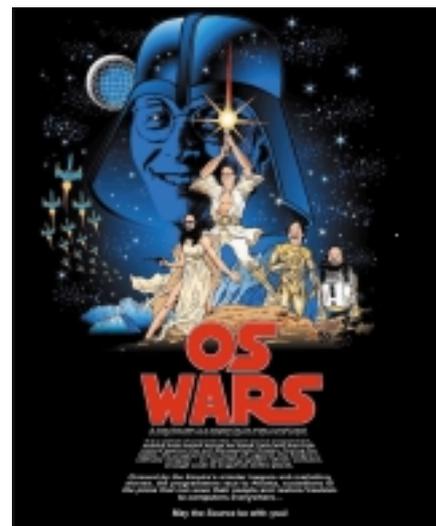
Aux armes citoyens !

Du côté des syndicats, même son de cloche, et on veut aller plus loin. C'est ainsi, par exemple, que les organisations syndicales de l'éducation et les associations du logiciel libre se sont réunies en janvier 2004 pour publier une note tout aussi significative : "Les baisses de prix que Microsoft a été amené à consentir sont avant tout le résultat de la percée considérable des suites bureautiques libres et gratuites, notamment OpenOffice.org, dans le monde de l'Éducation /.../. Les participants à la réunion /.../ rappellent cependant que l'usage d'une suite bureautique libre permet encore plus d'économies, et de façon plus pérenne. En outre, les réductions obtenues dans l'accord-cadre ne s'appliquent ni aux enseignants ni surtout aux élèves, alors que les suites bureautiques libres bénéficient à tous."

On note aussi que les associations défendant le logiciel libre dans l'e-gouvernement ne cessent de se multiplier, alliant des actions concrètes à une dénomination souvent humoristique, d'ailleurs bien caractéristique de l'esprit Open Source ! On citera par exemple l'ALLEGETICE (Alternative Logiciels Libres Pour les Ecoles Gardoises Evitant Toute Informatisation Coûteuse Exagérée) qui propose sur son site le téléchargement d'un kit complet d'applications libres, ou l'OISILLON (Options Innovatrices et Synergiques pour l'Introduction des Logiciels Libres dans les Organisations Nationales) qui " regroupe des fonctionnaires et des citoyens désireux de favoriser l'adoption de méthodes et de solutions logiciels libres dans les administrations publiques ".

L'avocat du diable

Alors, c'est le paradis ? Tout le monde semble d'accord. Individus, associations, administrations, états, entreprises privées même, tous apparaissent emportés par Linux et les logiciels libres dans un immense élan de générosité ! Les plus optimistes la qualifieront même



" d'alter-mondialiste ". Et quand ils sauront que Richard Stallman, le père des logiciels libres et de la licence GPL, vient juste de conclure une tournée triomphale de dix jours en Inde par une visite au président Abdul Kalam, comme Bill Gates, cette fois-ci ils n'auront plus de doute, la bonté numérique a submergé le monde ! Et vous ne pourrez vous empêcher de verser une petite larme...

Séchez-la vite avec votre mouchoir. Linux et le logiciel libre, c'est bien, mais les motivations des différentes parties, quoique louables, doivent peu à la générosité et ne peuvent en aucun cas faire croire à un miraculeux Puerto Alegre électronique. L'ultra-libéralisme soudainement submergé par la solidarité de l'Open Source ? Attendez : IBM doit plus de 17% des trois milliards de dollars de son chiffre d'affaires 2002 à Linux, augmentation prévue de 30% en 2003. Hewlett-Packard annonce que ses activités liées au système d'exploitation libre ont progressé de 40% en 2003, pour un chiffre d'affaires de 2,5 milliards de dollars. Une étude prédit en outre que le passage généralisé des administrations des pays industrialisés aux logiciels libres entraînerait une augmentation de chiffre d'affaires du secteur de 238% d'ici 2007... Bref, même si le passage des administrations à l'open source est souhaitable, prometteur de progrès, on est assurément loin des idéaux de Linus Torvalds ou de Richard Stallman ! Et n'oublions pas que Microsoft, quand il a dû consentir, en août dernier, à brader de 50% ses licences au gouvernement thaïlandais, a déclaré vouloir, lui aussi, "aider les plus démunis à accéder aux technologies modernes"...

■ Jean-Michel Maman

La Chine à Paris

Un peu d'exotisme dans les allées de Linux Solutions : quelques jours après la visite à Paris du numéro 1 chinois, un stand aux couleurs de l'Empire céleste, avec des exposants en costume de soie.

La société Ahead compte 300 employés, majoritairement développeurs. Elle s'est spécialisée depuis 10 ans dans le développement Linux. Elle a conçu une application, A6, comportant des fonctions décisionnelles et des outils de Knowledge Management, bien implantée en Chine, y compris dans l'administration.

La raison de sa venue au salon : trouver des distributeurs, proposer de l'outsourcing et du développement offshore, et de manière générale, trouver des partenaires français, servir de "pont" entre la France et la Chine, sur le marché du développement et des services Linux.

Ahead est une des 5 sociétés associées au projet d'un Linux national chinois. Nom de code : Red Flag. "On veut faire en Chine le grand centre mondial de R&D Linux", nous déclare Su CHEN, le président de la société. Ses atouts : le bas salaire local, évidemment, 300 euros en moyenne. Et l'immense marché à conquérir. Le budget voté pour 2005 par l'Etat chinois, en ce qui concerne l'informatisation de l'administration devrait s'élever à 25 milliards d'euros.

"Les Français doivent rattraper les Américains", plaide Su Chen, qui se dit impressionné par le niveau technique de notre beau pays. La France est un des premiers pays à avoir établi de bonnes relations avec la Chine, souligne S. Chen. Et puis, notre homme aime

la littérature française. Ses favoris : Le Rouge et le Noir de Stendhal, les Misérables ou Notre dame de Paris de V. Hugo, et toute l'oeuvre de Balzac ! Contact en France : Cheryl YU yuxuan@aheadsoft.com.cn

■ Jean Kaminsky



Percée mondiale 2003 de l'Open Source

Afrique du Sud : le Ministre des Services Publics diffuse un rapport affirmant la qualité et la fiabilité des logiciels libres, et conseillant leur utilisation. (septembre 2003)

Brésil : recommandation du Président Lula que 80% des ordinateurs achetés par les administrations en 2004 soient équipés en OS et logiciels libres. (décembre 2003)

Chine, Japon, Corée du Sud : annonce d'une collaboration pour développer un nouveau système d'exploitation, basé sur Linux, propre à ces trois pays. (septembre 2003) En Chine, Gou Zonghwen, du Ministère des techniques et de l'Information, affirme la nécessité de passer en Open Source pour assurer la percée technologique des administrations chinoises. (novembre 2003)

Etats-Unis : un rapport d'étude incite la NASA à passer à l'Open Source. Au Texas, un décret demande un rapport justificatif pour toute administration choisissant un logiciel ou système propriétaire au lieu du libre. (avril 2003)

Inde : le gouvernement indien relance la Linux India Initiative, "plan de migration vers les logiciels libres de l'administration". (juin 2003)

Malaisie : le Ministre des finances alloue des fonds spéciaux au Malaysian Venture Capital, pour soutenir toutes les entreprises de l'Open Source. (octobre 2003)

Thaïlande : fin 2003, 5% des services administratifs sont en "libre". Le plan du ministère annonce un objectif de 50% sur deux ans, avec développement d'un Linux spécifiquement thaïlandais, TLE Linux. (novembre 2003)

LE MAGAZINE DU DÉVELOPPEMENT
Programmez!

Abonnez-vous !

voir coupon d'Abonnement p 61

45 €

au lieu de 65,40 €

Guide IDEALX des logiciels Open Source

L'application de notre méthodologie de qualification des composants Open Source est résumée dans notre Guide sous forme de trois indices de synthèse, afin de faciliter l'analyse de premier niveau. (Extrait du tableau)

Grâce à notre expérience de terrain, nous rendons compte des caractéristiques de chaque logiciel selon les critères suivants :

- **richesse fonctionnelle** : note l'adéquation aux besoins constatés. Il n'est pas toujours indiqué d'utiliser le plus riche logiciel (plus coûteux à mettre en oeuvre ou à maintenir), tant sur le plan de l'évolution avérée et prévue (roadmap) que de la documentation ;
- **capacité en production** : le logiciel est d'autant plus apte à l'utilisation en production que le nombre d'étoiles est élevé, moins de trois

étoiles révélant le caractère encore immature d'un logiciel par ailleurs plein de promesses ;

- **tendance** : indique si la diffusion dans les environnements professionnels progresse, reste stable, ou bien régresse ;
- **observation** : commentaire global.

L'application de notre méthodologie de qualification des composants Open Source est résumée dans notre Guide sous forme de trois indices de synthèse, afin de faciliter l'analyse de premier niveau.

Poste de travail

Type	Nom	version	Richesse	Exploitation	Tendance	Commentaire
Interface	KDE	3.1	***	***	↑	Riche, cohérent, solide. Architecture évoluée.
	GNOME	2.4	***	***	↗	Évolution remarquable. Soins appréciables pour l'ergonomie.
Suite bureautique	Koffice	1.2.1	**	**	↑	
	OpenOffice.org	1.1	****	***	↗	Très riche et puissant, disponible en français.
Tableur	Gnumeric	1.2.2	***	**	↑	Simple et puissant mais inférieur à OpenOffice.org
Trait. de texte	AbiWord	2.0.2	**	***	↑	Simple et efficace.
Dessin bitmap	The GIMP	1.2.5	****	***	↑	Satisfera de nombreux professionnels. La colorimétrie manque.
Dessin vectoriel	Sketch	0.6.15	**	**	↑	Manque d'objets (galerie).
	Sodipodi	0.33	***	***	↗	Prometteur
Modeleur 3D	Blender	2.25	***	****	↗	Professionnel, multi-plate-formes, extensible. Ergonomie améliorée dans la prochaine release (2.3)
Navigateur	Mozilla	1.5	***	***	↗	Classique et efficace. Les clients autonomes, Firebird pour le Web, Thunderbird pour la messagerie, sont des successeurs talentueux qui faciliteront la migration Open Source des entreprises ayant déjà déployé IE ou Outlook par défaut.
PAO	Scribus	1.1.3	**	**	↗	Évolution intéressante.
Éditeur HTML	Bluefish	0.12	***	***	↗	
	Quanta	3.2	***	***	↑	
Groupware	Ximian Evolution	1.45	***	***	↗	Très complet. Bonne ergonomie, en particulier pour les utilisateurs de Outlook. Une référence.
	OpenGroupware	1.0	***	**	↗	Solution de remplacement Exchange. Interface Web complète. Prometteur
	Kollab	1.0	**	**	↗	Client serveur, compatible Outlook, avec un nouveau client KDE pour les bureaux Linux.
	hylafax	4.1.8	***	****	↑	Riche et adaptable
Émulation MS-Windows	WINE	200312	**	***	↗	Évolution toujours soutenue. Une brique essentielle pour la migration.
	CrossOver	1.3.1	***	****	↑	Non Open Source
Inter-opérabilité	rdesktop	1.3.0	**	****	↗	Client pour serveurs TSE (protocole RDP)
	x3270	3.3.2	***	****	↑	Émulateur 3270. Solide et éprouvé.

Outils de développement

Type	Nom	version	Richesse	Exploitation	Tendance	Commentaire
Documentation	doxygen	1.3.5	***	***	↑	
Débugage	Valgrind	2.0	***	***	↗	Aide à l'analyse des erreurs mémoire et thread.
Débogueur	DDD	3.3.8	***	****	↑	Supporte l'essentiel des débogueurs symboliques / langages. Excellent frontal graphique pour examiner les structures de données.
Éditeur	Emacs	21.3	****	***	↑	La référence
EDI	eclipse	2.1	****	***	↗	Incontournable.
	NetBeans	3.5	***	***	↑	
	KDevelop	2.1.5	***	***	↑	Environnement intégré de développement multilingage
	Anjuta	1.2.0	**	***	↗	Équivalent sous Gnome.

Gestion de conf.	CVS	1.11.9	↑	Éprouvé.
	Subversion	0.31.0	↗	Corrige les limitations de CVS. À suivre.
	Bugzilla	2.16.2	↑	Rapport et suivi de bogue via une interface web.
	GForge	3.1	↗	Environnement de développement collaboratif en mode Web « à la SourceForge »
Modélisation	ArgoUML	0.15.2	↗	Prometteur. Dès à présent : bon outil d'initiation.
Portabilité	autoconf, automake, libtool	N.A.	↑	Rendent les codes portables.
	Ant	1.6.0	↗	LE standard pour le monde J2EE
	IKVM	N.A.	..	.	↗	JVM pour .Net : permet d'exécuter des applications Java dans un CLR. Progression impressionnante.
Explorateur de source	Source Navigator	5.1.4	↑	IDE orienté exploration de code.
Références croisées	lxr	0.3	↑	Présentation Web.
	GLOBAL	4.6.1	↑	
Test	DejaGnu	1.4.3	↑	Gestion d'une suite de test
	JUnit	3.8.1	↑	Test unitaire -Java
	Cactus	1.4.1	↑	Test fonctionnel et intégration. Mise en oeuvre délicate.
Trace	syscalltrack	0.82	↗	Strace dynamique (interception possible), avec règles.

Exploitation

Type	Nom	version	Richesse	Exploitation	Tendance	Commentaire
Administration	Linuxconf	1.32r1	↑	
	webmin	1.121	↑	Ergonomique, souple, nombreux modules.
Analyse log Web	AWStats	5.9	↗	Riche. Surtout utile en mode CGI.
	Analog	5.32	↑	Complet et fiable.
	Webalizer	2.01-10	↑	Traite bien plusieurs mois d'activité.
Configuration	cfengine	2.1.0	↑	
Déploiement	stow	1.3.3	↑	
Monitoring	GKrellM	2.1.7a	↑	Vu-mètres d'activité système.
SGBDR	MySQL Navigator	1.4.1	↑	
administration	phpMyAdmin	2.5.5pl1	↗	Existe également pour PostgreSQL, Oracle 8 et Sybase.
Sauvegarde	Amanda	2.4.4	↗	
	Partition Image	0.7.2	↗	
Suivi réseau	RRDTOOL	1.0.46	↑	
Supervision	Ganglia	2.5.5	↗	Supervision de grands parcs.
	Nagios	1.1	↗	Riche et efficace. Mise en oeuvre délicate. Ex-Netsaint.
	Scotty	2.1.10	↑	Outils Tcl/TK pour la gestion SNMP.
	OpenNMS	1.0.2	↑	Plate-forme de supervision J2EE
Surveillance	MON	0.99.2	↑	Automatisation de la surveillance. Très bon outil.
Gestion d'incident	RequestTracker	3.0.3	↗	

Serveur d'applications et frameworks

Type	Nom	version	Richesse	Exploitation	Tendance	Commentaire
.NET	Mono dotGNU	0.29	↗	
	Portable .Net	0.6.0	↑	
Langages	Erlang	R9C-0	↑	
Open Source	Perl	5.8.2	↑	Puissant et doté de riches bibliothèques et documentations.
	PHP	4.3.4	↗	Parfaitement adapté au web.
	Python	2.3.2	↗	Efficace mais encore peu répandu.
ORB	ORBit	2.9.0	↑	Mature.
Serveur d'application	JBoss	3.2.3	↑	Bonne qualité. Nouvelle version intéressante avec l'ajout d'éléments AOP
J2EE	JOAS	3.3.1	↗	Complet. Novateur. Profite de la richesse des projets du consortium ObjectWeb

Portabilité GUI	wxWindows	2.4.2	***	***	↑	
Publication XML	Cocoon	2.1.3	***	**	↑	Framework de publication de documents XML, avec plugins et connecteurs (LDAP,SQL) – Technologie Java
	AxKit	1.6	***	**	↑	Équivalent dans le monde Perl
Service Webs	Axis	1.1	**	.	↗	Conteneur d'applications dédié pour les Services Webs
Plateforme	Zope	2.6.2	****	***	↗	Environnement riche. Technologie Python.
Frameworks Java	Struts		***	***	↑	JSP et servlets. La référence, en constante amélioration.
	Lucene	1.2	***	***	↑	Moteur de recherche texte
	Hibernate	2.1.1	**	**	↗	Mappeur relationnel /objet
	Avalon	N.A.	**	.	↗	Architecture de composants. Prometteur

Infrastructure

Type	Nom	version	Richesse	Exploitation	Tendance	Commentaire
Annuaire	OpenLDAP	2.1.25	***	***	↗	Robuste et bien documenté.
Gestion volumes	LVM	1.0.8	**	***	↑	Gestion des volumes de stockage.
	EVMS	2.2.1	***	***	↗	Gestionnaire de volume avec interface de gestion puissante. Haute disponibilité
	Kimberlite	2.0.0	**	***	↑	Gestion du clustering, avec stockage partagé.
	Heartbeat	0.2	**	***	↑	Surveillance et amorçage de services.
Cluster /Équilibrage de charge	Linux Virtual Server	1.0.10	***	****	↗	Très performant
Cluster /Consolidation de systèmes	OpenMosix	2.4.22-2	**	**	↑	
Stockage réseau	DRDB	0.6.6	**	**	↗	Équivalent à un « RAID réseau »
	OpenGFS	0.3.0	**	**	↗	Système de fichier réparti, redondant
Impression	CUPS	1.1.20	***	***	↗	En passe de devenir le standard.
Machine virtuelle	User Mode Linux	2.4.19-5um	**	**	↑	'Machine virtuelle' Linux complète.
Mandataire SOCKS	DANTE	1.1.14	****	****	↑	Mature.
Mandataire-cache Web	Squid	2.5	****	****	↑	Performant et robuste, configuration parfois délicate.
Messagerie	Exim	4.30	***	***	↑	Peu de déploiements industriels.
	Postfix	2.0p16	***	****	↑	Sûr et performant, mise en place et maintenance aisée.
	Sendmail	8.12.10	***	**	↑	Complexe, ancien.
	Messagerie IMAP	Courier-IMAP	2.2.1	***	****	↑
	Cyrus IMAP	2.1.16	****	****	↑	Plus riche fonctionnellement
Messagerie, anti-UCE et SPAM	SpamAssassin	2.61	***	***	↗	Très efficace.
Antivirus	Amavis-ng	0.3.12	***	***	↑	Connecteur très utilisé
	Clamav	0.65	**	**	↗	Offre à présent mature.
Messagerie client Web	SquirrelMail	1.4.2	***	**	↑	Ergonomique et doté de très nombreux modules.
	IMP	3.2	***	***	↑	Concurrent direct (ou est-ce l'inverse...)
Messagerie instantanée	Jabber	1.4.2	***	***	↑	Messagerie instantanée ouverte. Proposition de standard IETF.

Sécurité

Type	Nom	version	Richesse	Exploitation	Tendance	Commentaire
Chiffrement	GnuPG	1.2.7	***	***	↗	
Connexion	OpenSSH	3.7	****	****	↗	L'outil de référence. Cf. aussi le logiciel "PuTTY".
	PuTTY	0.53b	***	***	↗	SSH pour MS-Windows.
Contrôle	sysrince	1.4	***	**	↗	Limite les possibilités offertes aux programmes.
Interface d'administration / Filtrage	Firewall Builder	1.1.1	***	***	↗	Interface de configuration multi-plate-forme.
	isba	1.1	**	**	→	Interface de configuration et de diffusion.
IGC (PKI)	IDX-PKI	1.9	***	****	↗	Cohérente et modulaire.
Intégrité	AIDE	0.10	***	***	↑	Simple
NIDS et intégrité	Prelude	0.8	***	.	↑	Complet et industriel.
NIDS passif	snort	2.1	***	**	↗	Détecteur d'intrusions. Domaine prometteur, en cours d'exploration. Immature.

Protection Passive	Bastille Linux	2.0.4	↑	
	COPS	1.04	↑	
	Nessus	2.0.9	↗	Détection de vulnérabilités. L'outil le plus réputé.
	SARA	5.0	↑	
	grsecurity	1.9.13	↗	Patch Linux.Complet.
VPN	FreeS/Wan IPsec	2.0.4	↑	Préférer l'IPsec d'OpenBSD.
	(OpenBSD)	3.4	↗	Interopérable, souple et robuste.
Scanner	nmap	3.48	↗	Fiable, fonctionnel et éprouvé
Transport	stunnel	4.0.4	↑	Attention aux failles d'OpenSSL.
Technologie Firewall	Packet filter OpenBSD	3.4	↗	Puissant, simple d'administration, fonctionnel et stable ;
	Ipfiler	3.4.33	↑	Portable
	Netfilter	1.2.9	↑	Complexe à mettre en oeuvre.

Applications d'entreprise

Type	Nom	version	Richesse	Exploitation	Tendance	Commentaire
CAO	Open Cascade	5.1	↑	Composants logiciels. Complet et mature.
Commerce Web	Interchange	5.0	↑	Stable et éprouvé
	osCommerce	2.2	↗	Progression rapide, solide.
ERP CRM	Compiere EC	2.5.0	↗	Approche élégante, évolution rapide. Déploiement intéressants
	ERPS	0.1	.	.	↗	Prometteur
Forum	phpBB	2.06	↗	Un nouveau standard est né
Gestion de contenu	Back-End	0.7	↑	
	SIIP	1.6	↑	Intéressant pour les petites structures.
	PHP-Nuke	7.0	.	..	↑	Très orienté « news »
	Red Hat CCM	5.0.2	↗	Industriel, complet et ouvert. Nécessite une intégration importante. Technologie J2EE
	Midgard	1.5	↑	Architecture solide, environnement PHP, mais communauté un peu en retrait
	Bricolage	1.6.8	↑	Back-office complet. Technologie Perl
	Typo 3	3.5	↗	Ensemble cohérent et simple à mettre en oeuvre. Technologie PHP
	Plone	2.0	↗	Architecture et cohérence de l'évolution. Simple à utiliser et administrer. Technologie Python
	OpenACS	4.6.3	↑	Successeur du Ars Digita Community System. Riche et éprouvé. Parfois « décalé ». Technologie TCL
Générateur de rapports	DataVision	0.8.0	↑	Connecteur JDBC et pivot XML. Outil graphique de design des rapports.
	JasperReports	0.5	..	.	↗	Plus orienté outil et framework. Technologie J2EE
Moteur de recherche	htdig	3.1.6	↑	Développement ralenti mais la communauté utilisateur reste active.
	mnoGoSearch	3.2.15	↑	Riche et complet. Largement répandu.
	Lucene	1.3	↗	Moteur et toolkit reconnu pour les développements J2EE.
Portail	uPortal	2.1.4	↑	Complet et novateur. Très orienté vers le monde universitaire.
	jetspeed	1.4	↗	Nouveau développement très prometteur avec le support du JSR-168 et les soutiens industriels de la fondation Apache.
	ExoPortal	1.0 beta	↗	Support JSR-168. Utilise les extensions AOP du conteneur Jboss.
	Metadot	5.6.5.3	↗	Complet et solide. Technologie Perl.
	CTI	bayonne	1.2	↗
EAI	openadaptor	1.6.0	↑	Mature, bien documenté, communauté active.
	Business Integration Engine (BIE)	5.6.4	↗	Riche et performant. Interface graphique de design des règles de gestion.
	xmlBlaster	0.8.5e	↑	Bus XML, à incorporer dans une solution plus générale. Novateur.
e-Commerce	freebXML	N.A.	..	.	↗	Initiative prometteuse qui semble être bien suivie.
Workflow générique	jBPM 1.0	beta	↗	Moteur de workflow J2EE. Architecture intéressante.
Groupware	PHProjekt	4.1	↑	
	TUTOS	1.1	↑	
	twiggi PHP	1.10.16	↑	
	Groupware	0.9.14	↑	Complet et bien agencé.
	TWiki 01 Feb	2003	↗	Riche et solide. Un des meilleurs Wiki.
	Collaborative Portal Server (CPS)	2.2rc2	↗	Complet et ergonomique. Concurrent de Sharepoint.

Le projet " Pointers " de la SNCF Fret : .Net et Tablet PC

Aston plébiscite le choix de .NET pour intégrer de nouvelles applications à l'existant. Lors du séminaire " Dynamisez vos applications actuelles avec .NET " qu'ils ont animé en décembre dernier, les consultants d'Aston ont donné différents éléments en faveur de l'utilisation de la plate-forme Microsoft. Ils ont illustré leurs propos par un retour d'expérience sur le projet Pointers de la SNCF Fret.

L'activité fret de la SNCF correspond au transport de marchandises dans des wagons. Cela implique la gestion du parc roulant, la traction, l'infrastructure et les éléments communs à l'entreprise comme l'administratif, la comptabilité, etc. Cette entité de la SNCF est organisée de la façon suivante :

une direction avec les experts techniques, les architectes, etc.

- une division pour la maintenance et le rapport aux utilisateurs
- une division pour les projets avec leur conception, intégration, pilotage.

"Les principales attentes de nos clients : une

capacité de transport conséquente, la sécurité dans l'acheminement de tous types de marchandises, la régularité du transport plus qu'une performance de vitesse, une information et préinformation poussées, une capacité à traiter des transports internationaux" résume Gilles Hozé, responsable de la division projets.

Les problématiques techniques

Gestion des flux

Problématique :

- Gérer des flux XML du serveur d'échange vers le serveur de base de données central, et vice versa
 - Mettre en place un mécanisme de scrutation de modification de chaque côté
 - Garantir la livraison et la gestion des messages
- Pour gérer les flux XML, les schémas XSD et les feuilles XSL, nous avons utilisé les classes du .NET Framework à travers les namespaces "System.Xml.*".

Pour gérer le polling du côté de la base de données et des files MSMQ, nous avons décidé de créer des services Windows paramétrables. Cette solution permet à l'administrateur de les arrêter, redémarrer aisément via la console MMC des services. Aussi, Visual Studio .NET propose de créer des projets de cette nature, en toute simplicité.

Enfin, nous avons choisi d'utiliser des files MSMQ pour garantir la livraison des messages. MSMQ est géré directement depuis le code .NET à travers le namespace "System.Messaging" explique Nicolas Ménigon.

Déploiement applicatif automatique

Problématique :

- Application ASP.NET sur des portatifs (Tablets PC) et sur un serveur web central
- Mise à jour de la dernière version sur le central

- Détection sur les portatifs d'une version plus récente sur le serveur central
- Déploiement automatique (à chaud) et transparent de l'application sur le portatif

Dans un premier temps, un projet de déploiement Web a été créé sous Visual Studio .NET. Ce projet génère un fichier MSI qui correspond à la dernière version de l'application Pointers. Il est stocké sur un répertoire de téléchargement sur le serveur web central.

Ensuite, sur les portatifs, lorsque l'application ASP.NET est lancée, un composant COM+ vérifie la présence sur le serveur central d'une nouvelle version. Si c'est le cas, le fichier MSI est téléchargé, via http sur le portatif, et le composant utilise le moteur Microsoft Installer (MSIEXEC.EXE) pour mettre à jour "à chaud" l'application Pointers avec les dernières modifications. Tout doit se faire de façon transparente pour l'utilisateur, sans aucun écran assistant. Pour ce faire, l'application ASP.NET doit faire appel au moteur MSI et cette étape nécessite des privilèges supérieurs aux droits du compte par défaut ASP.NET. Il est donc nécessaire de passer par un "Serviced Component" qui s'exécute sous une identité ayant les autorisations idoines.

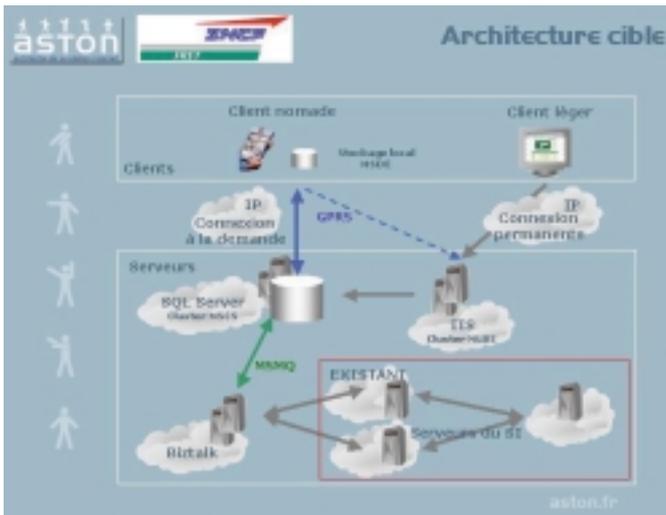
Synchro des données

Problématique :

- Synchronisation sélective des données sur demande des portatifs

- Propagation des dernières modifications de données (issues du central ou des postes nomades) à chaque synchronisation
- Manipulation directe de la base locale impossible pour l'utilisateur

Le choix de synchronisation s'est porté sur Microsoft SQL Server 2000 Enterprise sur le serveur SGBD central et sur MSDE sur chaque portatif. La réplication mise en œuvre est de type "Fusion", qui correspond au meilleur modèle pour une utilisation en mode "déconnecté". Afin de piloter la synchronisation depuis l'application ASP.NET Pointers sur les portatifs, l'équipe ASTON a créé un composant COM+ de réplication. Tout comme la mise à jour automatique, le fait de lancer depuis un portatif une réplication exige de propager un contexte d'identité avec des droits supérieurs au compte ASP.NET. Le "Serviced Component" de réplication sur le portatif, appelé par l'application web, manipule le composant COM de réplication fourni avec SQL Server pour dialoguer avec le serveur central et consolider les données. ASTON a créé un composant COM+ de réplication. Tout comme la mise à jour automatique, le fait de lancer depuis un portatif une réplication exige de propager un contexte d'identité avec des droits supérieurs au compte ASP.NET. Le "Serviced Component" de réplication sur le portatif, appelé par l'application web, manipule le composant COM de réplication fourni avec SQL Server pour dialoguer avec le serveur central et consolider les données.



Toutes ces caractéristiques métiers se retrouvent peu ou prou dans le projet Pointers. Ce projet est né du besoin de remplacer une application nomade vieille de 10 ans. Ses fonctionnalités principales étaient les suivantes : la collecte d'informations inhérentes à la composition des trains et à l'acheminement des wagons sur le terrain, la garantie, la saisie et le contrôle des branches avant leur départ et à l'arrivée, que ce soit sur poste fixe ou nomade, l'amélioration du contrôle et la sécurisation des règles de gestion pour contrôler les infos au plus près.

Les besoins énoncés dans le cahier des charges transmis à ASTON :

- offrir une réelle saisie nomade, contrôlée et ergonomique
- recueillir les informations du terrain et les réinjecter au sein du système d'information existant en vue d'être exploitées par d'autres populations d'utilisateurs.
- mettre en œuvre une architecture applicative et technique basée sur des standards et sur-

tout apte à évaluer avec les besoins métiers

"La volonté première était de ne pas remplacer tout le SI et de s'appuyer sur des composants logiciels commerciaux pour suivre les évolutions sans avoir à tout réécrire", précise Gilles Hozé.

- tirer profit à partir de la technologie .NET des gains de coût d'exploitation, de déploiement et de maintenance :

"Nous ne sommes pas dans un credo .NET contre J2EE : nous utilisons les deux, il est de toute façon difficile de suivre une voie unique. Il s'agit simplement d'en faire usage à bon escient" ajoute-t-il.

Et demain ?

L'architecture actuelle du projet Pointers est constituée au niveau des postes clients par des Tablets PC déconnectés et des postes fixes connectés.

La première évolution possible est de migrer vers une application sur poste nomade en utilisation connectée avec les évolutions de la téléphonie mobile (WiFi par exemple).

Ensuite, la plate-forme d'échanges de flux, basée sur BizTalk Server, doit servir de plate-forme commune pour d'autres applications à venir. L'architecture globale de ce projet pourra servir de référence en termes de mise en œuvre dans le cadre de besoins existants similaires.

■ Anne-Françoise Moal

Les enjeux du projet

- en termes de flux entrants et sortants : 50 flux physiques entre pointer et système central
- 400 postes de travail
- 1 000 à 1 200 utilisateurs qui sont amenés à s'en servir 7j/7 et 24h/24

Timide embellie sur le front de l'emploi

Bonne nouvelle en ce début d'année : Les SSII recrutent à nouveau ! nombre d'entre elles, pas forcément les plus importantes d'ailleurs, ont relancé des plans de recrutement, quasiment gelés depuis deux ans. Deuxième bonne nouvelle : les régions devraient largement bénéficier de cette reprise.

Des postes à pouvoir dans la plupart des entreprises

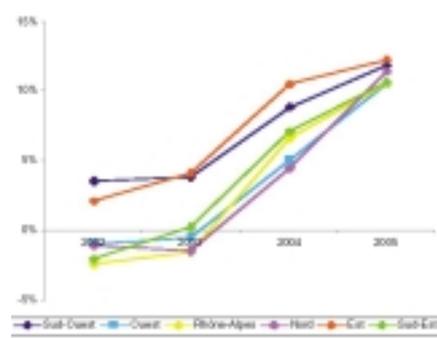
Plus de 150 offres d'emplois informatiques sur le site d'Assystem Brime Technologies (www.assystembrime.com), une société de 6 000 personnes : beaucoup de postes de développeurs, technologies requises, notamment : C++, J2EE, java. Une quarantaine d'offres chez Atos Origin, la plupart dans le développement (Java, C, C++) et autour des ERP (surtout SAP). Une vingtaine de types de profils informatiques sont recherchés par Cap Gemini Ernst & Young. De nombreux postes à pourvoir également chez Teamlog : ingénieurs systèmes (Unix essentiellement) ou de développement (www.teamlog.fr).

Gedas, SSII filiale du groupe Volkswagen, lance un plan de recrutement d'une centaine de personnes en France, dont des développeurs Java sous Weblogic, et de développeurs Forms, plus des ingénieurs d'études et chefs de projets sur grands systèmes. Elle recherche d'une part pour les missions dans le milieu de la finance, 50 ingénieurs d'études confirmés et 15 juniors qui seront formés, et pour ses business units infogérance et CRM (dans la distribution automobile), 35 Chefs projets / directeurs de projets (dont profils internationaux, 5 à 15 ans d'exp., 30 à 42 ans, anglais courant impératif). Homsys group, après avoir recruté 52 personnes en 2003 prévoit 80 nouveaux recrutements en 2004.

Valoris, société qui compte 600 personnes à Paris, Bruxelles, Barcelone et Londres, indique avoir " relancé sa politique de recrutement au cours du 2e semestre 2003, après deux années de gel " et ajoute que " cette tendance devrait

s'amplifier en 2004 ". La société prévoit de recruter 30 personnes en France : 10 consultants informatiques débutants, 10 expérimentés et 10 experts en techno. Niveau Bac +5 mini, compétences : BI, CRM et Web, et cela, dans les environnements java et .Net .

La société de conseil Vistali, spécialisée dans l'urbanisation des systèmes d'informations prévoit quant à elle de recruter 30 personnes (directeurs de mission, ingénieurs solution, consultant produit, consultant en architecture).



"évolution des différents marchés régionaux des services informatiques sur la période 2002/2005"

L'EAi est un secteur en pointe, comme vous l'a démontré le récent hors série de Programmez ! consacré à l'intégration, et les offres sont nombreuses pour les pros de ces technologies.

La situation s'améliore en régions

Pour faire du développement, passer par la région parisienne n'est plus indispensable. Au contraire, la croissance du marché des services

informatiques tendrait à être plus importante en région qu'en Ile-de-France. Selon une étude réalisée par Pierre Audoin Consultant, " le marché régional des services informatiques résiste mieux que l'ensemble du marché français ". Après deux années où certaines régions ont enregistré la décroissance du marché des services informatiques (-2% en Rhône Alpes en 2003, un taux similaire dans la région nord et une légère décroissance dans l'Ouest), toutes les régions devraient enregistrer en 2004 une hausse du marché comprise entre 4 et 11%. Les plus fortes croissances sont attendues dans l'Est et dans le Sud-Ouest, devant les régions Sud-est et Rhône-Alpes et enfin l'Ouest et le Nord. Toujours selon cette étude, ce taux de croissance devrait dépasser les 10% dans toutes les régions en 2005.

Il ne semble pas qu'il s'agisse d'un simple effet de rattrapage, mais d'un mouvement de fond plus durable, qui est à rapprocher de la tendance marquée à l'externalisation. Dans ce cadre, les SSII ont recours à l'offshore, c'est-à-dire la délocalisation de tout ou partie d'un projet à l'étranger, dans des pays comme l'Inde ou la Chine, voire la Russie ou la Roumanie, mais également au nearshore, la délocalisation de proximité. Elles ouvrent ainsi des centres en province, où les salaires sont moins élevés qu'en Ile-de-France et qui offre davantage de proximité avec les clients régionaux, notamment les PME-PMI.

Toutes les régions sont concernées par cet offshore de proximité. Une grande SSII canadienne avait innové il y a quelques années en ouvrant un tel centre dans la région de Montpellier, dont le but avoué était de familiariser ses clients avec la délocalisation, avant de franchir les frontières. Unilog s'est depuis quelques mois lancé dans l'aventure. Ainsi, Unilog dispose de six centres en province : à Amiens, Bordeaux, Grenoble, Lyon, Nantes et Toulouse. En ouvrant le centre d'Amiens l'été dernier, la SSII n'a pas caché qu'il était destiné à délocaliser les prestations de développe-

ment et de maintenance pour ses clients parisiens, et ce, à 150 km de Paris. Elle prévoit d'avoir 300 ingénieurs sur le seul centre de services d'Amiens en 2006.

L'étude de Pierre Audoin Consultants souligne que "le grand acteur du marché des services informatiques dans les régions reste Cap Gemini Ernst & Young" et précise que "le groupe est leader sur chaque marché régional". La rubrique recrutement du site (www.fr.cgey.com/carrieres/emploi/index.php), présente des offres d'emploi par région, et la plupart des offres sont pour des postes en région : six chefs de projets dans le Nord Est.

Quels profils ?

Les profils recherchés sont-ils différents sur Paris et en Province ? Difficile à dire. Tout au plus peut-on voir que certaines spécialités sont plus ou moins prisées dans certaines régions. Ainsi, il est assez logique que tout ce qui est lié à l'aéronautique soit assez apprécié à Toulouse. Dans cette région également, un certain nombre de demandes sur les technologies Windchill. D'autres exemple sont moins évidents : des spécialistes des ERP sont recherchés en Rhône Alpes et dans la région toulousaine : SAP bien sûr, mais également J.D.Edwards et PeopleSoft. En ce moment Cap Gemini recrute pas mal dans l'Est : six chefs de

projets dans le Nord Est, des Architectes intégrateurs dans la même région, des développeurs Nouvelles technologies (Java, J2EE, ASP, .Net) dans le Sud Est, des architectes système d'informations confirmés dans l'Ouest, 10 architectes nouvelles technologies, confirmés, dans la même région.

Ce sont souvent des profils déjà un peu expérimentés qui sont recherchés en région. Logique, dès lors que la plupart des SSII ont des contacts assez étroits avec les écoles d'ingénieurs locales et recrutent directement les jeunes diplômés à l'issue de leurs stages. Les offres de stages sont d'ailleurs assez nombreuses actuellement. ■ Carole Pitras

Silogic : une SSII toulousaine qui recrute



Créée en 1989, Silogic est une SSII de 430 personnes qui réalise un chiffre d'affaires d'environ 21 M€. Basée à Toulouse, elle est également implantée à Paris et à Sophia Antipolis. Cette société prévoit d'embaucher au premier semestre entre 60 et 100 ingénieurs sur son site de Toulouse, pour travailler pour le compte de son client Airbus, précise Marie-Pierre Mozzi, responsable des ressources humaines de l'entreprise.



Le bien-être des salariés avant tout

Marie-Pierre Mozzi souligne que c'est l'humain qui prime chez Silogic. La société a multiplié les initiatives pour assurer le bien-être de ses collaborateurs. L'accord sur les 35 heures, signé en 1998, avant même que la réduction du temps de travail ne soit devenue obligatoire, prévoit 24 jours de RTT par an, contre 12 à 13 en moyenne dans les SSII. Dans le même esprit, elle promeut le temps partiel : les demandes en ce sens sont toujours acceptées, souligne Marie-Pierre Mozzi, qui précise toutefois que le temps partiel n'est jamais imposé et que c'est le salarié qui gère son temps. Elle remarque d'ailleurs que depuis deux ans, de plus en plus d'hommes optent pour un temps partiel.

La SSII a d'ailleurs récemment été récompensée pour son égalité professionnelle et s'est vue remettre un prix par le Conseil Régional Midi-Pyrénées dans la catégorie "Articulation et gestion des temps de vie professionnelle et personnelle".

Silogic met actuellement en place un projet de covoiturage avec les autres entreprises du quartier. L'objectif est de réduire les frais de transport et de réaliser des économies de temps.

Les salaires proposés par Silogic sont dans la moyenne du marché et incluent intéressement participation, plan d'épargne entreprise, mutuelle, prime vacances...

Compte tenu de ces nombreux avantages, croyez-moi ils sont rares à ce niveau dans les SSII, la société affiche un turn over quasi nul. Et, s'amuse Marie-Pierre Mozzi, ce n'est pas en raison de clause de non-concurrence, pour la bonne raison qu'il n'y en a pas... là aussi cela devient exceptionnel.

Concernant l'évolution possible au sein de l'entreprise, la SSII a mis en place une gestion des compétences, avec évaluation par les salariés eux-mêmes, et par leur responsable. Les salariés peuvent s'évaluer par rapport à un poste supérieur et en fonction des lacunes constatées, des formations peuvent être proposées.

Profils recherchés :

- des ingénieurs en conception et développement de logiciels :
 - double compétence Scientifique/Informatique
 - nouvelles technologies (Java, J2EE, .net, XML)
 - temps réel embarqué
 - systèmes avioniques
 - architectures distribuées (Corba, RMI, Java, C++)
 - windchill (expérimentés)
- des consultants techniques et fonctionnels SAP
- des ingénieurs Assurance Qualité
- des experts en architectures distribuées
- des chefs de projet assistance à maîtrise d'ouvrage

Le recrutement comprend une partie technique, pour valider les compétences, et un entretien avec les ressources humaines pour évaluer la capacité d'adaptation, l'expression orale et l'esprit d'équipe. S'agissant de travailler pour un client, la SSII tient à s'assurer que ses collaborateurs refléteront les valeurs et la culture de la société.

Contact : marie-pierre.mozzi@silogic.fr



Zoom sur la certification MCDBA

La certification Microsoft Certified Database Administrator (MCDBA) s'adresse aux professionnels en charge de la conception, de la mise en œuvre et du développement de base de données Microsoft SQL Server™. Ils assurent la configuration et la gestion

Témoignage

Charlie Dancoisne consultant indépendant confirmé

Charlie Dancoisne, 39 ans, est aujourd'hui consultant informatique au sein de sa propre structure Activ Trainer. Titulaire d'un MIAGE, il débute sa carrière professionnelle dans la maintenance et le développement. Il décide rapidement de monter sa société spécialisée dans l'audit et le conseil en informatique, pour devenir consultant indépendant. Il accompagne désormais ses clients dans l'audit de performance et l'optimisation de code SQL. Certifié MCSE (Microsoft Certified Systems Engineer) mais aussi MCDBA dès 1998, il intervient aussi bien sur des missions très spécifiques dans des domaines variés tels que l'accompagnement auprès des services informatiques pour des projets ou réalisations critiques, l'audit et l'analyse méthodique de faisabilité. Il est également sollicité pour des séminaires de transferts de compétences très pointues. "Les bases de données ont toujours été mon cœur d'activité, c'est donc tout naturellement que je me suis penché sur la certification MCDBA, explique-t-il. Les experts qui ne sont pas certifiés ne maîtrisent bien souvent qu'une partie réduite d'un système. Ma certification MCDBA permet de me démarquer en mettant à la disposition des clients mon éventail de connaissances et d'expériences qui est évidemment bien plus large. En l'occurrence, cela m'apporte une vue à la fois de synthèse et approfondie de SQL Server et de tout ce qui gravite autour." Sa certification lui a donc permis

d'acquérir des connaissances multiples et complémentaires qui garantissent la qualité de ses prestations. Professions libérales, petites ou grandes entreprises, quelle que soit leur taille, ses clients ne s'y trompent pas. "Pour mes clients, la certification MCDBA est avant tout un gage de qualité des prestations que je vais fournir", précise-t-il. En amont, lorsqu'il y a mise en concurrence, cela donne également à un client une meilleure visibilité pour sa prise de décision. Mis à part les cas de dépannage en urgence, où le client n'a pas le choix et doit parfois faire confiance aux compétences avancées par le prestataire, l'exigence de travailler avec un interlocuteur certifié s'affirme de plus en plus. Les SSII ou le secteur de la banque et de l'assurance sont même familiers de ce type de demande. Pour se maintenir à un niveau d'excellence, Charlie Dancoisne suit de près les évolutions du produit et veille constamment à la mise à niveau de ses certifications. Cela lui demande un travail intense et constant. "Afin d'optimiser la préparation, rien ne vaut les formations dédiées, cela permet d'économiser du temps et de la fatigue !", conseille-t-il après avoir testé la préparation en "autodidacte" et celle par le biais de formations agréées. Détenir des certifications Microsoft actualisées, outre le fait d'être un avantage concurrentiel certain, l'encourage à toujours plus de réactivité et d'efficacité. "Les certifications sont sans conteste des labels qui permettent de normaliser des compétences et sont par là même gage de qualité pour les prestations proposées par les personnes qui les détiennent", conclut-il. ■



Ma certification MCDBA est un gage de qualité apprécié et parfois exigé par mes clients.

de l'infrastructure, ainsi que la sécurisation des données.

Les fonctions concernées par cette certification sont des administrateurs, des analystes ou encore des développeurs de bases de données. La certification MCDBA nécessite de disposer d'au minimum un an d'expérience dans l'utilisation de Microsoft SQL Server. En France, 250 personnes certifiées MCDBA sont recensées. Les conditions d'obtention sont strictes. Le candidat doit détenir 4 examens sur l'infrastructure, le développement de bases de données, l'administration de bases de données, et un examen complémentaire.

Cursus en détail

Le cursus présenté est le cursus type recommandé pour les individus compétents sur Microsoft SQL Server 2000. Pour connaître l'intégralité des examens éligibles au cursus MCDBA, veuillez vous référer au site officiel Microsoft (www.microsoft.com/France/formation)

Examens Principaux (3 examens requis)	1 examen requis	Infrastructure	70-215 Installing, Configuring, and Administering Microsoft Windows 2000 Server
			70-290 Managing and Maintaining a Microsoft Windows Server 2003 Environment
			70-291 Implementing, Managing, and Maintaining a Microsoft Windows Server 2003 Network Infrastructure
	1 examen requis	Développement de Bases de Données	70-229 Designing and Implementing Databases with Microsoft SQL Server 2000 Enterprise Edition
	1 examen requis	Administration de Base de Données	70-228 Installing, Configuring, and Administering Microsoft SQL Server 2000 Enterprise Edition
Examen Complémentaire (1 examen requis)			70-216 Implementing and Administering a Microsoft Windows 2000 Network Infrastructure
			70-293 Planning and Maintaining a Microsoft Windows Server 2003 Network Infrastructure
			70-305 Developing and Implementing Web Applications with Microsoft Visual Basic® .NET and Microsoft Visual Studio .NET
			70-306 Developing & Implementing Windows-based Applications with Microsoft Visual Basic .NET & Microsoft Visual Studio .NET
			70-310 Developing XML Web Services & Server Components with Microsoft Visual Basic .NET & the Microsoft .NET Framework
			70-315 Developing and Implementing Web Applications with Microsoft Visual C#™ .NET and Microsoft Visual Studio .NET
			70-316 Developing & Implementing Windows-based Applications with Microsoft Visual C# .NET & Microsoft Visual Studio .NET
		70-320 Developing XML Web Services and Server Components with Microsoft Visual C# and the Microsoft .NET Framework	

InfoPath : une technologie prometteuse

Bon nombre d'entreprises ont toujours recours à des formulaires papier pour répondre à des besoins très divers. Ces formulaires doivent bien souvent être ressaisis manuellement (ou dans de plus rares cas, font l'objet d'une lecture optique) dans le système d'information, afin de pouvoir être exploités. Cela induit alors des charges de travail supplémentaires, et représente également un risque de perte d'information.

XML se présente comme une solution pour répondre à ce besoin, et le W3C propose à cet effet une spécification appelée XForms. La réponse de Microsoft pour rationaliser ce processus de collecte d'informations est basée sur le produit InfoPath 2003, un composant logiciel de la suite bureautique Office 2003 Entreprise. InfoPath 2003 est un outil de création de formulaires électroniques, destiné à améliorer les mécanismes de collecte d'informations au sein de l'entreprise, et qui repose sur des standards technologiques (XML, services web, etc.) pour favoriser l'intégration avec les systèmes informatiques existants. Avec InfoPath 2003, il devient alors possible de créer rapidement et simplement des formulaires dont le contenu pourra être publié sur Internet, archivé dans le back-office ou bien simplement enregistré sur le poste de l'utilisateur.

Les domaines d'applications pour un outil comme InfoPath 2003 sont nombreux puisqu'il permet de répondre à toutes les problématiques de saisie de formulaires existantes. En particulier, Microsoft a déjà su séduire l'industrie de la santé aux Etats-Unis, qui a adopté InfoPath pour les transferts de documents aux standards CDA (Clinical Document Architecture), un format d'échange de données médicales électroniques.

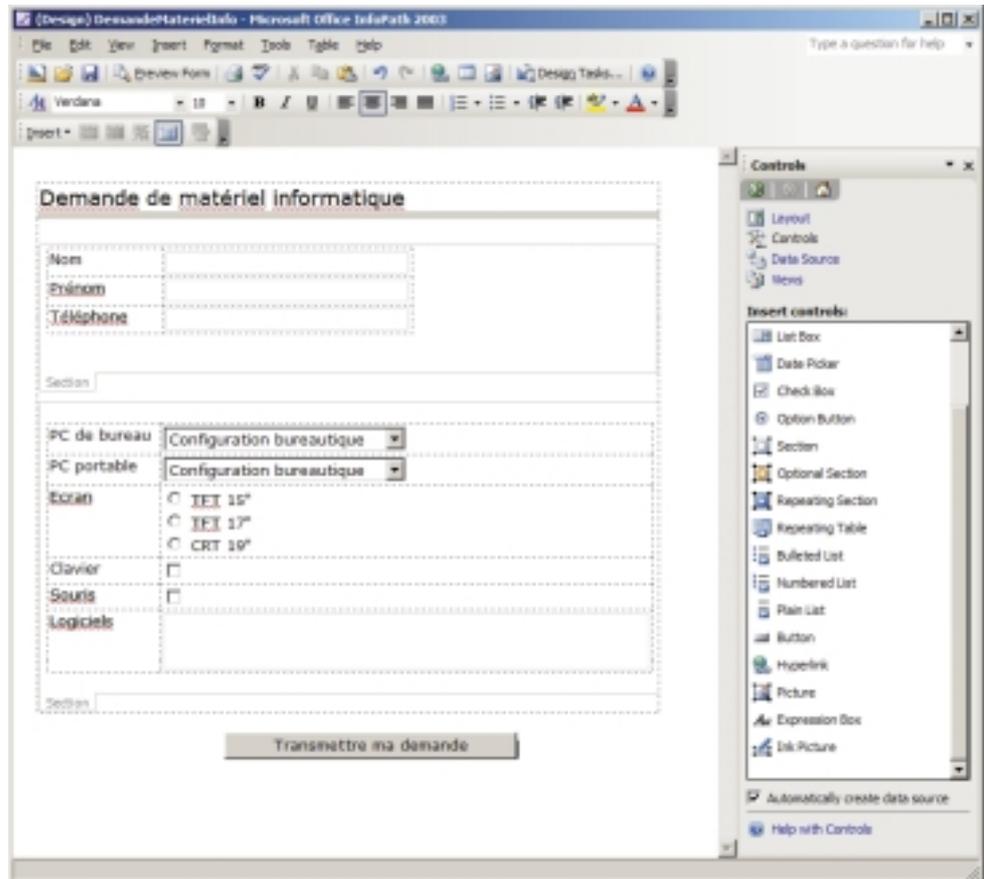


Figure 1 : Création d'un formulaire avec InfoPath 2003

L'utilisation d'InfoPath 2003

Ce qui surprend le plus lorsque l'on commence à utiliser InfoPath 2003, c'est sa grande facilité de prise en main. En effet, l'interface utilisateur tire parti des points forts de la suite Office. La création de formulaires est similaire à ce que peut proposer Access dans ce domaine, et en ce qui concerne leur mise en forme et la saisie d'informations, l'utilisateur retrouvera les fonctions les plus standard de Word ou FrontPage.

En effet, créer un formulaire ne prend que quelques minutes et pour cela on dispose d'une palette de composants suffisamment riche pour obtenir un résultat très élaboré. On retrouve d'abord tous les contrôles standard (cases à cocher, zones de texte, etc.), des contrôles plus orientés Web (lien hypertexte, images), et des contrôles optionnels, ou des listes répétitives pour gérer les choix multiples et les tableaux. Il existe même un contrôle qui

permet de récupérer sous la forme d'une image, des notes manuscrites saisies à partir d'un Tablet PC. (figure 1)

Pour chaque contrôle, il est possible de vérifier la cohérence de la saisie en créant des règles de validation sur les champs de saisie. Pour des traitements plus élaborés, InfoPath permet de faire appel à des fonctions écrites en JScript ou VBScript.

Des formulaires peuvent aussi être constitués à partir de sources de données existantes. Ainsi, on peut créer un nouveau template de formulaire, depuis une base de données Access ou SQL Server, un schéma XML (fichier XSD), ou un service web. Dans ce dernier cas, on peut spécifier si les traitements du formulaire sont de la simple consultation, de la mise à jour de données, voire même les deux en même temps.

Une fois que le formulaire a été créé, le créateur du template InfoPath doit le mettre à dis-

position dans un endroit accessible à tous ceux qui en ont besoin. Pour cela, il est possible de publier le template très simplement, via un assistant, sur un répertoire partagé accessible depuis le réseau local, sur un serveur Microsoft SharePoint, ou sur un serveur HTTP. Malheureusement, InfoPath 2003 ne gère pas la sécurité par lui-même (hormis une simple option permettant uniquement de désactiver les contrôles de modification des templates). Aussi, il faudra donc prendre en considération cette contrainte, afin de traiter les problématiques de sécurité associées aux formulaires au niveau des serveurs chargés d'héberger les templates.

Pour ce qui concerne la saisie des formulaires, l'utilisateur doit impérativement disposer d'InfoPath 2003 sur son PC, ce qui est un peu dommage, dans la mesure où cela limite le nombre de plates-formes cibles et notamment les applications nomades qui pourraient tirer un grand bénéfice d'une telle solution. A ce jour, il n'existe aucun moyen de saisir des formulaires créés avec InfoPath 2003 via Internet Explorer, depuis un PocketPC ou n'importe quelle autre plate-forme (Linux, MacOS, Palm, etc.). Cependant, les formats de fichiers d'InfoPath 2003 reposant totalement sur XML, gageons que des solutions venant d'éditeurs tiers ou du monde open source apparaîtront rapidement pour combler cette lacune.

Les mécanismes de publication d'InfoPath 2003

Structure d'un fichier InfoPath

InfoPath 2003 est entièrement bâti autour de

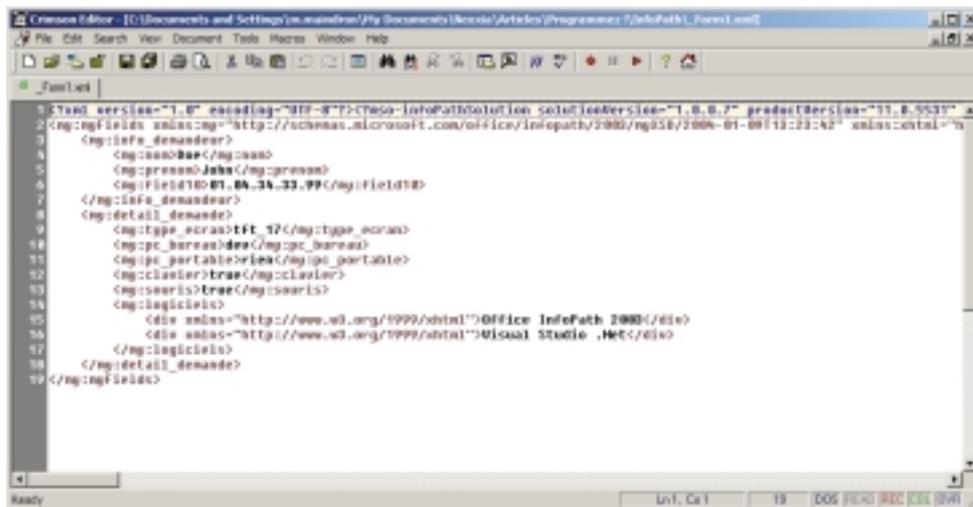


Figure 2 : Code XML généré lors de la saisie du formulaire

la technologie XML. Un template de formulaire (fichier .xsn) est en fait une association de plusieurs fichiers compressés au format CAB, tel que le fait OpenOffice avec le format zip.

Un fichier xsn se compose des documents suivants :

- **manifest.xsf** : décrivant la liste de tous les fichiers qui composent le template.
- **template.xml** : contenant les données en cours d'édition dans InfoPath.
- **myschema.xsd** : correspondant au schéma du formulaire qui a été créé, conformément aux recommandations W3C relatives à XML Schema.
- **script.js** : servant à stocker toutes les fonctions écrites avec les langages de scripting JScript ou VBScript

Le document InfoPath peut aussi contenir des

fichiers ajoutés spécifiquement tels que des images, des fichiers HTML ou XML.

Intégration d'InfoPath au système d'information

Lorsque le formulaire est validé par l'utilisateur, les informations saisies sont passées sous la forme d'un document XML, dont les noms des balises contenant les éléments sont ceux donnés aux contrôles qui le composent.

(Figure 2)

Ce document XML peut ensuite être traité de plusieurs manières et être, soit stocké dans une base de données Access ou SQL Server, via ActiveX Data Objects (ADO), transmis en XML/http, ou encore passé à un service Web. Il est aussi possible d'exporter les formulaires en tant que pages Web (mais ces dernières ne pourront être lues qu'avec Internet Explorer, car le format utilisé - .mht - est propriétaire à Microsoft), ou plus simplement en tant que fichier XML.

L'ouverture vers les services web d'InfoPath 2003 est un apport très important en termes d'interopérabilité et d'intégration avec les applicatifs qui composent un système d'information. Cette possibilité se révèle intéressante pour une entreprise, car elle autorise l'introduction de nouvelles tâches métier basiques très simplement au sein d'une architecture orientée services (SOA), sans pour autant avoir à engager des coûts de développement importants sur les applications clientes, puisqu'il suffira de créer un nouveau formulaire avec InfoPath 2003.

Les formulaires remplis manuellement puis

Quelques mots à propos de XForms :

XForms est une recommandation officielle du W3C depuis octobre 2003.

XForms est une application XML qui représente la prochaine génération de formulaires sur le Web.

Son principe consiste en la séparation d'un document XHTML standard en trois parties :

- le modèle de formulaire XForms,
- les données de l'instance du document,
- l'interface utilisateur.

Dès lors, XForms opère la distinction entre le contenu et la présentation, ce qui favorise la réutilisation et permet un typage fort du formulaire

Ainsi, XForms limite le nombre d'échanges entre une application cliente et le serveur, garantit une certaine indépendance vis-à-vis des terminaux, tout en réduisant l'effort de développement des écrans de saisie.

XForms n'est pas utilisable en tant que tel, il est destiné à être intégré à d'autres langages de balises comme XHTML ou SVG.

retranscrits ultérieurement dans une application peuvent dès lors, via InfoPath 2003, alimenter directement le back-office de l'entreprise et réduire le temps de traitement et d'exploitation de l'information. (Figure 3)

De la même façon, InfoPath 2003 peut être utilisé comme un service en amont d'un EAI tel que BizTalk Server. Ce dernier traite ensuite les flux XML qui proviennent d'InfoPath.

Fonctions de cache et de travail hors connexion

Une autre fonctionnalité intéressante d'InfoPath 2003 est la saisie de documents hors ligne. Pour cela, InfoPath 2003 met en cache les éléments qui constituent les templates des formulaires qu'il a déjà ouverts (schéma du formulaire, informations sur la validation des champs, etc.), ce qui permet à l'utilisateur de saisir des informations, même si le template n'est plus accessible depuis son emplacement d'origine. De même, si la source de données associée à un formulaire n'est plus disponible, suite à une perte de connexion par exemple, InfoPath va permettre à l'utilisateur de stocker les fichiers XML en local sur son poste de travail. Il sera alors possible d'envoyer ces données ultérieurement vers l'application cible, simplement en double cliquant sur le fichier, ce qui provoquera automatiquement l'ouverture d'InfoPath (une balise particulière dédiée à cet effet est ajoutée spécialement dans le fichier XML).

L'ouverture et le respect des standards

InfoPath 2003 est basé sur un grand nombre de standards associés à XML (WXS, DOM, XSLT). Dans son approche de la gestion de formulaires électroniques, Microsoft s'est donc bien conformé aux spécifications XML du W3C. Ce dernier propose d'ailleurs une recommandation pour la gestion des formulaires électroniques sur le web : XForms. Microsoft n'a pas retenu XForms pour InfoPath, tout simplement parce que sa spécification n'était pas encore validée par le W3C lors du début des développements du produit, et le fait de partir sur une base non validée aurait pu se révéler pénalisant par la suite.

D'autre part, InfoPath et XForms ne sont pas comparables en l'état : InfoPath est un produit, alors que XForms est une recommandation qui laisse place à de nombreuses implémentations possibles. Par contre, le format des données et la façon de les traiter peuvent eux, être comparés. Sur cet aspect, on peut noter

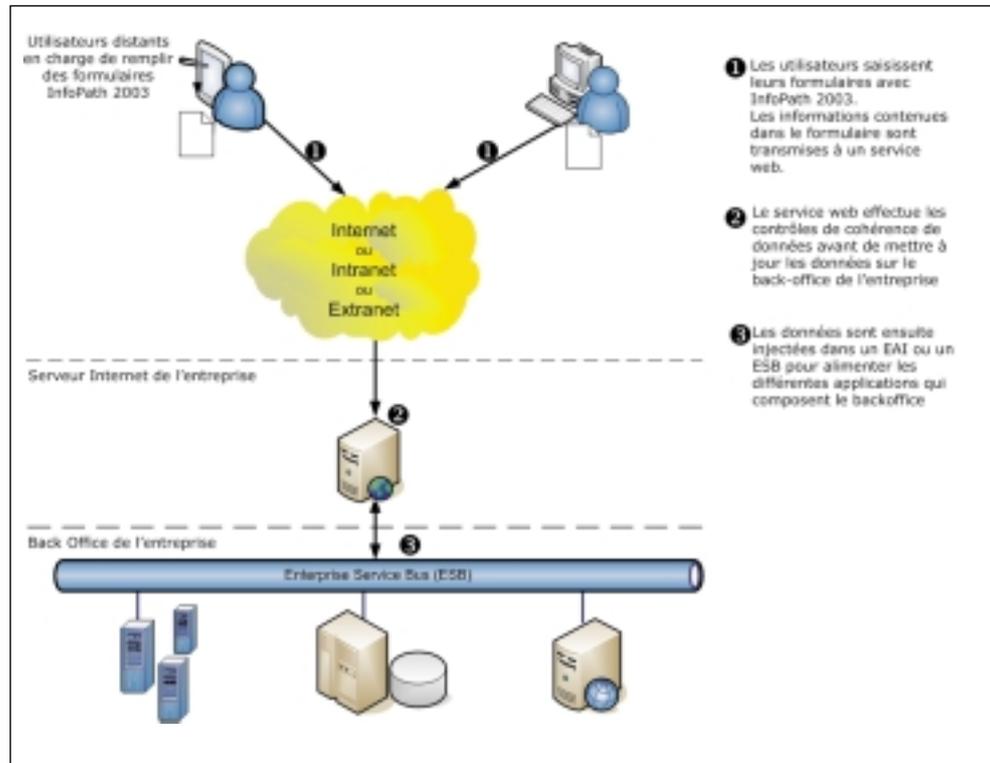


Figure 3 : Architecture d'interconnexion entre InfoPath 2003 et des services web

que le formalisme utilisé par Microsoft reste très proche des principes fondamentaux de XForms.

La limitation d'InfoPath 2003 se situe plutôt dans le fait que l'on doit utiliser la même interface pour créer et remplir les formulaires, ce qui restreint potentiellement son ouverture malgré la présence d'un standard on ne peut plus ouvert que XML.

Conclusion

InfoPath 2003 se présente vraiment comme une solution simple à mettre en place pour assurer la gestion de formulaires électroniques au sein d'une organisation.

Son interface utilisateur reste simple et familière, ce qui facilite grandement sa prise en main par les utilisateurs, et ses mécanismes de travail hors connexion sont un atout indéniable que ne peut pas proposer aujourd'hui une application web classique.

D'autre part, grâce à ses possibilités d'interaction avec des services web, il est possible de faire des formulaires InfoPath 2003 des éléments à part entière du système d'information, ce qui augmente grandement la flexibilité et l'ouverture du back-office d'une entreprise à des tâches unitaires et basiques qui en étaient jusqu'à présent écartées.

Cependant, on pourra regretter les limitations en termes d'accès à InfoPath 2003, plus particulièrement pour l'interface de saisie des formulaires, cette dernière n'étant accessible qu'à des PC ou des Tablet PC. Une ouverture vers Internet, via un module de saisie de formulaires depuis Internet Explorer ou un Pocket PC n'aurait pas été superflue et aurait pu contribuer de façon plus grande à l'intégration d'InfoPath 2003 dans l'entreprise.

C'est d'ailleurs ce dernier point qui constitue la différence fondamentale entre InfoPath et une application basée sur XForms qui pourra être ouverte à tous les types de terminaux, même les plus petits.

Pour résumer, InfoPath 2003 se révèle être une solution à la fois simple et puissante, qui permettra aux entreprises ayant des besoins importants d'intégration d'applications, de collecter, traiter et faire cohabiter plus efficacement leurs informations, sans nécessairement remettre en cause leur existant, ni engager des charges de développement importantes.

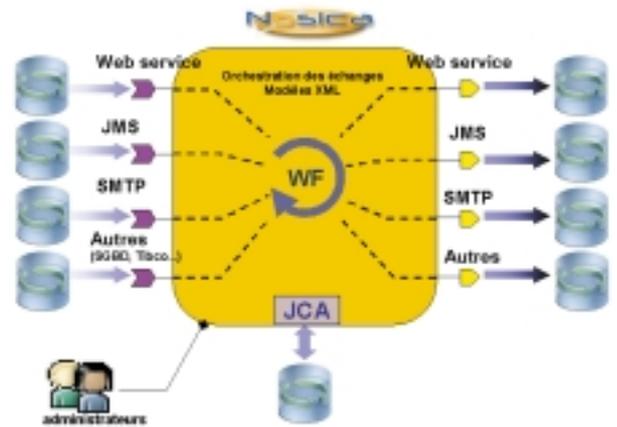
■ **Mickaël MAINDRON** est architecte technique chez Neoxia, intégrateur de systèmes spécialisé dans les nouvelles technologies de l'information

mickaël.maindron@neoxia.com



NoSICA : l'open source au service de l'intégration applicative

Si les offres commerciales dans le domaine de l'intégration applicative sont légion, force est de constater que, côté open source, la réponse à la problématique de l'intégration a été pendant longtemps éparpillée. Le serveur d'intégration NoSICA d'Open Wide constitue la première initiative de plate-forme d'intégration entièrement basée sur des composants open source.



NoSICA cible avant tout des projets d'intégration tactiques plutôt que stratégiques. L'approche est véritablement pragmatique et incrémentale : il s'agit de répondre pas à pas à chacune des demandes d'interconnexion d'une application à une autre. Cette approche est qualifiée de "bottom-up" en opposition à l'approche "top-down" qui débute par la modélisation des processus métiers pour aller vers la réalisation technique.

Pour que l'intégration soit un succès, il est indispensable que les interconnexions soient réalisées de manière transparente pour les applications et suivant un socle commun. Ce socle commun se doit d'être entièrement basé sur des standards, de manière à offrir le maximum d'interopérabilité avec les systèmes. Il se doit également d'être hautement communicant, en étant capable d'être contacté suivant une large diversité de moyens (comme JMS, RMI) et/ou de protocoles (tels que SOAP, HTTP, FTP, SMTP). Enfin le socle doit être fonctionnellement riche en proposant notamment quatre grandes briques : routage, transport, connecteurs et transformations.

La plate-forme J2EE s'impose comme un socle de base de choix pour faire tourner les briques évoquées plus haut : elle fournit en standard deux de ces briques, à savoir le mécanisme de transport et le mécanisme de connecteurs, mais elle apporte également une multitude d'autres services utiles, non seulement dans le cadre de l'intégration, mais aussi pour d'autres applications déployées sur le serveur. L'intérêt d'utiliser un serveur d'application J2EE est donc double :

il va constituer le moteur, le liant des briques d'intégration, mais il pourra aussi être réutilisable pour les applications et autres composants réalisés par l'entreprise.

Pour ce qui est du reste des fonctionnalités, beaucoup de composants open source répondent présent, mais peu semblent rencontrer trois contraintes : adaptabilité, robustesse, pérennité. En effet, chacun des composants se doit avant tout d'être aisément utilisé dans des environnements très variés : que dire si un des composants permettant de réaliser de l'intégration nécessitait lui-même un effort d'intégration ? Ensuite, les composants se doivent d'être fiables, même pour des charges élevées. Enfin, comme chacun le sait, le monde de l'open source est fortement évolutif. Chaque composant se doit donc d'être soutenu par une communauté active et réactive, gage d'une bonne visibilité concernant sa durée de vie.

Les grandes lignes de NoSICA étant tracées, les principales contraintes rappelées, passons à présent au cœur de la plate-forme.

Le moteur de workflow

Nous l'avons vu, le moteur de workflow est le centre névralgique de tout système dédié au final à orchestrer des flux. Au sein d'une architecture EAI, le workflow agit comme un hub (i.e. une gare de triage). Sa fonction est de recevoir des messages et de les renvoyer vers d'autres éléments du système, après les avoir éventuellement transformés.

Structurellement, le workflow est composé d'un moteur de workflow générique, adjoint

d'un fichier de règles (ou modèle de workflow) spécifique au système que pilote l'EAI. Le moteur de workflow va créer des instances de workflow qui vont suivre le modèle de workflow décrit.

Si la fonctionnalité d'un workflow est indispensable à la structure d'un EAI, il doit néanmoins absolument rester complètement indépendant des applications. Ainsi, lors de l'utilisation d'un moteur de workflow, il est indispensable de bien dissocier les données nécessaires au fonctionnement du moteur et à une instance d'un workflow, des données du système d'informations. De même, les traitements doivent être isolés du fonctionnement du moteur de workflow d'une part, et de l'application cible d'autre part.

Le moteur de workflow est le garant du couplage lâche entre applications. Une dépendance entre le moteur et une application annihilerait grandement son intérêt.

L'utilisation d'un tel moteur dans une architecture EAI offre plusieurs intérêts :

■ **Homogénéité des données traitées :** les données transitant par le moteur de workflow sont uniquement textuelles, faciles à interpréter et à traiter. Le travail d'encapsulation des données dans un format compatible avec "l'application d'entreprise" du système d'information est délégué en aval à un connecteur spécifique.

■ Fiabilité / réutilisabilité :

le moteur en lui-même est indépendant de la nature sémantique des messages; celle-ci est définie au niveau du fichier de règles. Ainsi, le moteur devient une brique indépendante qui, une fois conçue, peut être intégrée et réutilisée de façon générique dans divers projets d'intégration, indépendamment de la nature du métier concerné. Le réglage du moteur se fait à part, au niveau d'un fichier de règles spécifique à chaque EAI.

■ Indépendance fonctionnelle :

le moteur est structurellement indépendant des autres briques du système. Ainsi, tout en conservant une forte interopérabilité entre les briques, il est possible de changer l'une d'entre elles sans modifier le moteur. À l'inverse, il est possible de changer le moteur de workflow (suite à un upgrade, par exemple), sans que cela ait de conséquence globale sur le fonctionnement du système.

■ Programmation fonctionnelle aisée :

il est possible pour l'administrateur du système de modifier le fichier de règles — et donc le comportement de l'EAI — sans avoir à modifier le code du moteur ou d'un autre composant. Ainsi, on déporte la responsabilité du traitement des flux applicatifs : d'un codage spécifique à l'application elle-même (par ex. EJB), on est passé à un principe de description, via un mécanisme générique (on ne code plus, on décrit). Enfin, cette description fonctionnelle peut se faire par l'intermédiaire d'une interface graphique, générant automatiquement le modèle de workflow en format XML, sans que l'administrateur ait à connaître ce format.

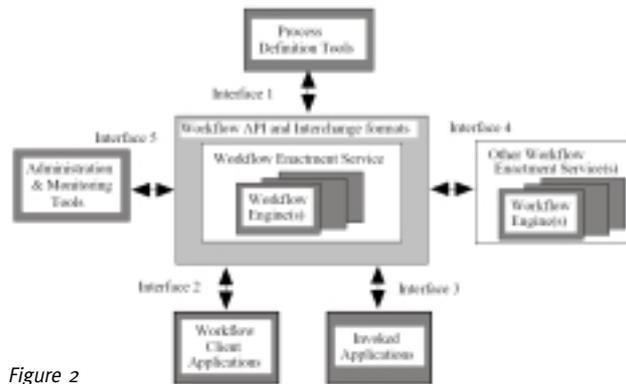


Figure 2

Le moteur de workflow, se trouvant au centre de la plate-forme d'intégration, il se doit d'être extrêmement interopérable. Le Workflow Management Coalition (WfMC) a bien compris cette nécessité. Le WfMC est un consortium regroupant plus de 300 entreprises, parmi lesquelles on retrouve des ténors tels que IBM, BEA mais aussi des éditeurs spécialisés comme W4. Ce consortium est à l'origine d'une spécification concernant les systèmes de workflow. Cette spécification se découpe en 5 interfaces (voir figure 2) qui identifient bien les différentes interactions du moteur avec son environnement.

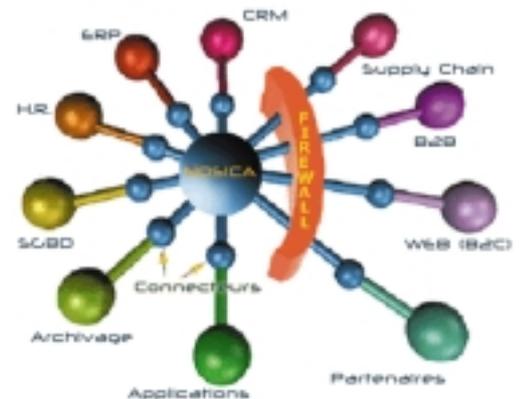
Aujourd'hui plusieurs projets Open Source ont choisi d'implémenter une ou plusieurs interfaces de la spécification WfMC. Le consortium ObjectWeb incube à lui seul 3 projets de systèmes de workflow plus ou moins conformes aux spécifications WfMC. Open Wide a choisi d'utiliser l'un deux, nommé Shark pour de multiples raisons. Tout d'abord, Shark est entièrement basé sur la spécification WfMC, sans aucune extension propriétaire. Ceci le rend très fortement interopérable avec d'autres outils. De plus, Shark est indépendant de son environnement d'exécution et se présente sous forme d'une librairie, ce qui lui permet de tourner indépendamment d'un serveur d'application. Shark possède une architecture modulaire définissant des interfaces entre

ses briques comme la persistance, l'authentification, les outils de scripting. Enfin, il permet une conception et un monitoring graphique à travers l'outil JaWE (Java Workflow Editor).

Communications

NoSICA est capable de recevoir et d'émettre trois grands types de messages à travers des composants divers.

Tout d'abord, la partie Web Services est assurée par une brique très reconnue dans le monde Java et proposée par Apache : Axis 1.1. Cette implémentation de la spécification SOAP 1.1 et 1.2 bénéficie d'une architecture élégante et souple. Elle découple bien les différents aspects de la chaîne de traitement des messages SOAP ce qui permet de moduler et d'étendre à souhait cette chaîne. Axis supporte les protocoles HTTP, comme d'autres implémentations Web Services, mais aussi SMTP et JMS. Le support des Web Services est particulièrement intéressant, car il permet de rendre la plate-forme interopérable avec les plates-formes non écrites en java.



Ensuite, l'échange de messages JMS est assuré à travers la brique JORAM fournie par ObjectWeb et constituant l'implémentation JMS du serveur d'application open source JOnAS, ou la brique JBossMQ, implémentation JMS du serveur d'application, également open source JBoss. Ces briques permettent l'envoi et la réception de messages asynchrones, soit en mode point à point, soit en mode publish-subscribe (un émetteur, n récepteurs). Ce mode de

A propos

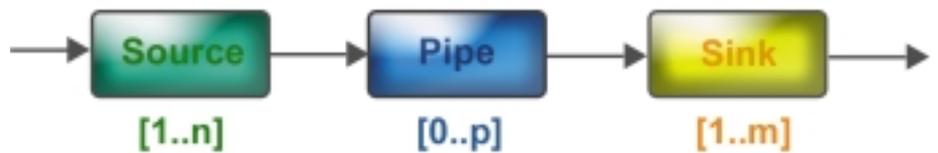
Open Wide, société de services spécialisée dans l'intégration de composants open source a eu l'idée de NoSICA dès sa création, à la suite d'un constat simple : devant l'explosion de la demande de plates-formes d'intégration, le monde de l'open source brillait par son absence. Pourtant, les briques techniques indispensables pour réaliser une telle plate-forme étaient bien là et n'avaient plus à démontrer leur richesse fonctionnelle, leur fiabilité et leur robustesse. Le cahier des charges du projet NoSICA, soutenu par l'ANVAR, est donc simple : mettre en commun toutes les briques open source qui concourent au montage d'une plate-forme EAI. Le terme EAI a cependant de multiples connotations et il est important de préciser quel est le rôle de NoSICA.

communication est souvent privilégié, car il est fiable et très simple à mettre en oeuvre. De plus, JORAM est également utilisable de manière autonome, ce qui permet de délocaliser ou de répartir les files de messages.

Enfin, l'échange de messages par mail est assuré en standard par tout serveur d'application J2EE à travers l'API JavaMail accompagnée de JAF (Java Activation Framework). De plus la toute nouvelle spécification J2EE 1.4 permettra de développer des EJB orientés message, capables d'écouter des serveurs mails.

Connecteurs

Comme nous l'avons déjà précisé, si la transformation des messages transitant dans la plate-forme constitue avec le moteur de workflow un point central, cette transformation ne doit pas être réalisée par le moteur de workflow lui-même. En cantonnant cette transformation dans des connecteurs, le moteur de workflow et les applications sont rendus indépendants. OpenAdaptor constitue un frame-



work de choix pour la partie connecteur. Il permet de constituer des chaînes de transformation à partir de briques réutilisables préexistantes ou écrites pour un besoin spécifique (figure 4). Une chaîne est constituée d'une source, d'un pipe de transformation chaînable et d'un émetteur (sink). Pour chacun de ces types d'objets, un bon nombre d'implémentations sont fournies en standard. Parmi ces implémentations, on trouve notamment des sources et des émetteurs pour Tibco Rendezvous, IBM MQseries, Oracle et Sybase, des pipes d'encryptage et de transformation XSL. Comme tous les autres composants sélectionnés pour NoSICA, OpenAdaptor se présen-

te sous forme de librairie et est donc capable de fonctionner dans des environnements très différents et notamment de façon autonome ou même, si nécessaire, embarqué dans une application.

Pour aller plus loin

<http://www.openwide.fr/NoSICA>

<http://www.openadaptor.org>

<http://forge.objectweb.org/projects/shark/>

<http://forge.objectweb.org/projects/joram/>

■ **Fabrice Dewasmes**

Architecte J2EE chez Open Wide



Introduction à la programmation en Java

multithread

Les ordinateurs gagnent sans cesse en vitesse d'exécution et les utilisateurs en impatience ! Les applications bien conçues se doivent de donner le sentiment à l'utilisateur qu'il n'est pas contraint d'attendre inutilement. Le thread est l'outil qu'il vous faut.

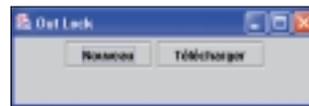
De nos jours, on trouverait inacceptable de ne pas pouvoir utiliser plusieurs applications en même temps, sous son système d'exploitation préféré. Pourtant, c'était le cas il n'y a pas si longtemps encore, à l'époque du laborieux MS-DOS. Désormais, tous les systèmes d'exploitation modernes sont multitâches. Ce terme signifie qu'il est possible de lancer plusieurs applications "simultanément". Chaque application dispose de sa pile, de ses variables internes, son code, etc. Si l'ordinateur est multiprocesseur, un processeur est, si possible, alloué à chaque application. Si ceci n'est pas possible, ou bien si tous les processeurs sont occupés, le système d'exploitation se charge de planifier les exécutions et attribue à chaque tâche une portion de temps processeur, donnant ainsi l'illusion d'exécutions simultanées.

Qu'est-ce qu'un thread ?

Le thread reprend le concept d'exécution répartie, non plus au niveau du système d'exploitation, mais au sein des applications elles-mêmes. Il s'agit cette fois de donner à l'utilisateur le sentiment qu'il peut effectuer (ou plutôt faire effectuer) plusieurs choses simultanément à l'intérieur d'une application. Prenons par exemple un logiciel de messagerie. Outlook sous Windows, ou encore KMail sous Linux et KDE. Il n'est pas question de devoir attendre passivement que le logiciel télécharge le courrier depuis un serveur pop3, sans pouvoir en même temps écrire un mot doux à votre petite amie (déjà qu'elle est fâchée de vous voir en permanence scotché à votre machine... :-). Ainsi, le téléchargement du courrier s'effectuera dans un thread. Comme pour le multitâche, un thread peut se voir attribuer un processeur ou un temps d'exécution. Ceci est du ressort du système d'exploitation et/ou du runtime du langage utilisé. En revanche, pour le programmeur la différence et aussi la difficulté sont que les threads ne sont pas cloisonnés comme le sont les processus. Au contraire, ils partagent code et variables, ce qui peut rapidement aboutir à des situations conflictuelles si l'on n'y prend pas suffisamment garde. Nous allons donc par cet article, entamer une initiation à la programmation multithread, du plus simple au plus compliqué. Nous en verrons tout d'abord les bases, puis certaines subtilités. Nous avons choisi le langage Java, car il dispense largement (mais pas totalement comme nous le verrons par la suite, les impatients peuvent s'informer relativement aux 'green threads' et aux 'native threads') de connaître le système d'exploitation sous-jacent, connaissance qui serait requise en C ou C++, et son orientation objet aide à l'organisation du code. Enfin, Java est tel que ce que nous dirons s'applique ou se transpose facilement à d'autres langages.

Le problème

Simulons donc une application de messagerie électronique. Le téléchargement du courrier est remplacé par une routine de compte à rebours affiché sur la sortie standard. Un bouton lance celui-ci. Un autre bouton simule l'écriture d'un nouveau mail, également sur la sortie standard.



Notre vrai faux logiciel de messagerie simulée :-)

Le code partiel de cette application unique au monde est donné ci-dessous :

```
public class DemoThread1 extends JFrame {

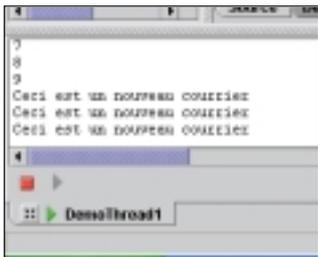
    public static void main(String[] args) throws HeadlessException {
        DemoThread1 demoThread = new DemoThread1();
        demoThread.setSize(300, 100);
        demoThread.setVisible(true);
    }

    void this_windowClosing(WindowEvent e) {
        System.exit(0);
    }

    void jButton1_actionPerformed(ActionEvent e) {
        System.out.println("Ceci est un nouveau courrier");
    }

    void jButton2_actionPerformed(ActionEvent e) {
        for(int i = 0; i < 10; i++)
        {
            System.out.println(i);
            try{
                Thread.sleep(1000);
            }
            catch (InterruptedException ie) {}
        }
    }
}
```

Le code complet de tous les exemples se trouve sur le Cd-Rom. Hormis les méthodes d'initialisation, ce code se compose essentiellement de trois gestionnaires d'événements. Un pour chaque bouton et un pour l'arrêt de l'application à la suite de la fermeture de la fenêtre. Normalement, on s'attend à ce que les événements en provenance de la souris soient dispatchés vers les méthodes des gestionnaires. Voyons ce qu'il en est en essayant. Cliquez sur le bouton 'télécharger'. Le compte à rebours commence et s'affiche sur la sortie standard. Jusque-là tout va bien. Mais si pendant le déroulement du compte à rebours nous cliquons sur le bouton 'Nouveau' nous ne voyons rien apparaître sur la sortie standard, du moins au moment attendu. Nous ne pouvons pas non plus fermer la fenêtre. En fait, les sorties correspondant au bouton 'Nouveau' apparaissent après l'arrêt du compte à rebours, comme si elles avaient été mises en attente, ce qui est réellement le cas. Car,

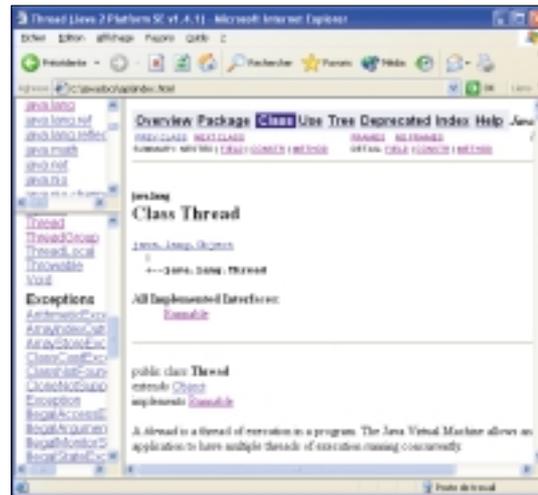


Oups! Voilà qui ne donne guère une impression d'exécution simultanée

lorsque qu'une application est lancée, la machine virtuelle Java fait exécuter le code dans... un thread, qui est bien sûr unique à ce stade. La JVM capte bien les événements de la souris, mais puisque les gestionnaires appartiennent au même thread, l'exécution d'une méthode ne peut avoir lieu que lorsque la précédente a rendu la main. Au passage, on remarquera que le compte à rebours est temporisé avec la méthode statique `Thread.sleep` qui est bien une méthode de `Thread`.

Une première solution

Ce que nous devons obtenir de la JVM est qu'elle exécute le compte à rebours dans un thread différent du thread initial qui est le thread principal de l'application. Ainsi, le gestionnaire d'événement du bouton 'Télécharger' rendra la main au thread principal qui sera alors capable de traiter sans attendre un clic de souris sur le bouton 'Nouveau' ou sur le bouton de fermeture de la fenêtre. Dans un premier temps, nous écrivons une classe `CountDown` qui dérive de la classe `Thread`, classe subtile dont les méthodes doivent être étudiées avec soin ce que nous ferons petit à petit. Dans le cas d'une dérivation, la documentation nous informe que le code à exécuter doit résider dans une méthode 'run' redéfinissant la méthode de la classe de base. Voici ci-contre (encadré 2) le code partiel de notre nouvelle application. Nous y trouvons notre nouvelle classe. Par ailleurs, le seul changement notable concerne le contenu du gestionnaire d'événement du bouton 'Télécharger'. Nous yinstancions notre classe `CountDown`, puis nous demandons au thread de commencer son travail par l'appel à la méthode `start`. Ceci a pour effet que la méthode `run` de l'instance sera exécutée dans un thread séparé. Attention, la méthode `run` ne doit pas être invoquée directement. Si on le fait, ce qui est possible dans l'absolu, alors son exécution aura lieu dans le thread courant et nous ne serions pas plus avancés qu'avec le premier exemple. C'est donc toujours à la JVM de s'acquiescer de cela. Par contre, cela montre bien qu'en programmation multithread, le code est partagé, contrairement à ce qui se passe avec les



La classe Thread et l'interface Runnable font partie du package java.lang.

Premier exemple d'application multithread

```
class CountDown extends Thread {
    public void run() {
        for(int i = 0; i<10; i++)
        {
            System.out.println(i);
            try{
                Thread.sleep(1000);
            }
            catch(InterruptedException ie) {}
        }
    }
}

public class DemoThread2 extends JFrame {

    public static void main(String[] args) throws HeadlessException {
        DemoThread2 demoThread2 = new DemoThread2();
        demoThread2.setSize(300, 100);
        demoThread2.setVisible(true);
    }

    void this_windowClosing(WindowEvent e) {
        System.exit(0);
    }

    void jButton1_actionPerformed(ActionEvent e) {
        System.out.println("Ceci est un nouveau courrier");
    }

    void jButton2_actionPerformed(ActionEvent e) {
        CountDown cd = new CountDown();
        cd.start();
    }
}
```

processus, ainsi que nous le disions au début de l'article. Ayant travaillé dans les règles de l'art, nous avons l'illusion de deux exécutions parallèles, ce qui est ce que nous souhaitons (figure 4). Une dernière remarque. Le compte à rebours est temporisé de la même manière que précédemment, avec la méthode sleep. Celle-ci agit toujours sur le thread depuis lequel elle est invoquée. On n'en attendait pas moins d'elle.

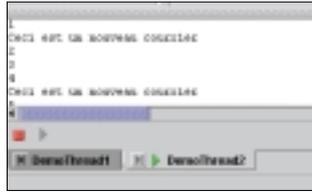


Figure 4 : Grâce à la classe Thread, Si dériver une classe de la classenous parvenons à nos fins.

L'interface Runnable

Si dériver une classe de la classenous parvenons à nos fins. Thread est la première chose qui vient à l'esprit, ce n'est pas forcément la plus judicieuse en Java. Si l'on considère que l'héritage est une relation "est un", on peut difficilement dire qu'un compte à rebours est un thread. En outre, cela pourrait conduire à des difficultés dans la conception d'une hiérarchie de classes. Prenons par exemple une classe 'Balle'. Une sous-classe pourrait être une classe BalleAnimee contenant une méthode destinée à être exécutée dans un thread séparé. Toutefois, 'BalleAnimee' devra dériver de 'Balle' et elle ne pourra pas dériver en plus de Thread, puisque Java ne supporte pas l'héritage multiple. La solution tient en l'implémentation de l'interface Runnable, plutôt qu'en la dérivation. Notre classe devient :

```
class CountdownRunnable implements Runnable {
    public void run() {
        for(int i = 0; i<10; i++)
        {
            System.out.println(i);
            try{
                Thread.sleep(1000);
            }
            catch(InterruptedException ie) {}
        }
    }
}
```

Comment lancer le Thread ? En effet, l'interface Runnable ne contient pas de méthode start et du coup notre classe n'en contient pas non plus. Nos lignes de code :

```
CountDown cd = new CountDown();
cd.start();
```

ne sont donc plus valables. Nous résolvons le problème en instanciant une classe Thread avec un constructeur acceptant une référence sur un objet Runnable, ici notre classe de compte à rebours, comme ceci :

```
Thread cdt = new Thread(new CountdownRunnable());
cdt.start();
```

Une autre solution, sans doute la meilleure, est de perfectionner notre classe de compte à rebours en la dotant d'une méthode start et, c'est l'occasion, d'une variable permettant de vérifier si la classe est déjà instanciée. Voici notre nouvelle classe :

```
class CountdownRunnable1 implements Runnable {

    public void start() {
        if(t == null) {
            t = new Thread(this);
            t.start();
        }
    }

    public void run() {
        for(int i = 0; i<10; i++)
        {
            System.out.println(i);
            try{
                Thread.sleep(1000);
            }
            catch(InterruptedException ie) {}
        }
        t=null;
    }

    static Thread t;
}
```

Et pour lancer le thread depuis la classe principale, nous revenons à :

```
CountDownRunnable1 cd1 = new CountdownRunnable1();
cd1.start();
```

Pour terminer, remarquons que la classe principale pouvait fort bien implémenter elle-même l'interface Runnable. Une telle solution est valable quand l'application n'a qu'une seule opération à effectuer dans un thread.

Un peu de terminologie

Java considère qu'un thread peut avoir 4 états : nouveau, exécutable, bloqué ou mort. Un thread est considéré comme nouveau lorsqu'il vient d'être instancié, mais qu'il n'a pas encore exécuté le moindre code contenu dans run, ou autrement dit, tant que la méthode start n'a pas encore été invoquée. Dès que start est invoquée, le thread bascule vers l'état exécutable... ce qui ne veut pas dire qu'il s'exécute réellement. En effet, il peut éventuellement être en attente d'un créneau de temps pour travailler. La JVM ne fait pas le distinguo entre exécutable ou réellement en cours d'exécution.

Un thread est bloqué lorsque sleep a été invoquée ou encore la méthode wait de la classe Object. Il est encore bloqué s'il effectue une opération d'entrée/sortie... bloquante. La lecture de données dans un socket (Cf Java et les Sockets -- Programmez! 59) en est un exemple. Un thread est également bloqué s'il essaie de verrouiller un objet qui l'est déjà par un autre thread. Ce qui nous amène doucement vers les subtilités de la programmation multithread. C'est pourquoi, dès le mois prochain nous aborderons la programmation avancée des threads: interruption d'un thread, synchronisation, priorités, et autres bonnes choses. À bientôt.

■ Frédéric Mazué - fmazue@programmez.com

Sur le
CD ROM

Mettre en oeuvre XML-RPC avec

Python et Zope

L'atout majeur du protocole XML-RPC est sa simplicité. La simplicité est également un atout majeur de Python et de Zope. Quand tout ce petit monde collabore, mettre en place un Web Service relève du jeu d'enfant.

Les besoins en informatique distribuée vont grandissant. Il n'est pas rare de voir l'amateur éclairé posséder son petit Intranet. Quant à l'entreprise, en raison du faible coût des PCs, il peut être plus intéressant pour elle d'avoir une architecture distribuée, éventuellement tolérante aux pannes plutôt qu'un mainframe. L'amateur qui voudra écrire un petit outil d'administration pour son réseau sera à juste titre rebuté par les redoutables et indigestes spécifications de poids lourds, tels que EJB ou CORBA, en supposant même que ceux-ci conviennent à cet emploi. Les professionnels y regarderont peut être à deux fois avant d'investir dans des solutions complexes, onéreuses et demandant un personnel très qualifié, et tout le monde souhaitera ne pas être ennuyé par des problèmes de pare-feu. C'est dans cette optique que le protocole SOAP, par exemple, a été conçu. Aujourd'hui, nous allons nous intéresser à un protocole encore plus simple que SOAP: XML-RPC. Notre intention est de montrer qu'avec ce protocole très simple, utilisé avec le langage Python, lui aussi très simple, et avec le génial Zope, développer un outil d'administration réseau ou installer un Web Service est à la portée du premier venu.

Le protocole XML-RPC

L'idée de base est toute simple : pouvoir invoquer depuis une machine A une procédure, méthode ou fonction résidant sur une machine B. Ceci explique une partie de l'acronyme dans lequel RPC signifie "Remote Procedure Call", ou autrement dit, appel de procédure distant. L'appel de la procédure est intégré dans une requête HTTP POST qui transite sur l'Intranet ou Internet, comme toute autre requête HTTP, ce qui explique pourquoi nous ne rencontrerons normalement pas de problème avec les pare-feu. Enfin le nom de la procédure invoquée et les paramètres qui lui sont transmis sont emballés dans des balises XML. La valeur de retour de la fonction également. Nous comprenons donc pourquoi notre protocole s'appelle XML-RPC. Vous pouvez en consulter les spécifications à <http://www.xml-rpc.com>. D'autre part, si vous souhaitez approfondir la question et tout connaître de la constitution d'une requête http, vous pouvez consulter le document RFC 1945 téléchargeable (par exemple) à <http://www.faqs.org/rfcs/rfc1945.html>.

Un exemple de requête XML-RPC

Cet exemple provient du site www.xml-rpc.com. Examinons le :

```
POST /RPC2 HTTP/1.0
User-Agent: Frontier/5.1.2 (WinNT)
Host: betty.userland.com
Content-Type: text/xml
Content-length: 181

<?xml version="1.0"?>
<methodCall>
  <methodName>examples.getStateName</methodName>
  <params>
    <param>
      <value><i4>41</i4></value>
    </param>
  </params>
</methodCall>
```

Il s'agit d'une requête POST. Le nom de domaine est betty.userland.com. L'URI fait que le serveur va router l'appel sur l'application /RPC2. Ensuite, toujours dans l'en-tête, le serveur est informé de la teneur du contenu (text/xml), contenu que le pare-feu doit bien entendu accepter, cela tombe sous le sens. Pour en terminer avec l'en-tête, nous voyons que la longueur du contenu doit être spécifiée et que la valeur doit être correcte.

L'avantage incontestable d'XML est qu'il est intuitif. Nous comprenons donc facilement que sera invoquée la méthode getStateName de l'objet 'examples' ou bien la méthode de nom examples.getStateName, selon le mode d'implémentation du serveur. Cette méthode reçoit un paramètre sous la forme d'un entier codé sur 4 octets (balise <i4>). Les balises possibles sont énumérées à xml-rpc.com. Signalons que xml-rpc accepte les structures complexes.

Par exemple :

```
<array>
<data>
<value><i4>12</i4></value>
<value><string>Egypt</string></value>
<value><boolean>0</boolean></value>
<value><i4>-31</i4></value>
</data>
</array>
```

Où le mot array n'est pas très bien choisi. Puisque les types sont différents, c'est plutôt à une liste que nous avons affaire.

En ce qui concerne la réponse, y compris en cas d'erreur, celle-ci est constituée de manière similaire à la requête. Le lecteur voudra bien se reporter au site officiel. Le peu que nous avons dit du protocole est suffisant pour nous montrer qu'il est indépendant des machines et même des langages et qu'il fera donc merveille dans des environnements hétéroclites. En outre, il est facile à implémenter. Par exemple, le lecteur pourrait facilement écrire un client xml-rpc en Java en se basant sur l'article "Java et les Sockets" (Programmez! N° 59). Des implémentations pour la plupart des langages courants existent déjà sur Internet. Des serveurs comprenant le protocole également. Avec en tête le fameux Apache, allié par exemple à Php.

XML-RPC et Python

Nous choisissons ici d'employer Python car nous pensons que ses types de haut niveau font merveille dans ce contexte (et ailleurs). La distribution Python (nous avons utilisé Python 2.2) vient avec les modules nécessaires. Ou, si par extraordinaire, ce n'était pas le cas de votre Python, vous pourrez télécharger à <http://www.pythonware.com/products/xmlrpc/>. L'archive ne pèse que 20K ce qui est plus sympathique que les 40 Mo de l'implémentation CORBA Omniorb, pour prendre un élément de comparaison.

La grande force de Python, qui est un langage de haut niveau, est de masquer les détails fastidieux. Le module xmlrpclib contient tout ce qu'il faut pour constituer une requête http en bonne et due forme. Cette opération est automatiquement effectuée en sous main, à chaque appel de méthode. Grâce à ce module nous écrivons notre premier client. Comme pour la requête d'exemple ci-dessus, il s'agit d'interroger le serveur `betty.userland.com`. La méthode `examples.getStateName` reçoit un entier et retourne le nom de l'état américain correspondant. Voici le code :

```
#!/usr/bin/env python

from xmlrpclib import *

server = ServerProxy("http://betty.userland.com")

try:
    print server.examples.getStateName(41)
except Error, v:
    print "ERREUR", v
```

Un client plus évolué

Un serveur XML-RPC bien élevé propose normalement trois méthodes permettant de connaître ses possibilités. Ces méthodes sont :

Méthode	Description
<code>system.listMethod</code>	Retourne une liste des méthodes publiques.
<code>system.methodSignature</code>	Retourne un tableau de signatures pour une méthode, une signature consistant en une liste de types de paramètres.
<code>system.methodHelp</code>	Retourne une chaîne d'aide associée à la méthode.

L'éditeur bien connu O'Reilly propose sous le nom de Meerkat, un serveur XML-RPC qui implémente tout cela dans les règles de l'art. Nous écrivons alors un client un peu plus élaboré :

```
#!/usr/bin/env python

from xmlrpclib import *

meerkatsvr = ServerProxy("http://www.oreillynet.com/meerkat/xml-rpc/server.php")
print meerkatsvr.system.listMethods()
print "_____"
print meerkatsvr.system.methodHelp('meerkat.getItems')

print meerkatsvr.meerkat.getItems(
    {'search':'/[Pp]ython/', 'num_items':5, 'descriptions':0})
```

Le résultat de l'exécution de ce script est montré figure 1. J'ai lancé ce script depuis l'insusable XEmacs. Cet exemple montre comment obtenir la liste des fonctions disponibles, comment obtenir l'aide sur la fonction `meerkat.getItems` et enfin invoquer la dite méthode. Pour ce dernier point, on notera que le nom des paramètres est associé aux valeurs et que ceci se fait en Python avec un dictionnaire, tout naturellement.

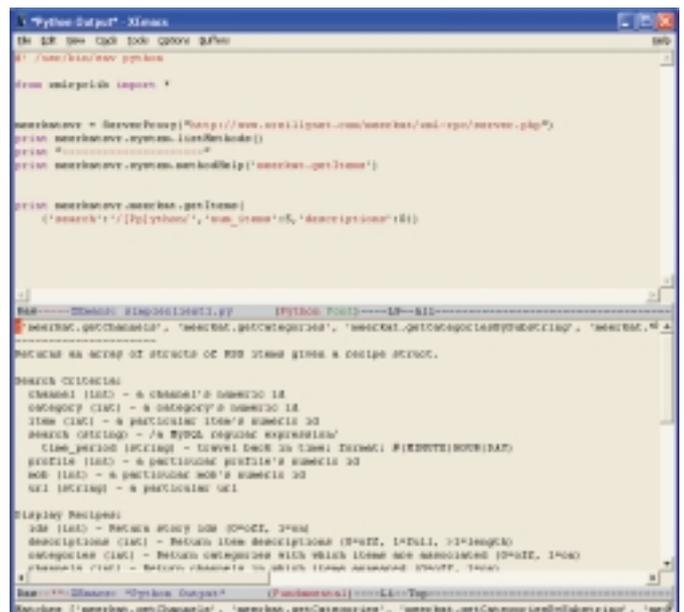


Figure 1: Un client xml-rpc Python lancé depuis XEmacs.

Un Serveur XML-RPC

Ecrire des clients est sans doute une bonne chose, mais écrire un serveur est beaucoup plus amusant. C'est donc ce que nous allons faire maintenant. Nous écrivons un convertisseur Franc/Euro avec une méthode pour chaque type de conversion. Rien ne vous empêche de modifier cet exemple rudimentaire pour en faire un outil d'administration de vos postes à distance. Le serveur tournera sur la machine 'Lune' de mon Intranet (adaptez le code à vos besoins) et écoutera le port 8000.

```
import SimpleXMLRPCServer

done = 0
def terminate(password):
    global done
    if password == 'sesame':
        done = 1
    # Juste pour éviter la levée d'une exception
    # qui polluerait la sortie sur le
    # terminal du serveur
    return done

class Convertisseur:
    def _fonctionprivée(self):
        pass

    def EtoF(self, value):
        return value * 6.55957

    def FtoE(self, value):
        return value / 6.55957

server = SimpleXMLRPCServer.SimpleXMLRPCServer(("lune", 8000))
server.register_instance(Convertisseur())
# si Python 2.3
# server.register_introspection_functions()

#Plutôt que d'invoquer cette méthode
#server.serve_forever()

#proposer un moyen de quitter proprement
server.register_function(terminate)

while not done:
    server.handle_request()
```

Commentons ce code. Nous créons une classe baptisée 'Convertisseur', qui contient d'abord une méthode privée (ne faisant rien), histoire de montrer que toute méthode dont le nom commence par _ est privée. Puis viennent les méthodes de conversion proprement dites. Enfin on instancie le serveur dans lequel on enregistre la classe et l'affaire est réglée. Comme il est également possible d'enregistrer une fonction indépendante, c'est l'occasion de montrer qu'il est peut être mieux d'écrire une fonction pour arrêter le serveur qui tourne perpétuellement si on le lance classiquement par l'appel à `serve_forever`. C'est le rôle de notre

fonction terminée. Dernière remarque. Cette méthode retourne une valeur qui n'est pas utilisée. Dans notre exemple, ce retour a uniquement pour but d'éviter une levée d'exception, parce que nous ne voulons pas polluer inutilement la sortie sur le terminal sur lequel le serveur tourne. Pour XML-RPC une fonction qui ne retourne pas de valeur n'a pas de sens, ce qui est, il faut le reconnaître plutôt logique. Dans un tel cas SimpleXMLRPCServer lève une exception, alors nous contournerons le problème en retournant une valeur "bidon" pour notre exemple, mais qui pourrait fort bien servir de booléen si le client voulait être assuré qu'il a bien réussi à fermer le serveur. D'un autre côté, et c'est toujours parfaitement logique, une fonction privée n'est pas tenue de retourner une valeur. Voici un client pour notre serveur :

```
from xmlrpclib import *

s = Server("http://lune:8000")

print s.EtoF(1)
print s.FtoE(6.55957)
s.terminate('sesame')
```

Bien sûr, si l'on veut créer un Web Service supportant la charge, nous avons besoin d'un serveur un peu plus solide. Passons donc à

Zope et XML-RPC

Non content d'être un serveur Web/serveur d'application génial, Zope a l'excellente idée de supporter xml-rpc en natif. Construisons ensemble notre Web Service 'convertisseur'. Créez un objet 'Folder' contenant deux scripts (respectivement identifiés `etof` et `feto`). Vous pouvez évidemment leur donner en surplus un titre plus parlant et un autre 'Folder' nommé `system` (figure 2). Le code des deux scripts est respectivement :

```
return value * 6.55957
```

et

```
return value / 6.55957
```

On peut difficilement faire plus simple. Notons qu'il faut configurer les scripts pour qu'ils reçoivent un paramètre du nom de 'value', comme montré sur la figure 3. Notre Web Service est déjà prêt :-). Mais nous voulons faire mieux et exposer les méthodes conventionnelles telles que

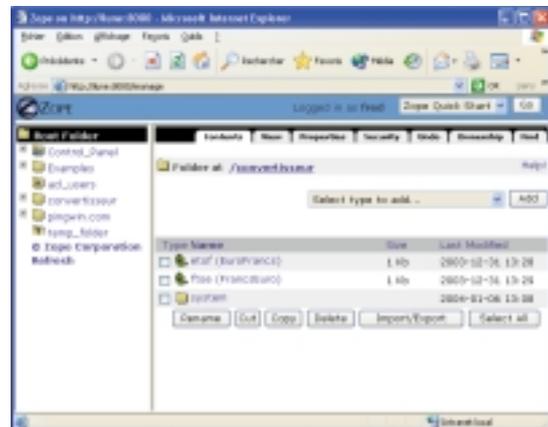


Figure 2: la racine de notre Web Service sous Zope.

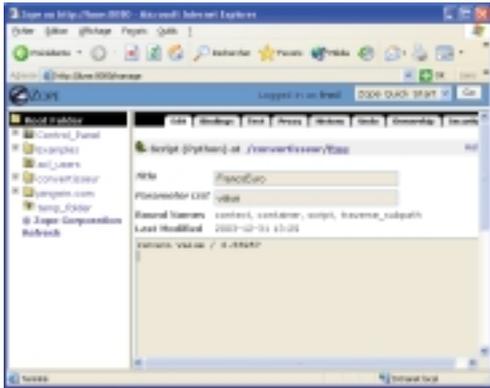


Figure 3 :
Nos scripts
reçoivent un
paramètre.

listMethod. Profitant de l'orientation objet de Zope nous avons créé un objet 'Folder' du nom de system. Dans celui-ci nous ajoutons une 'DTML Method' identifiée par listMethod et qui contient simplement ceci :

```
<dtml-var "convertisseur.objectIds(['Script (Python)'])">
```

En effet, avec Zope, un Folder ou répertoire, est aussi et même avant tout un conteneur d'objets et est bien sûr un objet lui-même. Objet qui dispose de méthodes telles que objectIds pour analyser son contenu. On appréciera que, grâce à la réflexivité de Zope, même si ultérieurement nous ajoutons d'autres fonctionnalités à notre Web service, celles-ci seront automatiquement listées, sans qu'il soit nécessaire d'intervenir en ce sens. Notons qu'au lieu d'une 'DTML Method' nous aurions tout aussi bien pu utiliser un script. Dans ce cas le code aurait été :

```
print context.convertisseur.objectIds('Script (Python)')
return printed
```

En supposant que nous soyons en phase de développement et que notre Zope écoute le port 8080 sur une machine baptisée 'lune', nous pouvons modifier notre client Python comme ceci :

```
s = Server("http://lune:8080/convertisseur")

print s.etof(1)
print s.foe(6.55957)
print s.system.listMethods()
```

Un Web Service réparti.

Ceci est fort bien. Mais puisque nous parlons Web Service, nous espérons légitimement accéder aux fonctions de notre serveur depuis une page Web. Il suffit pour cela d'ajouter la page en question dans le conteneur/répertoire. En fait, nous allons faire mieux. Nous allons simuler un Web Service réparti sur plusieurs machines. Cet agencement permet bien mieux la tolérance aux pannes et la répartition de la charge. Nous avons donc notre serveur attendant gentiment sur la machine lune qui reste en arrière plan et n'est pas connectée à Internet. Cette machine est unique pour l'exemple, mais nous pourrions en avoir de nombreuses autres chacune avec un Zope en attente. Puis vient la machine (elle se nomme soleil sur mon réseau, mais peu importe) qui va servir de frontal et qui est connectée à Internet. Cette machine reçoit les requêtes, invoque un serveur en backend via xml-rpc, récupère le résultat de l'appel et constitue une page pour présenter le résultat au navi-

gateur client. Le lecteur trouvera les archives Zope d'un backend et du frontal sur le CD-Rom. Constituons ensemble ce frontal. En premier lieu nous devons savoir, que puisque Zope supporte xml-rpc en natif, son Python intégré contient le module xmlrpclib. C'est une bonne nouvelle qui nous simplifie la vie. En revanche ce module ne peut être accédé directement depuis un script ou une DTML Method. Pour raison de sécurité bien entendu car il n'est pas question qu'un utilisateur authentifié, mais pas forcément de confiance, puisse faire n'importe quoi. Nous devons alors placer notre code dans une méthode externe. Ici, cela consiste en un fichier xmlrpcclient.py dont le contenu est :

```
import xmlrpclib

def ClientRpc(value):
    server_xmlrpc="http://lune:8080/convertisseur/"
    server=xmlrpclib.Server(server_xmlrpc)
    return server.etof(value)
```

et qui doit être déposé dans le sous répertoire 'Extensions' de l'arborescence physique de votre Zope, ce que seul l'administrateur est autorisé à faire. Créez ensuite un répertoire 'frontalzoperpc' à la racine de votre site frontal, répertoire qui contiendra trois objets. D'abord une méthode externe xmlrpcclient configurée. Remarque : il n'est pas obligatoire que l'identifiant de la méthode soit le même que le nom du script. Créez ensuite un 'DTML Document', dont le contenu est un simple formulaire.

La page d'accueil de notre Web Service.

```
<dtml-var standard_html_header>
<h2>Interrogation d'un service XML-RPC Zope depuis un frontal Zope
</h2>
<p>
<form method="POST" action="invoque_client">
  <p>Conversion Euro -&gt; Francs</p>
  <p><input type="text" name="Saisie:float" size="20">
    <input type="submit" value="Submit" name="B1">
    <input type="reset" value="Reset" name="B2">
  </p>
</form>
</p>
<dtml-var standard_html_footer>
```

Remarquez au passage l'attribut de balise :float qui assure que la méthode recevra un flottant du formulaire, ce qui est plutôt sympathique. Enfin puisque le bouton submit appelle invoque_client nous créons une 'DTML Method' sous cet identifiant et dont le contenu est :

```
<dtml-var standard_html_header>
<h2>le résultat de la conversion demandée est:</h2>
<p>
<dtml-var "xmlrpcclient(value=Saisie)"> Francs
</p>
<dtml-var standard_html_footer>
```

Remarquez combien est aisé le passage de paramètres du formulaire vers la méthode externe. Faites pointer l'URL d'un navigateur client sur l'objet frontalzoperpc (par exemple: <http://soleil:8080/frontalzoperpc>).

■ Frédéric Mazué

Application-servers.com : LA communauté Java francophone

Créée et maintenue par une poignée de développeurs, "a19s" n'en est pas moins la communauté Java francophone de référence.



Didier Girard

Programmez : Peux-tu nous présenter brièvement Application-servers.com ?

L'objectif du site est d'apporter une information fiable et d'actualité à la communauté informatique mettant quotidiennement en œuvre les technologies Java telles que UML, XML, J2EE, Struts, JSF, EJB, JDO, .Net, Open Source Java, Hibernate, Eclipse, Maven, AOP, OffShore, Gestion de projets, Tests Unitaires, etc. Pour en savoir plus allez consulter les metatags ;-)

Qu'est-ce qui t'a poussé à créer cette communauté ?

Frustré de ne trouver que de l'information disséminée, j'ai cherché à créer un site pour les développeurs d'entreprise, animé par des développeurs d'entreprise. L'idée étant qu'il devienne une base de connaissance qui reflète en permanence l'état des technologies Java et permette à chacun de progresser rapidement dans son domaine.

Que penses-tu apporter aux membres ?

Avant tout, de l'information de qualité : actualité, les technos à surveiller, etc. J'espère aussi leur faire gagner du temps. Les news que je

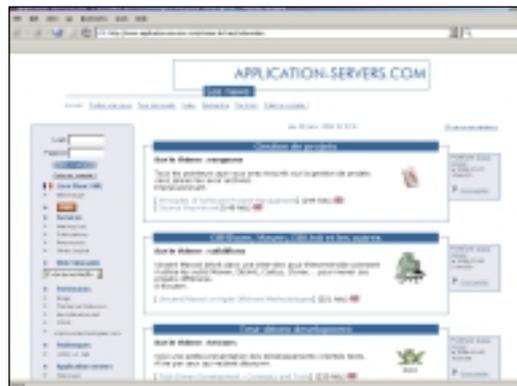
publie proviennent de plus de 200 sites que je surveille quotidiennement. Cela représente un gros travail de compilation. Je pense aussi qu'"a19s" a participé au lancement de pas mal de technologies en France : Struts, XML, AOP, Eclipse, JDO, Hibernate et plus récemment Maven. L'influence est visible sur les mailing list utilisateurs.

Quel est le point fort d'application-servers.com ?

L'indépendance totale sans laquelle "a19s" n'aurait pas la même réputation de sérieux et aussi la base de connaissances accumulées au fil du temps. J'ai une totale liberté sur le contenu et sur le ton de mon site. Je peux parler librement des technologies et des éditeurs.

Combien êtes-vous à faire vivre Application-servers.com ?

Je suis seul à publier des news mais quelques chatters m'épaulent : sebadmin (administration), gvounet (debuggage), pelpinou (orthographe), objectsavantag (supporter), brunoprovence (supporter). En fait, des milliers, en incluant les développeurs des produits open source que j'utilise pour l'infrastructure du site (Apache, Tomcat, Struts, Xerces, Xalan, Eclipse,



MySQL, Linux). J'en profite pour remercier IMPROVE, la société qui m'emploie et m'offre toute l'infrastructure d'hébergement ainsi que Noé, Léna et Laurence.

Quels sont tes projets ?

Depuis toujours, je me fixe pour objectif d'améliorer l'impact du site à travers trois idées directrices : partager avec la communauté un état de l'art toujours d'actualité, malgré un contexte mouvant, permettre à chacun de progresser dans son domaine de compétences, agir sur le niveau qualitatif global des projets mettant en œuvre de la nouvelle technologie.

■ *Propos recueillis par David Thévenon*

Contact communauté : Didier Girard, directeur technique d'Improve et fondateur d'Application-servers.com

Une aide à la décision

Je suis un lecteur assidu d'Application-Servers.com depuis son lancement, il y a près de 4 ans. Le premier intérêt du site est d'amener des éléments de décision pour l'usage d'une technologie ou d'un outil : quelle problématique il adresse, dans quel contexte il est indiqué de l'utiliser, mais aussi quelles limites il présente. Les articles sélectionnés permettent de positionner rapidement ces éléments et d'identifier les solutions les plus pertinentes. L'autre aspect d'"a19s" (pour les initiés !) qui m'apporte le plus est l'information sur des sujets plus structurants, comme l'architecture, les choix méthodologiques ou la gestion de projet qui donne une réelle ouverture sur les meilleures pratiques dans ces domaines.

Cette source d'information est extrêmement utile pour inscrire ses choix techniques et d'organisation dans une démarche cohérente. Une part importante de mon activité de veille technologique s'appuie sur Application-Servers et le site est devenu pour moi un réel outil de travail. En un mot, je pense qu'Application-Servers répond à son ambition : donner une vue d'expert sur "l'actualité des technologies web pour l'entreprise".

■ Vincent Blondel, responsable des projets informatiques de The Phone House

Fiche communauté

Adresse : www.application-servers.com

Date de création : mars 2000

Thème : Java et tous les thèmes dérivés

Cible : toute personne concernée par Java en entreprise

Niveau : Motivé

Ressources : actualité, article, Livres Blancs, extrait d'ouvrage, liste de diffusion.

Nombre de membres : 10 250

Visiteurs : 10 000 uniques par mois

Nombre de contributeurs actifs : 1

Nombre d'inscrits à la Newsletter : 6 750

Nombre d'inscrits au Flash : 250

Nombre d'inscrits au canal RSS : 100

PLAYSTATION 2 SOUS LINUX

Le kit proposé par Sony permet d'implanter Linux sur une console PS2. Un jugement rapide du processeur Emotion Engine (MIPS) 128bits à 295Mhz et ses quelques 32Mo DRam l'apparente à un gadget. En réalité, l'ensemble se révèle particulièrement étonnant.



Le kit Linux pour PlayStation 2 comprend des périphériques et une version de linux dédiée à la console (de jeux). Le système d'exploitation se présente sous la forme de deux DVD respectivement dédiés à l'environnement d'exécution (Runtime) et aux codes sources (Software Packages). Les équipements fournis autorisent la transformation de la console en un véritable micro-ordinateur : un adaptateur Ethernet (10/100 Mb), un disque dur (40Go), un clavier USB, une souris USB, un câble de raccordement à un moniteur et aux haut-parleurs. En complément, il convient de se doter d'une console PS2, d'une carte mémoire 8 Mo pour PS2, d'un câble Ethernet (RJ45) et d'un écran VGA. L'écran doit impérativement supporter la synchronisation sur la composante verte, notée SOG (Sync On Green). Malgré l'imposante liste proposée sur le site Web de Sony (<http://playstation2-linux.com/sog.php>), le choix d'un moniteur s'avère assez difficile, car les modèles référencés sont anciens ou peu distribués. La solution consiste à se déplacer avec l'ensemble dans une surface commerciale spécialisée. Une brève explication avec le chef de rayon et la prise d'un rendez-vous (en période creuse) permettent de procéder aux essais et d'opter pour l'écran adéquat : fonctionnel et abordable. L'ensemble des modèles

Dillo (www.programmez.com), XMMS (Les Visiteurs), GMIX, AbiWord (Texte), navigation dans un menu FluxBox et plaympeg (perroquet)

doit être évalué. Les documentations précisent rarement cette spécificité (SOG) et si les plus onéreux se montrent presque toujours compatibles, quelques modèles de base proposent également cette fonction.

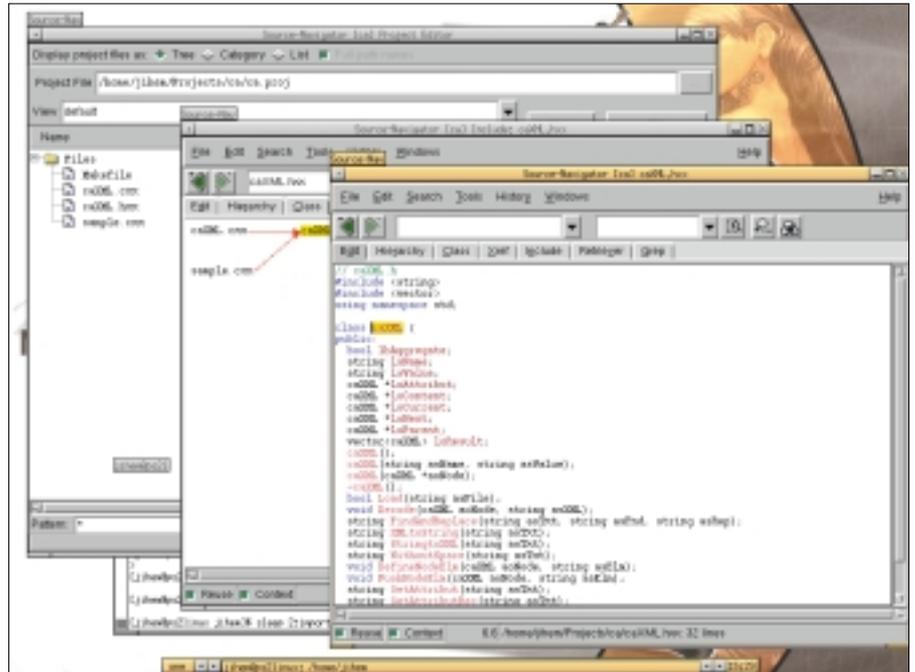
Préalablement à la mise en service du kit, la configuration du système doit être établie par l'intermédiaire d'un téléviseur et du câble délivré avec la console (et non celui du kit) : langage, zone horaire, heure d'été, etc. L'installation débute – hors tension – par le retrait du cache arrière de la console. Elle se poursuit par le raccordement du disque dur IDE à l'interface Ethernet et à l'insertion de l'ensemble dans l'emplacement dédié à cet effet, devenu apparent suite au retrait du cache. L'adaptateur présente une excroissance dotée de deux vis le solidarissant de la console, une fois en place. L'ensemble clavier – souris se branche directement sur les prises USB présentes en face avant, voire en cascade. Le clavier dispose à cet effet d'une prise USB femelle. Le câble de raccordement au moniteur VGA se connecte à la place de celui exploité pour le téléviseur. Il dispose de deux fiches audio, de type cinch femelle, respectivement

blanche (canal gauche) et rouge (canal droit). En l'absence de connecteur audio sur le moniteur, l'emploi d'une paire d'enceintes dédiées au PC requiert l'usage ou la réalisation d'un adaptateur doté de deux prises de type cinch mâle d'un côté et d'une prise jack 3,5mm femelle de l'autre. L'introduction de la carte mémoire dans l'emplacement "memory card 1" termine la préparation matérielle.

L'installation du système d'exploitation requiert l'introduction du premier DVD (Runtime) dans le lecteur, immédiatement après la mise sous tension. Le moniteur signale une absence de signal vidéo jusqu'à la prise en charge du support. Le menu de démarrage de l'environnement : "install", "boot", "rescue" apparaît finalement. La sélection de la première option ("install") active la procédure d'installation. Le disque dur et la carte mémoire s'avèrent conditionnés à l'usage exclusif de Linux et ne peuvent être réemployés à d'autres fins. Les écrans d'installation ressemblent à ceux des distributions RedHat (avec noyau 2.2) en mode texte. La sélection du langage se limite à une confirmation : seule

la langue anglaise est supportée (english). L'indication "us" correspond au type du clavier à gérer (en l'occurrence qwerty). Les indications de changements de disque doivent faire l'objet d'une attention particulière. A défaut, toute la procédure doit être recommencée. La sélection des packages s'opère selon le type d'installation retenue : station de travail (Workstation), serveur (Server system) ou personnalisé (custom system). La dernière option autorise, au choix, une sélection plus précise, ou l'installation de l'intégralité de la distribution. La partition du disque dur s'effectue à l'aide des utilitaires Disk Druid et / ou fdisk. Elle requiert une partition initiale (hda1) de 2048Mo dont la taille croît jusqu'à remplir le disque (cocher "grow to fill HDD") rattaché à la racine "/" et une zone swap de 128Mo (hda5).

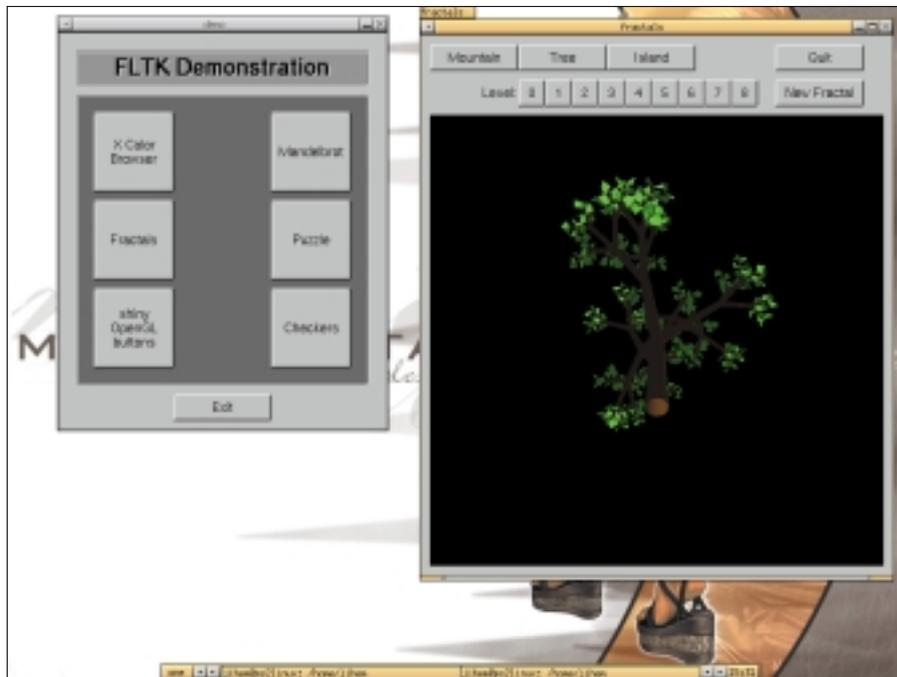
Les paramètres réseau nécessitent l'adoption d'un nom pour le système (ex. ps2linux) et la détermination des règles d'adressage (ex. BOOTP/DHCP). Le fuseau horaire "Europe/Paris" précise les modalités relatives à la gestion de la date et de l'heure. La saisie du mot de passe de l'administrateur (root), la création d'un ou plusieurs comptes utilisateur et l'indi-



Red Hat Source Navigator 5.1

vation des partitions nécessitent quelques instants, etc. Au terme de l'installation, le système redémarre (restarting system), il convient alors d'opter pour l'option "boot" du menu de

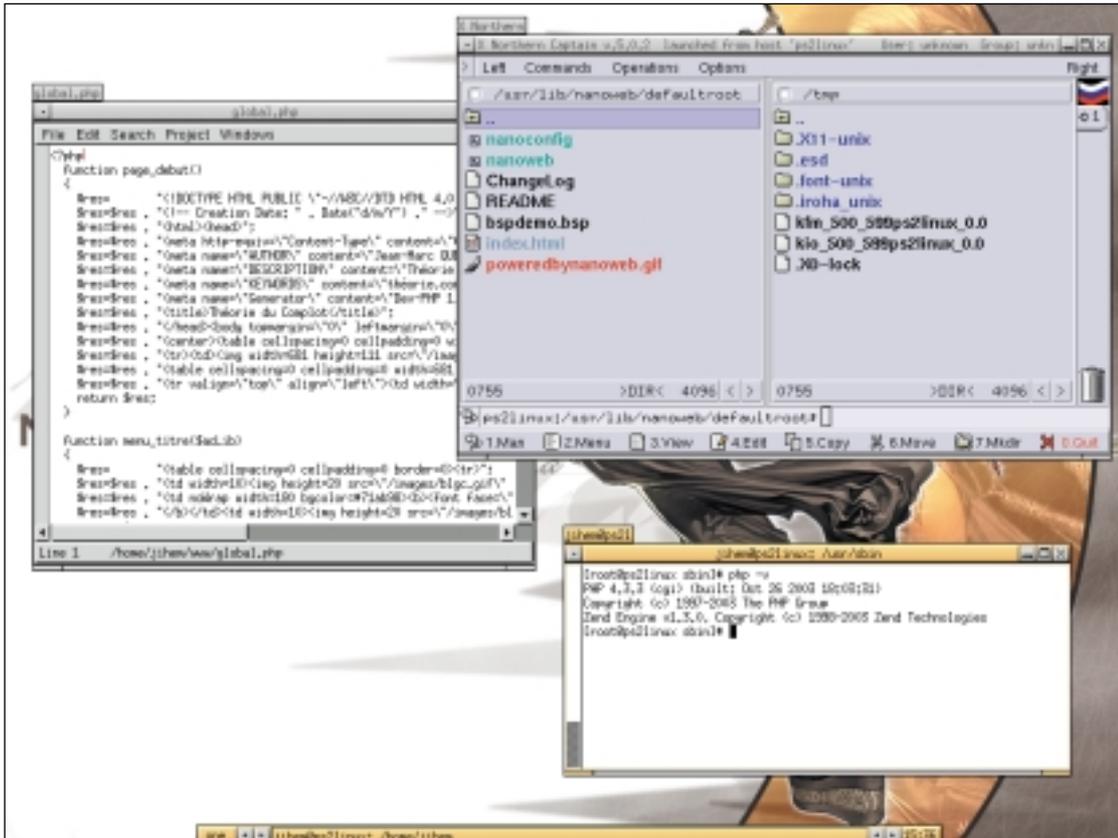
teur (root) avec le mot de passe préalablement indiqué permet d'activer l'utilitaire "sdr" pour sélectionner l'interface graphique utilisée : Fvwm2, Gnome, KDE, WindowMaker, etc. KDE (1) s'avère relativement pratique (pour les habitués de Windows). Les initiés (à Linux) peuvent adopter leur gestionnaire préféré (UserDefined). L'adoption de FluxBox (<http://fluxbox.sourceforge.net/>) permet ainsi d'économiser les ressources, tout en bénéficiant d'une interface des plus efficaces. Le téléchargement des sources de la dernière distribution stable (0.1.14) autorise la constitution de binaire compatible (mips) à partir d'une session terminal à l'aide des trois instructions ci-après : ". /configure --build=mips", "make", "make install". L'usage de l'option "-build", lors de l'appel du gestionnaire de configuration, explicite la nature de l'environnement qui peut ne pas être automatiquement détecté (car inhabituel). La création du fichier ".Xclients", doté de la ligne "exe /usr/local/bin/fluxbox", dans le répertoire associé au compte exploité, en permet l'usage, suite à l'appel de "startx". FluxBox offre différents styles graphiques personnalisables dont "LemonSpace" (voir illustrations). L'ensemble de la configuration (menu, touches spéciales, ...) s'effectue au sein de fichiers texte (dans ~/.fluxbox). Elle ne requiert qu'un éditeur texte et la configuration proposée par défaut en facilite la compréhension. Le menu comporte des balises hiérar-



FLTK démonstration de fractals (arbre)

cation de la nature des mots de passe (Shadow, MD5) sécurisent le système. Les opérations suivantes se limitent à des confirmations : choix du serveur X (gs), démarrage à partir de la carte mémoire, formatage et acti-

démarrage, puis d'indiquer "memory card" pour désigner l'emplacement de l'image à amorcer. L'invite ne tarde pas à s'afficher : "PS2 Linux release 1.0, Kernel 2.2.1 on a mips, ps2linux login :". La connexion en administra-



zoinks (éditeur),
nanoweb (serveur Web
sous php), php et xnc
(gestionnaire
de fichiers)

chiques (entre crochets) et des libellés à afficher (entre parenthèses). La commande à exécuter, associée à une balise d'exécution (exec) s'exprime entre accolades. Le principe, simple, se montre d'une efficacité redoutable. La modification du fichier des paramètres du serveur X (/usr/X11R6/lib/X11/XGSCConfig) offre une résolution de 1024x768 en 24bits (Subsection "Display", Depth 24, Modes "1024x768" "800x600" "640x480", EndSubsection) exploitant au mieux le moniteur VGA raccordé. Dans le même esprit, l'usage de la section "Pointer" ci-après autorise l'usage de la molette de la souris : Section "Pointer", Protocol "IMPS/2", Device "/dev/mouse", ZAxisMapping 4 5, EndSection.

```
~/fluxbox/menu
[begin] (Fluxbox-0.1.14)
[exec] (Terminal) {xterm}
[submenu] (Programs)
[exec] (Editor) {zoinks}
[exec] (File Manager) {xnc}
...
[end]
[end]
```

Le kit comporte la solution logicielle de partage de volumes disque et d'imprimantes Samba (<http://www.samba.org>). Elle facilite la mise en

réseau de la station PS2 Linux avec d'autres ordinateurs, notamment des PC sous Windows. Le système ainsi composé permet d'exploiter les périphériques de ces derniers (imprimantes, graveurs, ...) à partir de la console, en particulier pour effectuer des sauvegardes. L'obtention d'un nouveau logiciel se limite – en général - au téléchargement des sources, à leur compilation et à l'implantation des binaires résultants. Le projet CFYC (compiled for your convenience) propose une alternative sous la forme d'une collection de binaires prêts à l'emploi. La consultation des sites Web associés (<http://playstation2-linux.com/projects/cfyc> et <http://cfyc.hikey.org>) évite souvent de longues compilations. L'utilisateur peut ainsi bénéficier des logiciels phares de la plate-forme (Linux) : AbiWord (traitement de textes), Dillo (navigateur Web), Gimp (dessin), PlayMPEG (lecteur vidéo), XMMS (lecteur audio), etc.

Le développeur peut également s'intéresser aux applications suivantes : zoinks (éditeur, <http://zoinks.mikelockwood.com>), fltk (toolkit graphique, <http://www.ftk.org>), php (langage, <http://www.php.net>), lua (langage, <http://www.lua.org>), source navigator (environnement de développement intégré, <http://sourcnav.sourceforge.net>) et xnc (gestionnaire de fichier, <http://xnc.dubna.su>).

Elles se compilent assez aisément et se montrent particulièrement performantes. Le serveur Web, écrit en php, nanoweb (<http://nanoweb.si.kz>) autorise l'implémentation d'un serveur Web de pages dynamiques (php) sur la station. Il requiert une version de php (4) dotée des sockets et de pcntl. La compilation de php doit donc préalablement être effectuée à l'aide des instructions suivantes : ". /buildconf", ". /configure --enable-sockets --enable-pcntl", "make", "make install". Le serveur peut ensuite être activé et stoppé par : "nanocli start" et "nanocli stop".

En conclusion, la solution PS2 Linux associée à un accès ADSL (ou en réseau avec un PC) constitue une plate-forme idéale pour déployer Linux. La console étonne par ses performances, en particulier sa capacité à démarrer des services (serveur de fichiers, de pages Web, ...) et une interface graphique avec seulement 32Mo de mémoire. Elle donne toute sa dimension en matière de développement sous Linux. Le SDK de Sony permet de bénéficier pleinement de l'architecture Playstation 2 : moteur 3D, gestion du son, unités de calcul spécifiques, etc. Son usage requiert toutefois une approche préalable de Linux.

■ Jean-Marc QUERE