

Programmez!

Mensuel - Novembre 2005 - N°80 • 5,95 €

TESTS: DÉJOUER LES BUGS



64-bit: SOYEZ PRÊTS

L'ACTU DU DÉVELOPPEMENT JEUX

EDITIONS ATLAS
Route des Dragons
1103 CHEVAUX/LAUSANNE

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

M 04319 - 80 - F: 5,95 €



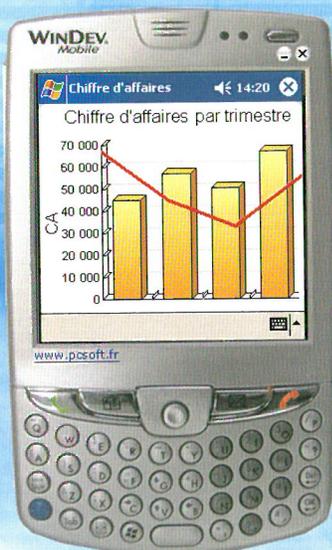
.NET C# 3 : une évolution majeure - **LINQ** : un langage de requêtes détonnant **PDC 2005** Les annonces Microsoft pour 2006 et 2007
Java Manipuler Java Imaging **Open Source** Des Web Services faciles avec XFire **Windows** Interopérabilité COM et .Net

DÉVELOPPEZ 10 FOIS PLUS VITE

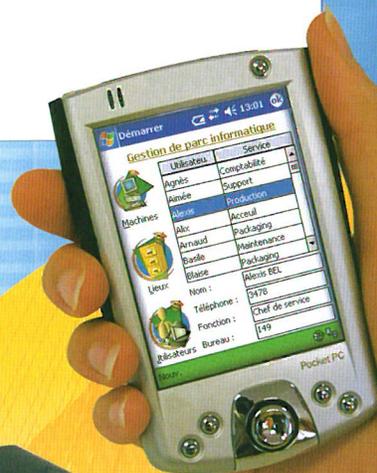
DÉVELOPPEZ DE PUISSANTES APPLICATIONS SUR MOBILE

«On peut commencer à
développer sur Pocket PC
en 10 minutes»

Magazine Réseaux et Telecom



Un graphique dans une
application de gestion.

POCKET PC**TERMINAL**

WINDEV Mobile est certainement le meilleur environ-
nement professionnel pour développer des applications
destinées aux terminaux mobiles !

En français, doté d'un langage de 5^e génération,
WINDEV Mobile permet de développer 10 fois plus vite
tous les types d'applications, autonomes, répliquées ou
reliées en temps réel aux données de l'entreprise.

Compatible avec **WINDEV** et **WEBDEV**.
Déploiement libre, Hot Line incluse.

Demandez le dossier technique gratuit, accompagné de
100 témoignages et d'un CD

SMARTPHONE

(Windows Mobile)



Sommaire

EDITIONS ATLAS

Route des Dragons 7

1033 CHESEAUX/LAUSANNE

n°80

/ACTUS

L'actualité en bref.....	6
Agenda.....	19

/ÉVÈNEMENT

Professional Developer Conference à Los Angeles.....	12
--	----

/TESTS

CodeSmith : génération automatique ; CodeShare : partager les codes.....	16
--	----

/CAS ENTREPRISE

Le Monde se met à XUL.....	18
----------------------------	----

/SGBD

.....	20
-------	----

/INTÉGRATION

.....	22
-------	----

/DOSSIER : LE DÉFI DE VOTRE CODE

1 » Méthodes

Les tests et rien d'autre.....	25
Maîtriser les tests unitaires.....	29
Ne pas sous-estimer les tests de non régression.....	33
Un robot pour améliorer les tests.....	34
L'art subtil du test.....	38
L'art de détecter les bugs introuvables.....	39
France Telecom : une cellule de tests mutualisés.....	41
Tests automatiques-tests manuels : choisir une méthode adaptée.....	42
Nuxeo pratique le test en Open Source.....	44

2 » Outils

Les outils commerciaux.....	46
Les outils de tests Open Source.....	49
Les tests dans Visual Studio 2005.....	50
Testeur, un métier en pleine revalorisation.....	51

/FOCUS : 64-BIT SOYEZ PRÊTS POUR LE FUTUR

Le marché du 64-bit.....	52
Comment bien tirer parti des processeurs double cœur.....	54
.Net, Java et le 64-bit.....	55

/TECHNOLOGIE

Interopérabilité .Net-PHP (2 ^e partie).....	58
--	----

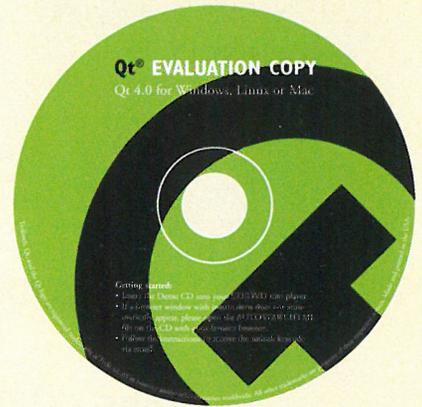
/PRATIQUE

XFire : développer des services web.....	61
C#3.0 : attention, chaud devant.....	64
LINQ : le futur framework .Net de requêtage des données.....	67
Manipuler vos images avec Java.....	70
Mettre en œuvre l'interopérabilité COM/Net.....	73
Développement d'applications avec Visual Studio 2005 Tools for Office (2 ^e partie).....	77

/TEMPS LIBRE

Ludique.....	80
Livres.....	82

CD-ROM PROGRAMMEZ N°80



Trolltech

La librairie Qt est devenue en quelques années une référence de la programmation multi-plate-forme C++ que ce soit pour l'interface ou la conception d'applications critiques. Découvrez la dernière version de Qt 4 pour en apprécier la puissance (plus de 400 classes) et les multiples possibilités. Grâce à ce CD-Rom (version d'évaluation 30 jours) vous aurez tout le nécessaire pour tester réellement Qt, les outils de développement, en créant vos applications ou en testant les exemples fournis.



Macromedia

Dreamweaver 8

Le plus populaire des environnements de développements Web revient dans une version largement améliorée : éditeur visuel CSS, prise en charge des flux RSS, meilleure intégration des données XML. On bénéficie d'un éditeur de code plus ergonomique, plus pratique. L'utilisation de vidéo se résume à quelques clics de souris !

Contribute 3

Désormais intégrée à la suite Macromedia Studio, Contribute 3 simplifie la publication de contenus et de

Suite page 5 ►



International Conference Center:
2005 CONFERENCE ON
Visual Studio .NET
Development for Windows
AND MAC, LINUX ETC.

**PRÉSENTATION DE Qt 4
VÉRITABLE ENVIRONNEMENT
INTER-PLATEFORMES POUR MICROSOFT®
VISUAL STUDIO® .NET**

Qt, l'environnement par excellence de développement C++ inter-plateformes, est désormais intégré en toute transparence à Visual Studio .NET. Les développeurs Windows peuvent donc désormais cibler en mode natif les plateformes Windows®, Mac® OS X et Linux à partir d'un code source unique.

Pour cibler un nouveau système d'exploitation, vous n'avez guère plus à faire qu'une simple recompilation. Vous pouvez ainsi étendre facilement vos compétences Visual Studio .NET sur de multiples plateformes. Faites passer le mot!

Enregistrez votre demande d'évaluation sur
le CD dès aujourd'hui ou directement sur
www.trolltech.com/windows.

Qt® BY TROLLTECH
CODE LESS. CREATE MORE.



Optimized for

Microsoft
Visual Studio.net

Trolltech, Qt, and the Qt logo are registered trademarks of Trolltech AS in Norway and/or other countries worldwide. All other trademarks are property of their respective owners.

► Suite de la page 3

pages web sur un site. Même sans connaître HTML et la programmation, en quelques minutes, on crée et on transfère des pages web.

Borland

Delphi 2005 Personal Edition

L'un des outils Borland les plus populaires, Delphi, permet de développer rapidement des applications en langage Delphi et .Net (C#). Vous pouvez facilement créer des logiciels Windows (en utilisant Win32) et pour le framework .Net. Cette édition personnelle intègre toutes les fonctions de base pour bien démarrer. On dispose de tous les langages, d'outils visuels pour l'interface et pour le mapping de données. Il intègre aussi de puissants débogueurs et outils de tests.

Lyria

Leonardi 3.1 Free Edition

Pour créer rapidement des interfaces d'applications complexes, Lyria offre la nouvelle version de son framework Leonardi. Entièrement écrit en Java et XML, Leonardi s'utilise facilement avec Eclipse. Les interfaces créées s'utilisent ensuite sur des clients légers et lourds (Swing, SWT...) et quel que soit le système.

PHP

WampServer 5

Vous êtes sous Windows et vous voulez installer et utiliser MySQL, Apache et PHP 5 ? Rien de plus simple, WampServer fait tout ou presque à votre place, grâce à un installateur graphique.

Maguma OpenStudio

Pour profiter tout de suite de PHP sous Windows, nous vous proposons l'un des IDE PHP les plus connus et les plus simples : Maguma OpenStudio. Il possède toutes les fonctions indispensables : éditeur personnalisable, bibliothèque de code, module FTP, gestionnaire de projet, etc.

Sauvons le développeur !

Nous l'écrivons souvent dans Programmez!, le développeur doit concilier des exigences contradictoires.

MDA, SOA, XML, services web, d'interface à langage descriptif, cycle de vie, modélisation, etc. rentrent désormais dans les mœurs et les bonnes pratiques. Ils obligent à une mutation profonde de l'organisation du travail et de la méthode de développement. On unifie enfin le codage, mais on oblige à apprendre toujours plus de nouveaux outils, de nouvelles fonctions. Il faut faire cohabiter telles applications, avec telle base de données, puis les envoyer sur un site web, via des services Web en environnement SOA. Implémenter de nouvelles couches logicielles, de nouveaux standards...

Effectivement, l'utilisateur bénéficie de plus de fonctionnalités, l'entreprise devient plus flexible, plus réactive à son environnement. Mais cela alourdit le travail du développeur, dont on exige toujours plus de compétence et d'intelligence.

À force de le mettre en cage, de le brider avec une foule de processus, d'outils en tout genre, n'est-on pas en train de faire disparaître son rôle ? Et comment cette espèce en voie de disparition pourrait-elle longtemps garder sa motivation ?

A trop dévaloriser son rôle, doit-on s'étonner que de nombreux étudiants en informatique se détournent du métier ? L'architecte, le chef de projet sont importants, mais n'oublions pas le développeur. Il représente un maillon que l'on aurait tort de sacrifier. Sauvons le développeur. Relâchons-le dans la nature du code et des langages. Remettons-le aux pizzas, aux quiches, au Coca (allégé bien entendu) !

■ François Tonic



QUOTIDIEN

www.programmez.com

Thumbnail of the Programmez.com website interface showing various news articles and navigation menus.

A visiter absolument! Actus, Forums, Téléchargement

LE MAGAZINE DU DEVELOPPEMENT | Programmez!

Rédaction : redaction@programmez.com

Directeur de la Rédaction : Jean Kaminsky.

Rédacteur en Chef : François Tonic.

Ont collaboré : A. Agahi, P. Couzy, A. Crepet, N. Dasriaux, X. Leclercq, F. Mazué, Y.A. Oudghiri, P. Pameux, A. Kolawa, A. Kerhuel, G. Renard, J. de Schryver, M. Varandat.

Maquette : PLB Communication.

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

Editeur : Go-02 sarl, 6 rue Bezout - 75014 Paris
Dépôt légal : 4° trimestre 2005 -
Commission paritaire : 0707K78366 - ISSN : 1627-0908
Imprimeur : ETC - 76198 Yvetot

Directeur de la publication : Jean-Claude Vaudecrane.

Ce numéro comporte deux CD Rom et un coupon d'abonnement encarté après la couverture.

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

PROCHAIN NUMÉRO n°81 - Décembre 2005 - Parution le 30 novembre

DOSSIER

.Net 2.0

Avec la sortie du nouveau framework .Net, découvrez Visual Studio 2005, Team System, la gamme Express, SQL Server 2005 !

FOCUS

Base de données

Au cœur des applications, la donnée devient de plus en plus complexe. My SQL peut-il s'imposer ? Comment le développeur doit-il implémenter la donnée ? Quel SGBD choisir ?

MOBILE

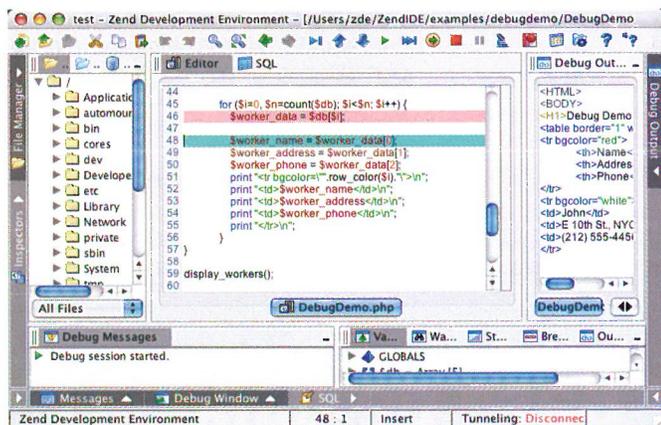
Palm perd son pilote

Une page s'est définitivement tournée dans le monde des PDA. Après avoir dominé ce marché et en grande partie raté le virage du PDA communicant, Palm a décidé d'utiliser Windows Mobile 5.0 pour son futur smartphone Treo (sortie prévue début 2006). Cette annonce prouve que Microsoft commence réellement à s'implanter sur le marché des téléphones « intelligents » après avoir su prendre une bonne position sur les PDA classiques, même si le succès du BlackBerry continue à fortement concurrencer Windows Mobile. Trois semaines avant l'annonce Windows Mobile, Palm Source (s'occupant du système) se faisait racher non pas par Palm mais par Access, un éditeur japonais. À l'heure actuelle un certain flou entoure l'avenir de PalmOS et de son évolution. Palm a tout de même annoncé qu'il envisageait de poursuivre la collaboration avec Palm Source via Access, mais sur quelle version du système : PalmOS 5.x ou 6.x ?

PLATE-FORME

Zend étend sa compatibilité

Après un partenariat avec IBM qui a suscité beaucoup de rumeurs, c'est au tour d'Oracle. Zend Technologies a annoncé fin septembre la disponibilité d'une préversion de Zend Core for Oracle. La version finale de l'environnement PHP est attendue fin 2005. « Oracle nous a approché. Mais nous travaillons avec l'éditeur depuis fin 2004, sur le Studio. » indique Michel Gerin (Zend Technologies). Comme les utilisateurs d'Oracle utilisent aussi PHP, il manquait une plate-forme technique pour faire cohabiter les deux environnements sans heurts. Oracle constate aussi que la concurrence n'est pas inerte. MySQL fonctionne très bien avec PHP, DB2 est maintenant pris en charge. Oracle ne pouvait pas rester en dehors de ce mouvement s'il souhaite étendre son marché. Et Microsoft ? « Microsoft se rend compte du succès de PHP. Il y a quelques mois, ils sont venus nous voir pour que l'on teste IIS avec PHP. » poursuit Michel Gerin. Si, officiellement, aucun projet d'un Zend Core for Microsoft n'est en préparation, ce serait une suite logique et une ouverture pour SQL Server. PHP sort peu à peu de son côté purement frontal. Il descend de plus en plus sur le serveur et le Back Office. « Le but de Zend est de simplifier le développement et la production avec PHP, via des exemples, des



assistants, tout en assurant les performances de l'application » précise Michel Gerin. L'éditeur a sorti la version 2 de son Zend Platform et la bêta de Zend Studio 5. Côté plate-forme de développement, Windows et Linux font à peu près jeu égal, même si Windows demeure très légèrement en tête. Étonnement, MacOS X est utilisé entre 15 et 20 % ! Sur le calendrier et les futures versions, Zend demeure très discret. Tout juste sait-on qu'il y aura de nombreuses annonces produits dans les mois à venir, et qu'il faudra s'attendre à des annonces Zend - IBM.

JAVA

Alliance Google-Sun



Les deux sociétés ont annoncé début octobre un partenariat centré sur les logiciels. Sun proposera dans l'environnement runtime Java (JRE) la barre Google. Pour sa part, Google devra promouvoir les logiciels et technologies Sun. Deux environnements sont mis en avant dans cette annonce : Java (essentiellement la JRE) et OpenOffice. Il s'agit pour Sun de mieux diffuser ses logiciels et technologies (Java, OpenOffice, OpenSolaris) sous-entendu, face aux outils Microsoft (.net et MS Office). Pour les développeurs, le fait de disposer de la Google Toolbar facilitera l'intégration de cette barre de recherche Google. Sun espère sans doute conforter Java face à la plate-forme .Net qui monte en puissance. Pour Google, il s'agit de continuer à étoffer les services et les applications autour de son moteur de recherche. Sur OpenOffice, les deux sociétés ont annoncé qu'elles travailleront ensemble pour y apporter de nouvelles fonctions (quid de StarOffice ?). On s'attend à une offensive de Google sur le poste de travail.

TECHNOLOGIES

JBoss interopérable avec Microsoft

L'éditeur libre JBoss et Microsoft ont annoncé une interopérabilité entre le système middleware d'entreprise JBoss et les outils Windows Server. L'objectif est de mieux intégrer les outils JBoss sur la plate-forme serveur Windows en assurant une interopérabilité maximale. À terme, cela doit faciliter l'intégration et le déploiement de solutions JBoss sous Windows. Cette annonce concernera l'annuaire Active Directory, les services web (interopérabilité assurée par WS-*) et SQL Server (optimisation des performances pour ceux qui utilisent JBoss Hibernate).

Lotus

IBM®



IBM LOTUS NOTES & DOMINO PRESENTE

VOUS FACE A LA MESSAGERIE D'HIER ET AUX EXIGENCES DE DEMAIN

LA COLLABORATION STRATEGIQUE DEFIE LA PRODUCTIVITE AFFAIBLIE

A L'AFFICHE : LE NOUVEAU LOTUS NOTES & DOMINO 7
PLUS FORT QU'UNE MESSAGERIE, PLUS FORT QU'UN AGENDA, PRET A RELEVER TOUS LES DEFIS

AVEC : UNE SECURITE POIDS LOURD,
UNE GESTION POIDS PLUME,
UNE INNOVATION PERMANENTE
AU SERVICE DE LA COLLABORATION

DES OUTILS DE DEVELOPPEMENT APPLICATIF
QUI ONT DU PUNCH

LOTUS NOTES & DOMINO 7 - MEMBRE DE LA FAMILLE IBM WORKPLACE.

★ LE MIDDLEWARE* IBM. PUISSANCE ET FIABILITE. ★
RIPOSTEZ AVEC IBM.COM/MIDDLEWARE/COLLABORATE/FR OUVERT A TOUS LES CANDIDATS

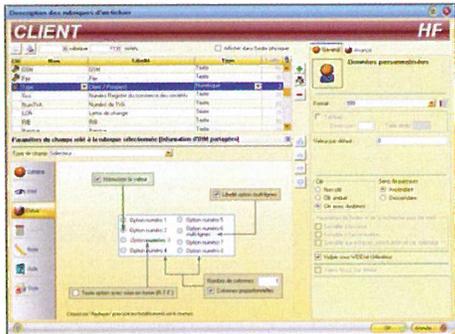
*Logiciels d'infrastructure. IBM, le logo IBM, Workplace, Lotus, Lotus Notes et Domino sont des marques d'International Business Machines Corporation aux États-Unis, dans d'autres pays ou les deux. © 2005 IBM Corporation. Tous droits réservés.

Lotus

OUTILS

WinDev puissance 10

L'éditeur PC Soft annonce une évolution de toute sa gamme (WinDev, WebDev, WinDev Mobile) en version 10. Avec 151 nouveautés Sur WinDev. L'éditeur a mis le focus sur l'assistance au développement en équipe et à son automatisation, le suivi de projets, l'optimisation des performances. On dispose désormais d'un outil d'états et de requêtes



L'interface de saisie des rubriques privilégie le partage des informations

gratuit, livrable avec son application. Sur la gestion des sources, on bénéficie de l'outil GDS (gestion des sources) qui offre toutes les fonctions d'un versioning. Pour la bonne gestion du projet, WinDev inclut un contrôleur de planning et de suivi. Sur les données, l'environnement implémente une réplique universelle

(ex. : répliquer une base mobile avec un serveur utilisant Oracle). Sur le langage interne, 250 nouvelles fonctions sont annoncées. Toutes les nouveautés fonctionnelles ne concernent pas les trois outils (WinDev, WebDev, WinDev Mobile). Notons qu'il est maintenant possible d'importer des projets et éléments WebDev dans WinDev. De plus, pour simplifier la création des écrans et éviter la génération de code ou d'en écrire, on dispose maintenant du RID (rapid graphical Interface Design). Sur le langage, on notera une meilleure gestion des ports USB (liste des matériels USB connectés, détection de connexion d'un matériel USB). Le passage de la v9 à la v10 se fait par simple compilation, indique l'éditeur. Site : <http://pcsoft.fr>

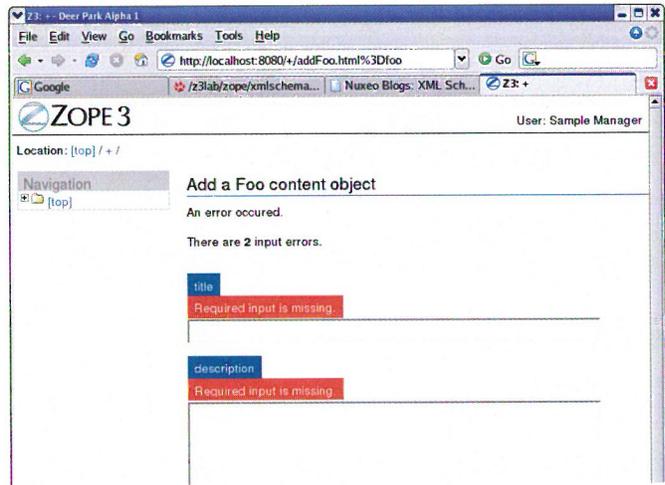
TECHNOLOGIES

Macromedia travaille sur Flex 2 et Flash Player 8.5

L'éditeur travaille déjà sur Flash Player 8.5, à peine la v8 disponible. Elle doit apporter de meilleures performances aux applications et animations. Elle inclura une nouvelle machine virtuelle, une gestion des erreurs au niveau runtime et surtout ActionScript 3.0 qui prendra en compte les dernières spécifications du ECMAScript. Sur Flex, Macromedia a annoncé la gamme Flex 2. On disposera du Flex Framework 2 (modèle de développement et librairie de composants) et surtout du Flex Builder 2, basé sur Eclipse (nom de code : Zorn). On pourra utiliser les nouvelles fonctions de Flash 8. Flex Enterprise Services 2 offrira des fonctions de tests, les services de données et toute l'architecture nécessaire aux applications Flex. Enfin, Flex Charting Components 2 fournira de nouveaux éléments pour visualiser les données. Flex 2 facilitera l'interopérabilité avec XML et les services web. Depuis la mi-octobre, les versions alpha sont en principe disponibles, la sortie des versions finales est envisagée dans la première moitié de 2006.

GESTION

Zope se simplifie



Le projet Zope est désormais en version 3.1.0. La v3 de Zope apportait une orientation très développeur de l'environnement en rompant assez radicalement avec la v2.x. Cette évolution a eu pour conséquence de ne pas provoquer une migration massive vers la nouvelle version 3.0. La v3.1.0 fournit un nouveau module d'authentification (similaire à Zope 2 PAS), l'architecture composant a été revue et corrigée pour le simplifier (plus de concept de services, gestion plus simple, support des sous-menus), ajout du framework index, amélioration de l'API doctool. Le projet Five continue, en parallèle, à évoluer. Il a pour objectif de porter les fonctions de Zope 3 dans Zope 2.x.

Site : <http://www.zope.org>
 Projet Five : <http://codespeak.net/z3/five/>

ACCÈS

IBM aime aussi les seniors

Avec le vieillissement de la population et les seniors de plus en plus nombreux à utiliser les nouvelles technologies, le développeur doit en tenir compte dans les applications et les sites web. Pour aider le développeur et les entreprises, IBM propose sur son site Alphaworks, une suite d'outils et de technologies dédiées : un optimiseur de clavier permettant d'ajuster son clavier à sa manière de taper (ex. : une ou deux mains), adaptation du Web (adopter les pages web selon les handicaps), lecture de souris (non répercution de tremblements excessifs de l'utilisateur sur la souris), et Reflexive User Interface Builder (pour construire des applications répondant aux besoins des seniors). Ce dernier élément est destiné aux développeurs, afin de faciliter le développement d'interfaces adaptées aux utilisateurs âgés. Il se présente sous forme de plug-in Eclipse. On conçoit des interfaces Swing ou SWT, en se basant sur un fichier de description XML. Il permet de définir les propriétés de validation et les liens entre les composants. Malheureusement, cette technologie s'utilise uniquement sous Windows. Son utilisation nécessite l'API de Reflection et une bonne dose de programmation aspect...

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

Offrez ce qu'il y a de mieux à vos projets de développement

Bénéficiez d'un aperçu de XMLSpy® 2006 et comprenez pourquoi il est essentiel de maîtriser XML.

Nouveautés dans la version 2006:

- Support schema-aware XSLT 2.0
- Support schema-aware XPath 2.0
- Intégration Microsoft® Visual Studio® .NET 2005
- Intégration Eclipse 3.1

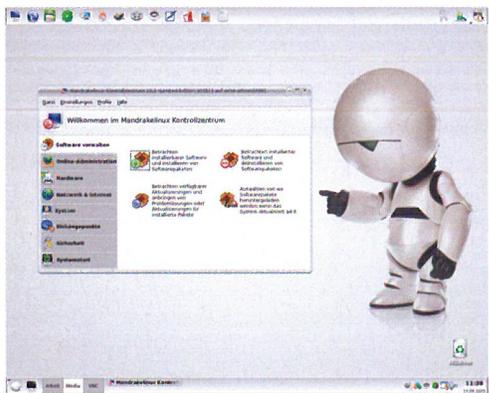
Altova® XMLSpy, l'outil standard de développement XML, est la référence en matière de modélisation, d'édition, de débogage et de transformation de toutes les technologies XML. Sa conformité aux normes, ses outils de gestion innovants et son intégration de plate-formes avancées vous permettront d'optimiser votre stratégie. Avec XMLSpy 2006, vous pouvez structurer des schémas XML, créer des documents XML puis générer automatiquement un code d'exécution dans plusieurs langages informatiques.

Vous aussi, devenez un génie de la programmation!

Téléchargez XMLSpy® 2006 dès aujourd'hui: www.altova.com

SYSTÈME

Deux nouvelles distributions GNU Linux



- **Mandriva Linux 2006** : cette nouvelle version incorpore les dernières évolutions des outils et bibliothèques. Il supporte nativement le processeur Centrino et inclut en standard Skype ! L'éditeur français intègre pleinement les outils et technologies rachetés depuis quelques mois (Conectiva et Lycoris). Mandriva 2006

intègre aussi un nouveau moteur de recherche, Kat. Distribution disponible sur 32 et 64-bit. Site : <http://www.mandriva.com>

- **SUSE 10.0 / OSS 10.0** : Novell rend disponible les nouvelles éditions de SUSE. Cette version incorpore OpenOffice 2, le moteur de recherche Beagle et une préversion de l'outil de virtualisation, Xen. Pour le développement, on bénéficie de Mono, Eclipse ainsi que MySQL et Deby. Site : <http://www.novell.com>

IDE

2006 s'annonce chez Borland

L'éditeur continue à dévoiler les nouvelles versions de ses outils, en annonçant Delphi 2006, C++ Builder 2006, et surprise, C# Builder 2006 ! Surprise, car cet IDE pur .Net n'avait plus connu d'évolution depuis presque 2 ans et de son intégration dans Delphi .Net. Cette fois-ci, son orientation est très clairement entreprise en incluant le framework ECO III et une intégration poussée avec la modélisation de Together. Le fonctionnement est bidirectionnel pour maintenir l'intégration du modèle et du code en cas de modification. Delphi 2006 renforce les aspects développement en équipe, sans délaisser le développeur. L'outil incorpore .net, ASP.net, VCL et VLC.Net ainsi que Win32. L'un des objectifs de

Borland est d'aider à la migration des projets Win32, en projet .net. Sur le cycle de vie, Delphi 2006 s'intègre à Together, CaliberRM et StarTeam. C++ Builder 2006 récupère les fonctions des autres IDE maison pour proposer aux développeurs C++ une plus large panoplie de fonctions : librairie VCL de Delphi, intégration aux outils de cycle de vie. Ces trois outils font partie de Borland Developer Studio. Il s'agit d'une plate-forme de développement Windows / .Net, d'un unique environnement. En achetant un des outils, on pourra bénéficier de tous les langages : C++, Delphi, C# et C. Vous avez encore quelques semaines, ces outils ne seront pas disponibles avant le 4e trimestre 2005. Together a connu peu avant, une mise à jour en version 2006 pour Eclipse. Il intègre outre UML 2, les concepts de modèle MDA, des patterns avec génération de codes, analyse de processus métiers BPMN et de rétro-documentation avec Live Source, sans oublier les métriques et audits de code. Il utilise la conception QVT : Requête, vue et transformation. Together 2006 pour Eclipse comprend les éditions développeur, architecte et designer.

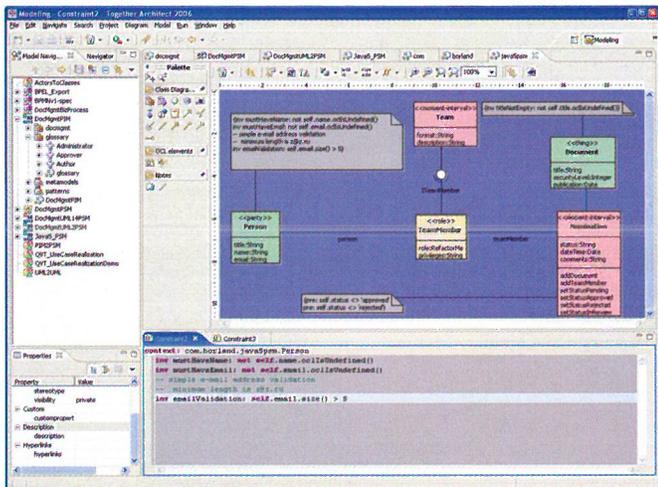
Site : www.borland.com

OUTILS

Software AG renforce son SOA

L'éditeur allemand renforce sa stratégie SOA annoncée en mars dernier, avec la sortie de Composite Application Integrator. Cet outil vise à publier et à concevoir des applications web riches plus rapidement en utilisant l'architecture SOA et l'interface Ajax. Il inclut un RAD pour concevoir visuellement l'interface et l'ensemble de l'application. On déploie ensuite sur un environnement Java, soit sur un navigateur, soit sur un client lourd Java. Elles s'intègrent aussi aux portails du marché (IBM, SAP...). Cette annonce fait suite au rachat de Casabac Technologies en août dernier. Pour la gestion des architectures SOA complexes, Software AG et Fujitsu s'associent. La première concrétisation est CentralSite, un référentiel pour gérer les composants SOA, afin de faire cohabiter les intervenants techniques et métiers. Il propose un format de métadonnées unique. Les deux éditeurs incluent le support dans CentralSite de leurs productions : Interstage Business Process Manager de Fujitsu, Enterprise Information Integrator et Enterprise Service Integrator de SoftwareAG (un ESB). L'objectif est d'étendre le support de CentralSite à d'autres outils SOA, web services durant les prochains mois.

Site : www.softwareag.com



XML

Altova se met déjà à 2006

L'éditeur spécialiste du XML dévoile la gamme 2006 de ses différents outils. XMLSpy, l'outil phare, inclut désormais un moteur XSLT 2 et Xquery et surtout le support de .Net et de Visual Studio 2005. Côté Java, l'outil s'intègre à Eclipse 3.1. Sur MapForce, l'outil d'intégration données - web services, il est possible de générer des web services visuellement. Il génère le code lié aux données pour Java et C#. L'outil SchemaAgent (gestion de schéma XML) s'intègre maintenant avec le mapping de données de mapforce. Et enfin semanticworks, l'éditeur RDF / OWL, qui autorise l'édition et la création visuelles des formats RDF, OWL, DL.

Site : www.altova.com

SUIVEZ LA VOIE DE L'AUTOMATISATION

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

Leonardi Logiciel Client Universel

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

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

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

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

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

Gestion de la persistance des données et des transactions.

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

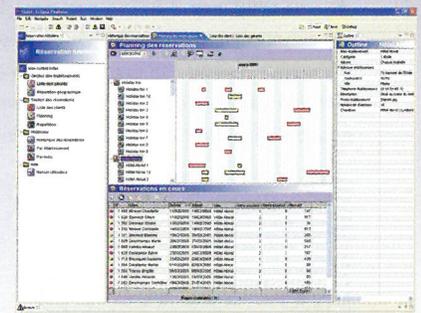
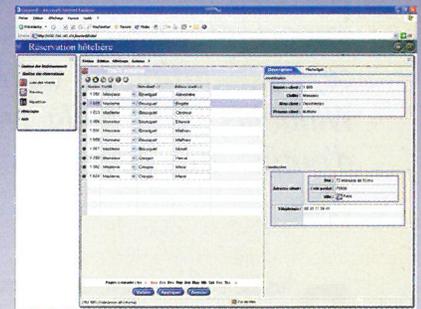
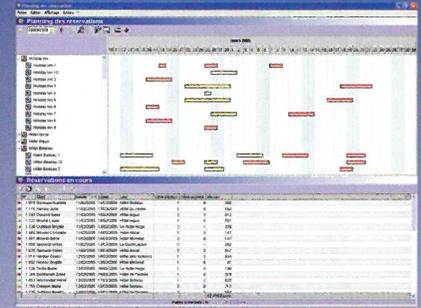
Livré avec Leonardi Studio, environnement de
développement intégré dans Eclipse.

Implémentation du code métier en JAVA.

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

Différents types de vues disponibles :

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



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



www.lyria.com

Client Software Solutions

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

Professional Developer Conference à Los Angeles :

Les annonces majeures de Microsoft



Microsoft a voulu montrer que les promesses faites en 2003 ont été tenues : Windows Vista et la WinFX existent bel et bien. Deux annonces importantes ont aussi marqué cette conférence, l'intégration du workflow à l'intérieur de Windows Vista et la création d'une palette d'outils de développement d'applications Web.

Plus de 8 000 personnes ont assisté du 13 au 16 septembre à Los Angeles à l'édition 2005 de la Professional Developer Conference (PDC), ouverte par Bill Gates. Beaucoup de nouveautés annoncées, mais le but de cette conférence était principalement de démontrer que les annonces fracassantes effectuées il y a deux ans ont bien été suivies d'effets. WinFX, les systèmes d'exploitation Longhorn, le nouvel Office, tout cela existe et est bien avancé. Malheureusement, rien

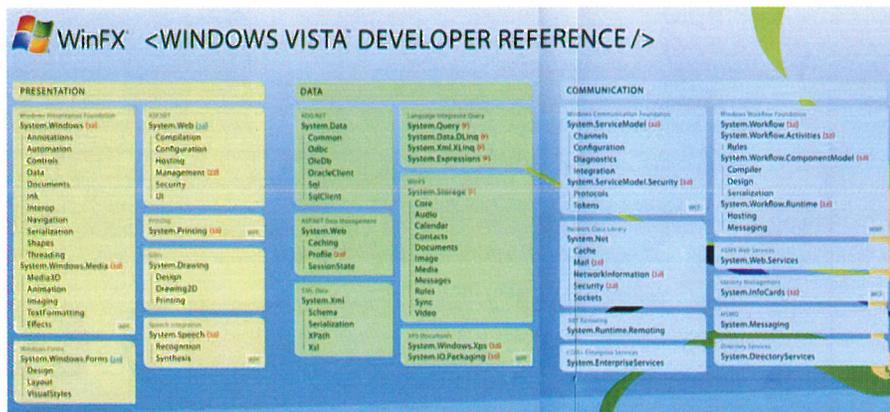
n'est encore achevé et il va encore falloir patienter au moins un an avant de pouvoir en profiter. En attendant, les développeurs pourront toujours travailler sur les versions bêta qui leur ont été distribuées.

Sept différentes versions pour Windows Vista

Dans un premier temps, la R2 de Windows 2003 Server devrait être disponible pour la fin de l'année. Cette version disposera d'une nouvelle console de gestion des fichiers DFS, d'une nouvelle console de gestion de l'impression, de gestionnaires de ressources de stockage pour les NAS et pour les SAN, d'une gestion du matériel avec la plate-forme IPMI (Intelligent Platform Management Interface), d'une meilleure interopérabilité avec Linux/Unix et enfin et surtout de l'Active Directory Federation Services, qui lui permettra de récupérer les identités d'utilisateurs extérieurs à l'entreprise (partenaires, fournisseurs...). Windows 2003 Server R2 bénéficiera également (comme par la suite les serveurs Longhorn) de la plate-forme WS-Management.

Les annonces les plus attendues concernaient Windows Vista. Ce système d'exploitation a bien évolué depuis les précédentes annonces. Comme prévu, le système de fichiers WinFS en est absent. Il faudra attendre enco-

re un ou deux ans après la sortie officielle du produit pour pouvoir en bénéficier. En revanche, il y a une vraie grande nouveauté, c'est l'intégration d'un moteur de Workflow, directement intégré à l'API. Des Windows Vista, il y en aura pour tous les budgets, car pas moins de sept versions sont déjà prévues ! Deux d'entre elles se destinent aux PME, une version Small Business et une version Enterprise Edition, les autres se différencieront par leurs possibilités multimédias, avec des déclinaisons familiales et professionnelles comme pour Windows XP. Ensuite, vers 2007, la version serveur de Longhorn devrait être lancée. Elle sera basée sur l'Active Directory Federation Services, et bénéfi-

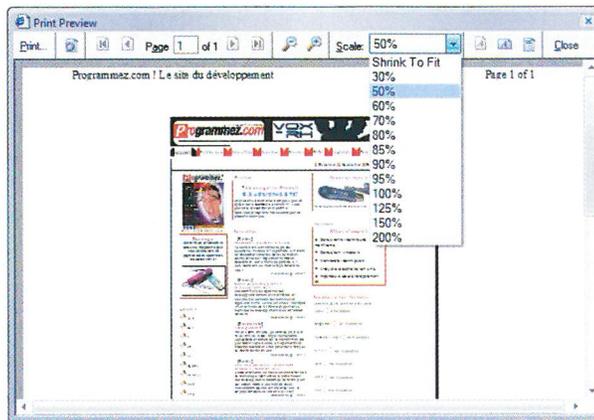


La WinFX s'enrichit d'un nouveau moteur de workflow

ciera de grandes possibilités de gestion des applications. Au WS Management, viendra s'ajouter un langage nommé Monad. Il s'agit d'un langage de script .NET qui permettra d'automatiser les tâches d'administration des applications. Monad pourra, par exemple, être utilisé pour réaliser les tâches de sauvegarde, ou servir encore à définir les actions à entreprendre lorsqu'une application ne fonctionne plus correctement.

La WinFX s'enrichit d'une couche workflow

Le développement d'applications pour ces nouveaux systèmes se fera avec WinFX et non plus avec la Win32. La couche de présentation Avalon s'appelle désormais Windows Presentation Foundation. Elle remplace l'ancienne GDI et y ajoute des fonctions 2D et 3D, du dessin vectoriel et des applications multimédias audio et vidéo. La couche de communication Indigo se divise en deux parties. La première est la couche de communication proprement dite nommée maintenant Windows Communication Foundation qui permettra de développer des Web Services et la deuxième est la toute nouvelle couche de workflow, baptisée sans surprise Windows Workflow Foundation. Ce moteur de workflow s'adresse au



La nouvelle fonction de Zoom d'Internet Explorer 7 améliore l'affichage et évite les gaspillages de papier

client et au serveur et sera utilisé par les futurs produits Microsoft tels que Office 12, BizTalk Server et Dynamics Products (anciennement nommés Business Solution).

À côté de ces systèmes, d'autres plus étonnants ont été présentés, comme Windows SideShow, le système qui fait tourner des applications lorsque l'ordinateur est éteint ! Il est souvent contrariant en effet d'allumer son ordinateur uniquement pour vérifier un numéro de téléphone. Le SideShow évite cela, puisqu'il permet d'accéder à de petites applications sans qu'il soit nécessaire d'allumer l'ordinateur, à condition bien sûr que ce dernier soit

spécialement équipé d'un petit écran supplémentaire et de boutons de navigation comme sur les PDA.

Des outils de création de pages Web

La PDC a également vu la présentation du langage de requête LINQ. Cette présentation ainsi que les démos qui ont suivi ont fortement impressionné l'assistance. LINQ (Language Integrated Query) permet à partir de langages orientés objets, d'accéder simplement et surtout, de la même manière, à toutes les sources de données, que ce soit des

Reprendre un 2^e croissant n'a jamais été aussi stratégique...

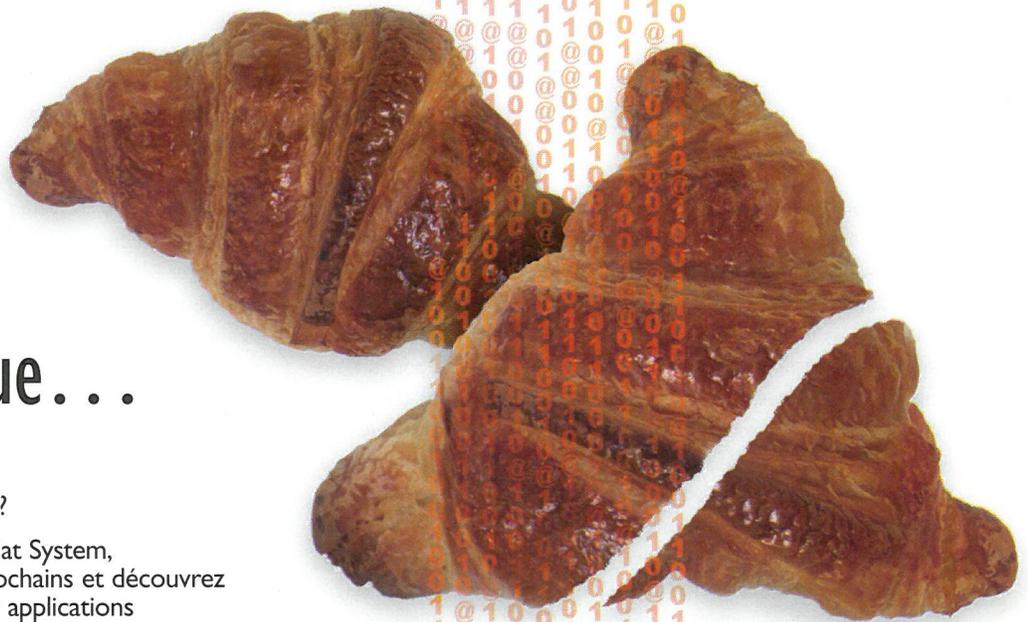
Vous souhaitez développer en J2EE ?

Participez aux petits déjeuners de Nat System, du 22 septembre au 8 décembre prochains et découvrez comment développer facilement vos applications de gestion en J2EE avec Nateo.

Editeur de NS-DK et NatStar, Nat System s'est attaqué avec succès aux nouvelles technologies. Pour plus de 500 grands comptes, les applications générées avec nos outils sont devenues stratégiques.

Et pour vous ?

Nat System donne un nouvel élan aux applications spécifiques métiers ; venez en discuter autour d'un -ou deux- croissants.



Nat System
Stratégiquement vôtre

*Inscrivez-vous à l'un de nos petits déjeuners sur www.natsystem.fr ou matinales@natsystem.fr ou par fax : 01 45 14 73 74

Nat System - Immeuble Le Péricecentre - 16 av. Louison Bobet - 94120 Fontenay sous Bois

bases relationnelles, ou des données en XML. Ce langage de requêtes qui s'appuie sur le .NET Framework est donc intégré aux langages de programmation C# ou VB .NET. Pour terminer avec le développement, notons qu'une nouvelle suite d'outils de création d'application Web a été dévoilée sous le nom d'Expression. Cette suite se compose de trois outils. Le premier, Acrylic Graphic Designer sera utilisé pour créer et modifier des images bitmap, le second Quartz Web Designer pour concevoir des sites Web et enfin le dernier, Sparkle Interactive Designer pour créer des interfaces graphiques vectorielles s'appuyant sur le langage XAML (XML Application Markup Language) pour Windows Vista et Internet Explorer. La présentation d'Expression a été suivie d'une démonstration très applaudie qui a montré les possibilités étonnantes de cette suite, ainsi que les capacités d'affichage de Vista avec des objets en 3D. Toutefois, Adobe et Macromedia risquent d'être moins enthousiastes, car bien qu'ils ne fonctionnent que sur Windows Vista, ces nouveaux outils entrent en concurrence directe avec leurs produits.

Office 12 : 100 % XML

Office 12 aura été l'autre grande vedette de cette PDC. Fini les menus et les sous-menus ! Dans cette dernière version, la navigation s'effectue plus simplement à l'aide d'onglets et d'une barre d'outils contextuelle. Si l'on insère par exemple un encadré dans le texte de Word, cette barre affichera les icônes de tous les types d'encadrés disponibles. De même, le simple fait de passer le curseur sur une police de caractères affiche l'ensemble de la page avec cette police, sans aucune autre manipulation. PowerPoint également bénéficie de modules de transformation de textes en graphiques et offre de nombreux effets visuels. Tous les documents créés par la suite Office sont désormais stockés au format XML. Cela va permettre à n'importe quelle application de créer ou de lire des documents Office, sans qu'il soit nécessaire de posséder la suite bureautique. Il faut noter la montée en puissance d'InfoPath et de Share Point qui voient leur champ d'utilisation augmenter.

Une arme anti-phishing redoutable

La dernière grande nouveauté est l'annonce de la version 7 d'Internet Explorer. Ce navigateur sera intégré à Vista, mais il sera également disponible avant, afin d'être utilisé sous Windows XP SP2 ou sous Windows Server 2003 SP1. La sécurité d'abord, point faible des anciennes versions, a été nettement améliorée. Le navigateur pourra tourner en mode protégé (sous Vista uniquement), mode dans lequel il ne pourra écrire nulle part ailleurs que dans les fichiers Internet temporaires (Temporary Internet Files) sans le consentement explicite de l'utilisateur. Les ActiveX seront désactivés par défaut et leur installation ne pourra s'effectuer que sous le compte administrateur. Enfin, un système anti-phishing est également prévu. Grâce à une base contenant tous les sites dangereux, l'utilisateur sera immédiatement averti par une URL affichée en rouge dans la barre d'adresse, lorsqu'il navigue dans des eaux troubles. Les modifications ont également porté sur l'interface. Les pages visitées sont affichées sous forme d'onglets sous la barre d'outils, ce qui permet d'y accéder beaucoup plus rapidement et plus simplement. La nouveauté la plus intéressante est la présence d'un outil de Zoom qui permet d'afficher la page à la taille désirée. La visualisation pourra donc être optimale quelle que soit la résolution d'écran, et surtout l'impression des pages pourra s'effectuer sans gaspillage de papier.

■ Alain KERHUEL

Microsoft

100 nouveaux produits en 18 mois !

Quelques jours après une conférence développeur, la PDC, très riche en annonces et présentations, Microsoft France a voulu préciser ou repréciser certains points, notamment sur le planning des 18 prochains mois. Microsoft s'intéresse à 4 grands acteurs dans l'informatique : le développeur, l'administrateur, l'utilisateur à la maison et l'utilisateur au bureau. Pour répondre à ces cibles, l'éditeur met en avant trois piliers : fiabilité, innovation, liberté. Que ce soit à la maison ou en l'entreprise, de nombreuses solutions sont ou seront proposées. Si on prend l'univers professionnel, tout tourne autour de Dynamic Systems Initiative qui permettra intégration, interopérabilité, flexibilité et communication des données et des applications. Pour réaliser cela, Microsoft annonce des mises à jour et des nouvelles versions majeures à la pelle. Pas moins d'une centaine de nouveaux logiciels en 18 mois ! Pour le développeur, il y a l'immédiat avec .net 2 et Visual Studio 2005, demain Windows Vista, WinFX, le 64-bit, le Web Services Enhancements 2, après-demain ce sera un nouveau .net, Visual Studio Orcas (avec de nouveaux langages), Office 12, WinFS encore plus loin... Pour l'entreprise, ce sera Vista Server, Vista Client, les nouveaux outils serveurs (Commerce Server, Live Server, Groove, BizTalk, Exchanger Server 12...

Comme on peut le constater, les annonces vont s'accélérer à partir de début 2006. Mais à la question, comment va-t-on digérer tout cela, les réponses demeurent assez floues. Car il ne faut pas oublier l'existant. De nombreux utilisateurs et entreprises fonctionnent encore avec du Windows 2000, du VB6, du SQL Server 7, du Windows NT 4, de l'Office 2000 voire antérieur, etc. La mise à niveau et la migration de tout cet existant ancien qui a une ou deux générations de retard pose un réel problème. Car au-delà du logiciel pur, pour la partie serveur, cela signifiera souvent de mettre à niveau son matériel, de vérifier la migration des documents vers les nouveaux formats, les nouvelles fonctions. Il s'agira aussi d'assurer la formation des utilisateurs, développeurs, administrateurs. Cela aura un coût financier, impossible à chiffrer à l'heure actuel.

Téléphonie IP et P2P

Doit-on pour autant être pessimiste et/ou prudent ? En partie oui, pour le reste, il faut reconnaître que Microsoft doit, sur certains points, combler un sérieux retard. Le système client et serveur n'ont pas connu de mises à jours majeures depuis plusieurs années. Dans le logiciel, la situa-



tion est parfois critique comme avec Internet Explorer. Il s'agit aussi d'offrir de nouvelles fonctions connaissant une forte demande : la téléphonie IP et le P2P (avec l'outil Groove). La qualité et la sécurité des applications et du système sont deux credo vitaux pour l'éditeur. Windows Vista constitue une étape importante car les nouveaux modes d'exécution ainsi que la disponibilité de futurs outils de sécurité (anti-virus, anti-malware) doivent donner une image sécurité de la plate-forme. Il fallait donc répondre à la concurrence et contrer l'offensive de l'open source et du libre. Office 12 en est un parfait exemple. Office, depuis la version 2003, n'est plus une suite bureautique mais une plate-forme de documentaire, de développement pour applications et documents métiers. Grâce à cette énorme mise à jour, Microsoft harmonise totalement ses fondations techniques avec .Net, XML, Web Services. En théorie, l'interopérabilité entre les applications permettra une intégration très poussée des données, de leurs flux. Cela facilitera les échanges, la manipulation, la transmission. L'annonce du Windows Workflow Foundation va dans le sens d'une harmonisation interne des outils Microsoft et inter applications, en proposant un modèle de flux unique reposant sur .net. Malgré tout, même si les standards vont permettre une ouverture vers les environnements extérieurs, tout cela demeure très Windows - Windows.

Ce chantier gigantesque mobilise les ressources de l'éditeur. En attendant Windows Vista Server, il faudra se contenter de la release 2 de Serveur 2003 qui inclura de nombreuses nouveautés (.net 2, nouvelles fonctions d'annuaires, gestion unifiée à distance...). Un service pack 3 pour Windows XP est prévu, mais pour le moment aucune date n'est avancée, Vista passant en priorité. D'autres technologies, plus discrètes prendront de l'importance dans l'avenir. On peut citer Monad, un langage de commande pour créer des interfaces .net vers des objets de management qui s'utilise en ligne de commande COM et .net, avec un accès uniforme aux données. Il sera accessible en shell et avec des langages scripts tiers

comme Python. La date de disponibilité n'est pas certaine (après Vista).

■ F.T.



Windows Server Longhorn

Plateforme d'application IIS 7 WinFX	Réseau NAP Perf débit QOS	Management Server Manager unifié MMC Snap-ins riche Un seul binaire
Terminal Services Redirection USB Console d'admin unifiée Accès à travers pare-feu Publication et Remoting	Stockage RDC APIs Quota APIs TxF	Nouvelle installation (image) Nouveau Task Scheduler Nouvel Event Log Composition
Les fondamentaux		
Fiable - Système de fichiers hot-plug - Self-healing file system	Sécurisé - Sécurisé à l'installation - Renforcement Services - Chiffrement du volume	Performant - Priorité des threads disque - Dynamic kernel memory pooling

3.248 heures pour développer le code...

184 pour trouver ce sacré bug!

142 heures de réunions...

108 pizzas!

14 week-ends à l'eau!

11 nuits blanches...

Protégez tout d'un seul appel!

HASP®

SOFTWARE DRM

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

1	Développement		<ul style="list-style-type: none"> • Pour Windows, Mac OS, Linux
2	Protection unique		<ul style="list-style-type: none"> • Cryptage matériel des données au standard AES 128 bits • Encapsulation renforcée et obscurcissement du code pour déjouer toute tentative de débogage ou reverse engineering • Générateurs de code d'API pour simplifier l'implémentation et la personnalisation
3	Multi-distribution		<ul style="list-style-type: none"> • Gestion des licences mono et multi-utilisateurs • Mise à jour sécurisée des licences déjà déployées sur le terrain • Modèles avancés de gestion des licences, définis séparément des mesures de protection



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



Aladdin®

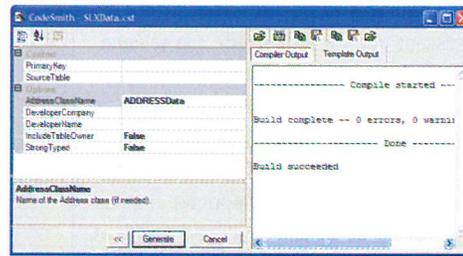
SECURING THE GLOBAL VILLAGE

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

CodeSmith : génération automatique

CodeSmith est un générateur de texte (code source) basé sur des schémas (application de templates de code sur des méta-données). Par exemple, il générera le code source de procédures stockées en se référant à une table d'une base de données. L'outil vous aidera à automatiser l'écriture de certaines parties d'une application (génération d'une couche de persistance, d'un prototype) ou à obtenir du code fortement typé (génération de collections typées). Concrètement, CodeSmith sera surtout utilisé par le développeur afin d'économiser du temps lors de la création de code répétitif. La syntaxe du schéma Design Pattern d'extension « .cst » est très proche de la syntaxe ASP.NET. En voici un court extrait :

```
<%@ CodeTemplate Language="C#" TargetLanguage="Text"
  Description="Simple template to show main syntax" %>
<%@ Property Name="Filter" Default="*.cst"
  Type="System.string" Category="Masks"
  Description="Mask for files in the directory" %>
...
Response.WriteLine
  ("List of files in template directory (using mask "
  + Filter + ")");
...
DisplayDirectoryContents(Filter);
</script>
```



C'est l'attribut « Language » qui définit le langage utilisé (ici du C# mais vous pouvez aussi employer CodeSmith avec VB.NET par exemple, ou avec n'importe quel autre langage Java y compris, car CodeSmith peut générer n'importe quel

texte). CodeSmith s'intègre parfaitement à VS.NET sous la forme d'un add-in. Il suffit de sélectionner le schéma qui vous intéresse et de le faire glisser dans la fenêtre de code source pour qu'une boîte de dialogue s'ouvre. Si vous laissez les défauts, il ne vous reste plus qu'à générer. La version pro comporte un IDE de création de template. Les dernières

versions comportent aussi un gestionnaire de debug et le support de l'XML comme source. En résumé, la génération de code est très pratique (même si vous pourriez envisager d'arriver au même résultat en partant du XML en utilisant des XSLT)... Enfin, si vous désirez réaliser du mapping objet relationnel avec CodeSmith rendez-vous à l'adresse <http://cstemplates.sourceforge.net/> pour y télécharger des templates.

Fiche technique

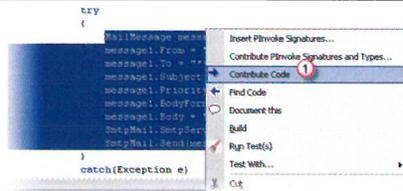
Nom de l'outil : CodeSmith version 3.1
Disponibilité et prix : Commercial ou 30 jours à l'essai (+ 50 € version standard ou + 300 € version pro), la version 2.6 est gratuite.
Éditeur : CodeSmith
Configuration : Windows XP/.NET
Téléchargement : www.codesmithtools.com/license.aspx
Les + : Gain de temps, code consistant, intégration à VS.NET
Les - : IDE non compris en standard

■ Xavier Leclercq

CodeShare : partager les codes

CodeShare est un add-in pour Visual Studio.NET qui permet de partager des bouts de codes en entreprise. Après son installation vous obtenez deux options supplémentaires lors de la sélection (menu contextuel) d'un bout de code dans votre IDE. Soit vous décidez d'ajouter le bout de code au dépôt (central repository), soit vous demandez de trouver un bloc de code présent dans la base. Pour que cela fonctionne il y a trois éléments à installer : la base de données (qui s'initialise en exécutant le script codeshare_sql.sql), le Web Service, qui s'installera sur un serveur de votre intranet d'entreprise, et l'add-in qui devra être installé sur chaque poste client.

La nouvelle entrée du menu contextuel « Contribute Code » permet de stocker dans le dépôt le bout de code sélectionné. Un autre développeur situé sur votre intranet pourra ainsi extraire ce code en utilisant le menu contextuel « Find Code ». En ce qui concerne la sauvegarde et la recherche vous devez définir un mot clé. Vous pouvez aussi enregistrer votre nom (par défaut le login courant) ce qui permet de filtrer la liste par utilisateur. A côté de l'add-in existe un outil indépendant du nom de CodeShare



Fiche technique

Nom de l'outil : CodeShare
Disponibilité et prix : Gratuit

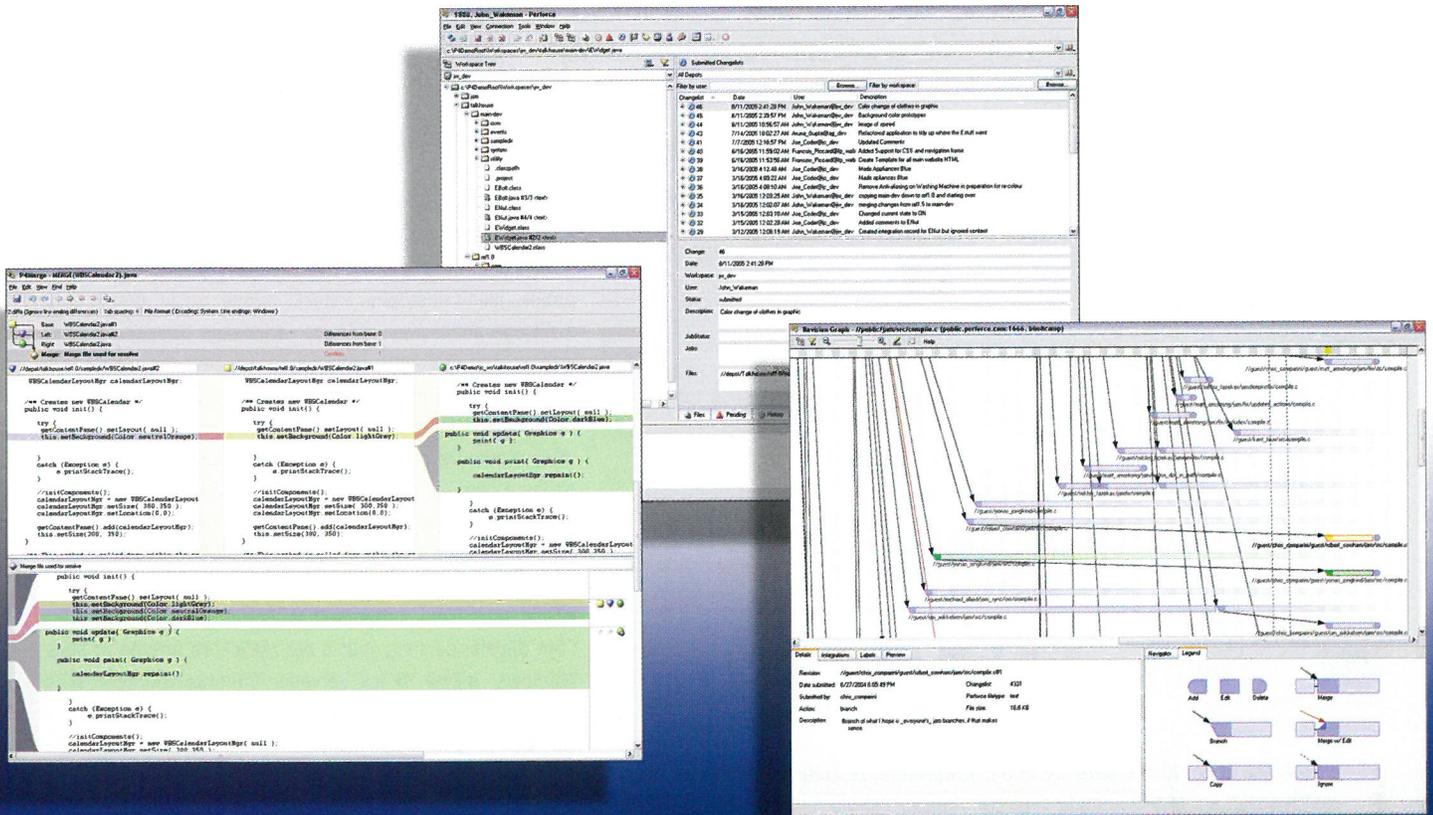
Auteur : Nauman Leghari
Configuration : Windows XP/.NET
Téléchargement : <http://weblogs.asp.net/nleghari/archive/2004/12/01/273100.aspx>
Les + : Très simple d'utilisation une fois l'installation terminée, outil CodeShare client indépendant de VS.NET
Les - : Difficulté de déploiement sur de nombreux postes

Client. L'installation par défaut permet une intégration à Internet Explorer. A l'aide de cet outil, vous pouvez copier/coller du code le plus simplement du monde sans passer par Visual Studio. En fait l'add-in ne fonctionne que dans la fenêtre du code et non avec un fichier aspx, html ou un fichier de configuration. CodeShare Client permet de contourner cet inconvénient. Pour l'intégrer à VS.NET il suffit de l'appeler comme outil externe (Tools->External Tools) puis de créer des raccourcis clavier pour l'invoquer (Tools->Customize). ■ X. L.

Gestion de configuration logicielle Perforce

Puissance

et évolutivité



[Rapide]

[Évolutif]

[Distribué]

Perforce assure le suivi et la gestion du code source et du contenu multimédia pour vous offrir un contrôle maximal avec un minimum d'ingérence.

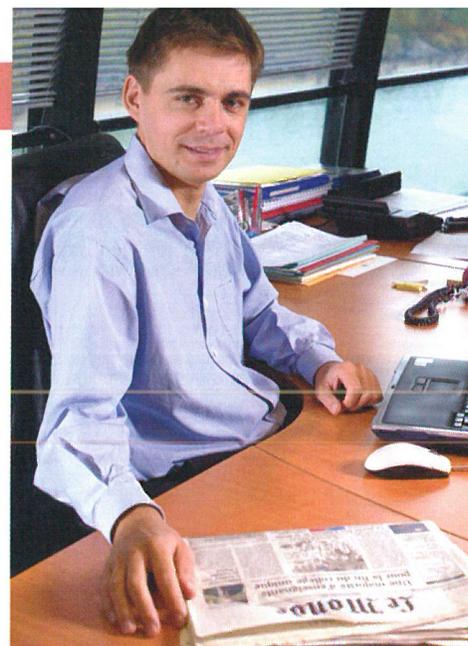
Perforce autorise l'intervention de milliers d'utilisateurs simultanés. Ce logiciel gère les bases de code de plus d'un million de fichiers. Il traite des téraoctets de données - du code source aux documents, en passant par le contenu Web et les fichiers d'images.

Perforce est un système GCL à la fois rapide et performant. Peu importe la taille de l'équipe et le volume de données. Perforce relève le défi !

PERFORCE
SOFTWARE

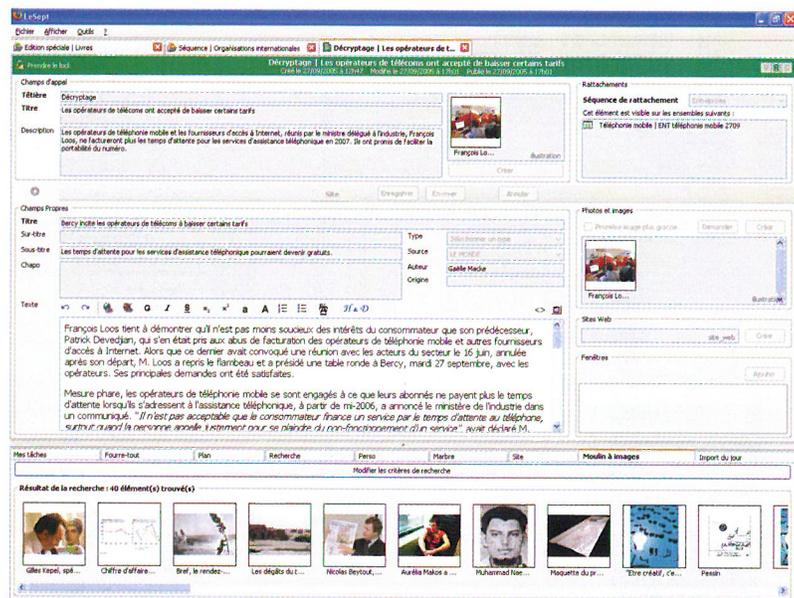
Téléchargez sans conditions une copie gratuite de Perforce sur www.perforce.com. Vous bénéficiez d'une assistance technique gratuite pendant votre période d'évaluation.

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



Le Monde se met à XUL

Le premier quotidien français travaille depuis 2003 à refondre une partie de son informatique et notamment son outil de publication interne servant aux journalistes. Récemment, le journal a décidé d'utiliser le langage XUL pour son outil stratégique...



Le tournant open source date de 2003. Jusqu'alors, le cœur applicatif reposait sur la technologie Vignette, dont Le Monde fut le premier client français. Vignette servait aussi bien à l'outil de publication qu'au back office. La décision fut prise de passer à l'open source en migrant de Vignette à PHP. La migration de l'outil de publication alla donc de soi. Cet outil sert au journal pour la création de contenus destinés au journal papier et au journal en ligne. Les journalistes saisissent leurs textes, incluent les photos, puis le texte passe dans un workflow interne entre les différents services. Pour la base de données, élément critique, Le Monde demeure « conservateur », en restant sous Oracle. Actuellement en version 8.1.7, la migration vers la 10g est en passe d'être effective, les derniers tests ayant été réalisés. L'application de publication, et donc l'infrastructure, sert aussi bien à la rédaction du site, du journal papier qu'aux commerciaux. Presque 200 articles et contenus multimédias sont montés et publiés tous les jours.

Du XUL en interface

Le projet SEPT (Système Editorial de Production et de Trafic) doit apporter une interface riche avec de nouvelles fonctions plus conviviales. « Avant, on utilisait du HTML classique. On souhaitait aller plus loin : proposer une interface plus conviviale, une fonction de glisser – déposer... », précise Jean-Christophe Potocki (directeur informatique du

Monde.fr). Pour ce faire, Le Monde a décidé d'utiliser le langage descriptif XUL soutenu par Mozilla.

Si XUL était en phase avec les objectifs technologiques du projet, Jean-Christophe Potocki avoue qu'au début, il y avait une petite peur, même si le langage n'est pas récent. « On pense que cette technologie est pérenne, et un langage d'avenir » poursuit-il. Au début du projet, le manque de documentation a été un peu pénalisant mais sans être dramatique. De plus, l'utilisation de XUL sur le poste client oblige à installer des navigateurs compatibles, Le Monde a donc déployé sur les ordinateurs, Firefox. « On utilise essentiellement des PC. L'application XUL fonctionne aussi sur Mac, nous en avons quelques-uns, et nous avons aussi vérifié sous Linux », poursuit M. Potocki.

Côté développement, tout a été réalisé en interne et sous Windows, même si les serveurs frontaux tournent sous Linux. Les outils de développement n'ont pas été imposés aux développeurs. Les éditeurs du type Vi sont très utilisés. Pour gérer les versions, WinCVS a été déployé. « Passer à PHP ne fut pas très difficile, à XUL un peu plus. Cependant, c'est un langage strict, il possède une approche structurée et impose une bonne séparation des couches. Ce n'est pas révolutionnaire mais XUL constitue une nouvelle méthode de développement, de conception », conclut M. Potocki.

Fluidité des données

Ce projet PHP – XUL a permis de mieux répartir la charge des traitements entre le serveur et le poste client. Ainsi, aujourd'hui, certaines fonctions peuvent s'exécuter directement sur le poste de l'utilisateur, évitant ainsi d'interroger et de charger le serveur et la bande passante. « On répartit plus que l'on ne minimise. On a beaucoup travaillé sur les transferts de données », précise M. Potocki. Pour assurer la fluidité des données entre la base Oracle et le serveur PHP / Interface, une couche applicative a été spécialement conçue. Pour réaliser le flux des données, l'ensemble travaille en objet et le mapping se fait en objet. Le tout transite par un web service avant d'être traité par le serveur, puis affiché par l'interface. Pour concevoir les web services, Le Monde a opté pour wddx. WDDX facilite les échanges de données entre différents langages. Il s'interface aisément avec PHP et XUL.

■ François Tonic

TEST

Du nouveau pour Bugzilla

Comment rendre Bugzilla plus attractif, avoir une interface plus fouillée et fournir une meilleure productivité ? La réponse est : le client Java Deskzilla. Il inclut un moteur de recherche, une arborescence des bugs, un système de cache, un fonctionnement en ligne et hors ligne. Un excellent complément pour ceux l'utilisant quotidiennement ! Il est vendu 99 dollars. Site : <http://deskzilla.com>

OUTILS

Wily se met aussi au SOA

L'éditeur Wily Technology sort la version 6 de son outil Wily, un environnement de gestion d'applications d'entreprise. L'éditeur est connu par son outil Introscope pour le code Java. Avec Wily, il est possible de monitorer les transactions et de mesurer les performances des applications J2EE et des portails. La version .net est prévue, à une date non communiquée. Il s'agit de sortir de la vision boîte noire des architectures J2EE / XML / SOA. Une des principales nouveautés est la possibilité de remonter les bugs et erreurs générées dans la JVM même. Ainsi, l'outil identifie les composants et classes mis en cause. Il supporte Java 5.

CONFÉRENCE

BEA fait son show à Paris

Mi-octobre, se tenait à Paris le BEA World, rencontre entre l'éditeur, les développeurs et les utilisateurs. BEA en a profité pour annoncer quelques nouveautés. Tout d'abord, le rachat de ConnecTerra, société spécialisée dans les solutions RFID (marché évoluant très rapidement), fonctionnant dans des architectures SOA. Toujours dans le SOA, BEA avait dévoilé fin septembre son WebLogic RealTime Edition pour garantir les temps de réponses des applications J2EE WebLogic.

agenda

NOVEMBRE

9 et 10 novembre
Paris 7; 84 rue de Grenelle
5^e édition du Forum PHP,
organisé par l'AFUP
2 jours de conférences, avec
la présence de Rasmus Lerdorf.
www.afup.org/pages/forumphp

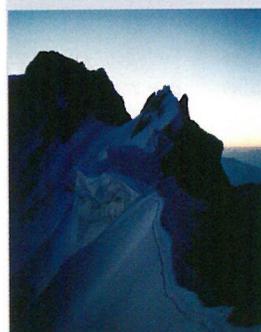
Les Matinales Nat System,
éditeur d'outils de développement J2EE
organise des petits déjeuners, dans ses
locaux, à Val de Fontenay (94),
les **jeudis 10, 24 novembre et**
8 décembre, de 8h à 10 h
www.natsystem.fr

Du 23 au 24 novembre
Paris-La-Défense
Salon de la sécurité informatique
Le rendez-vous annuel du marché de la
sécurité des systèmes d'information
www.infosecurity.com.fr

24-25 novembre
Paris (Gentilly)
Formation Scrum
formation "officielle" et certifiante de 2
jours, en français, tarif: 995 € H.T. les deux
jours, par personne, places : 20 maxi.
Scrum est une méthode agile, orientée
sur les pratiques de gestion de projet plus
que sur les aspects techniques,
"compatible" avec Extreme Programming.
www.controlchaos.com

23 & 24 novembre 2005
CNIT Paris / La Défense
FORUM INTEGRATION 2005
Une nouvelle édition en adéquation avec
les problématiques actuelles du marché
de l'Intégration, de l'Urbanisation et
l'Architecture des Systèmes
d'Information... une large place sera
consacrée aux SOA, à l'urbanisme des
SI, à l'EAI, à la gouvernance et au BPM.
www.integration.fr

Formations
pour l'
Open Source



30 formations
aux technologies libres

- Apache
- OpenOffice
- Java
- MySQL
- Linux
- PHP
- PostgreSQL
- The GIMP
- Perl

Prochaines dates
21/11/2005
12/12/2005



<http://www.anaska.fr>

« Maintenant, MySQL a les technologies pour aller parler aux entreprises »

A l'occasion de la sortie de MySQL 5.0, nous avons rencontré David Axmark, co-fondateur de MySQL AB. Avec le lancement de cette nouvelle version de sa base de données, il espère se faire une place dans le cercle très fermé des trois leaders du marché.



David Axmark, co-fondateur de MySQL AB

Les procédures stockées vous permettent désormais d'attaquer le marché de l'ERP, mais est-ce suffisant ?

Non, bien sûr. Mais maintenant que nous avons les technologies, nous pouvons parler aux intégrateurs et éditeurs afin de nouer des partenariats, établir des accords et certifier MySQL 5.0 sur des plates-formes, afin d'apporter aux entreprises une solution fiable, moins chère et performante.

Pensez-vous disposer de la crédibilité nécessaire, en termes d'image notamment, pour convaincre les entreprises ?

Nous sommes pour l'instant surtout présents sur du site Web, c'est vrai. Mais un site Web est aussi une application critique qui a besoin de haute disponibilité. MySQL AB va encore une fois devoir grossir pour apporter le support,

négoier les partenariats, etc. qu'impliquent ces nouveaux marchés. Mais nous n'avons cessé de grossir depuis notre création, ce n'est donc pas un problème nouveau pour nous.

Comment comptez-vous vous imposer avec une version qui finalement ne propose que de « vieilles » technologies intégrées par vos concurrents depuis longtemps, y compris chez PostgreSQL ?

Si PostgreSQL veut être la base la plus avancée dans l'univers de l'open source, c'est parfait. Parce que nous n'avons jamais cherché à jouer sur ce terrain là : nous voulons qu'elle soit la plus simple et la plus fiable. Enfin, il reste encore un gros marché d'applications sans base de données. On est aujourd'hui tout à fait conscient qu'on ne peut pas rivaliser avec Oracle, mais nous avons tout de

même des arguments convaincants pour répondre à ce marché.

La montée en puissance fonctionnelle de MySQL implique de nouveaux outils d'administration. Disposez-vous par exemple d'outils pour tracer les requêtes distribuées de type XA ?

Non, pas encore. Mais nous y travaillons et ce n'est d'ailleurs pas le seul outil d'administration qui nous manque aujourd'hui. En attendant, nous avons des accords avec des partenaires, comme Embarcadero ou Quest qui comblent le manque.

■ Propos recueillis par Marie Varandat

Oracle rachète le moteur de bases de données open source Innobase Oy

Difficile de savoir si Oracle est simplement atteint de fièvre acheteuse ou si sa dernière opération cache une autre intention que celle d'investir dans l'open source. Avec la société finlandaise Innobase Oy, développant le moteur de bases de données InnoDB, Oracle, qui a déjà dépensé 18 milliards de dollars en rachat cette année, porte ses acquisitions au nombre de sept en à peine 12 mois. Mais pourquoi le leader du marché de la base de données aurait-il besoin d'un nouveau moteur de base de données transactionnel ? Régi par une licence GPL ou propriétaire, InnoDB est un moteur de stockage doté de capacités transactionnelles, actuellement distribué par MySQL AB. En d'autres termes, InnoDB est la solution la plus populaire pour enrichir MySQL de fonctions transactionnelles, de verrouillage, etc. avancées et ainsi, lui faire passer le cap de la simple base en lecture utilisée par les sites Web. Solution aujourd'hui détenue par Oracle, qui a ainsi les moyens désormais



Kaj Arnö, Vice Président des Relations avec la communauté Open Source de MySQL

d'influencer l'un de ses principaux concurrents sur le marché des bases de données en environnement open source. Kaj Arnö, Vice-Président des Relations avec la communauté Open Source de MySQL se veut toutefois rassurant : « Avec ce rachat, Oracle reconnaît MySQL

AB et le marché de la base de données open source ». Sur les conséquences pour MySQL, il ajoute : « Innobase fournit le moteur de stockage le plus populaire de MySQL. Mais une des choses que nos utilisateurs apprécient le plus dans MySQL est la capacité de son architecture à recevoir différents moteurs de stockage. Vous pouvez choisir parmi plusieurs, dont MyISAM, Memory, Merge, Cluster and InnoDB et dans la version 5 nous en avons même ajouté un autre ». Mais derrière le calme apparent des dirigeants de MySQL AB, la communauté des développeurs s'agite sur les forums. « Peut-être l'heure de passer à SQL Server » plaisantait même un développeur que l'ombre d'Oracle semble faire frémir...

DÉVELOPPEMENT RAD
 GÉNÉRATEUR DE CODE
 ÉDITEUR DE FORMULAIRE
 ÉDITEUR D'ÉTATS RAPIDES
 BIBLIOTHÈQUE D'OBJETS
 LANGAGE DE 4^{ME} GÉNÉRATION (L4G)
 ÉDITEUR D'ÉTIQUETTES
 GÉNÉRATEUR DE GRAPHES
 GÉNÉRATEUR D'APPLICATIONS
 LANGAGE EN FRANÇAIS
 CORRECTEUR ORTHOGRAPHIQUE
 ÉDITEUR DE MÉTHODE
 DÉBOGUEUR INTERACTIF
 COMPILATEUR INTÉGRÉ
 SGBD/R TRANSACTIONNEL SÉCURISÉ
 IMPORT / EXPORT MULTI-FORMATS
 SAUVEGARDE INTÉGRÉE
 ODBC INTÉGRÉ
 CONNEXEURS ADO
 API POUR JAVA
 SUPPORT OCI (ORACLE)
 PLUG-INS ET COMPOSANTS
 SUPPORT DES BLOBS CONNEXEURS
 MYSQL, POSTGRES
 COMPATIBILITÉ TSE/CITRIX
 COMMON GATEWAY INTERFACE
 SERVEUR WEB INTÉGRÉ
 SÉCURITÉ SSL 128 BITS
 CLIENT SMTP/POP3/IMAP
 CLIENT FTP
 CLIENT HTTP
 PUBLICATION DE WEB SERVICES
 RECHERCHE EN NOTATION XPATH
 PARSER DOM ET SAX
 SERVEUR SQL
 INVOCATION DE WEB SERVICES
 SOAP, RPC ET WSDL
 TRANSFORMATIONS XSL (XSLT)
 IMPORT/EXPORT XML
 DIALOGUE D'IDENTIFICATION
 INTERFACE NATIVE WINDOWS XP
 INTERFACE NATIVE MAC OS X
 DÉPLOIEMENT AUTOMATIQUE
 SERVEUR D'APPLICATION



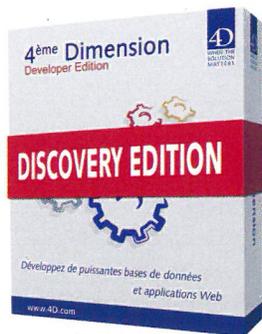
WHEN THE
 SOLUTION
 MATTERS*



Nantes
 Lille
 Bruxelles
 Bordeaux
 Toulouse
 Montpellier
 Marseille
 Lyon
 Paris
 Strasbourg
 Lausanne

Êtes-vous satisfait de vos outils de développement ?

Venez rencontrer nos ingénieurs sur le 4D Tour 2005.
 Découvrez l'originalité de 4^{ème} Dimension et repartez avec votre pack "Discovery Edition".

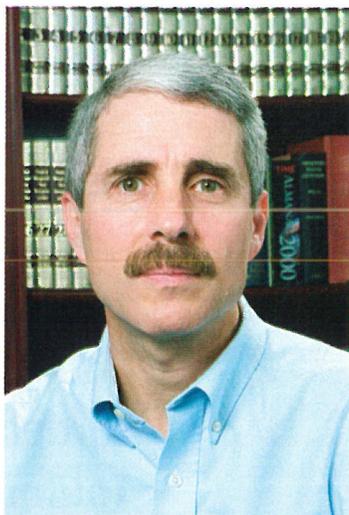


- IDE moderne intégrant un SGBD/R transactionnel et un serveur Web sécurisé.
- Langage de 4^{ème} génération enrichi de mécanismes RAD.
- Support intégral des standards : SQL, XML, XSLT, ODBC, OCI, ...
- Utilisation automatique en architecture orientée services : SOAP, .NET, J2EE...
- Scalabilité de 1 à plusieurs centaines d'utilisateurs simultanés.
- Pérennité des développements grâce à la compatibilité ascendante totale.
- Écriture unique du code pour Windows, Mac OS, ...
- Extensibilité et réutilisation : plug-ins, composants, bibliothèques d'objets.
- Sécurité à tout niveau : accès, données, transactions.
- Frontal des principales bases de données : Oracle, IBM DB/2, MS SQL Server, PostgreSQL, MySQL...

Leader des outils de développement multi-plateformes

** détails sur le site

Inscrivez-vous : www.4D.fr/tour2005



« Vers plus de complexité incontrôlable »

A l'occasion de son passage à Paris, nous avons rencontré Ross Altman, ancien « chief technology officer » de SeeBeyond, fraîchement nommé au poste de CTO de Sun Microsystems pour la division SOA & Business Integration. Convaincu que le rachat de SeeBeyond par Sun change la donne sur le marché de l'intégration, il reconnaît également que les entreprises s'orientent vers des architectures de plus en plus complexes, qu'elles vont avoir du mal à administrer.

Programmez : en quoi le rachat change-t-il la position de SeeBeyond sur le marché de l'intégration ?

Ross Altman : Nous sommes plus gros ! Sérieusement, cela change surtout notre position vis-à-vis de Tibco, notre principal concurrent, car, adossé à Sun, nous pesons plus lourd sur le marché et nous apportons plus de garanties de pérennité. Ceci étant, être gros n'est pas toujours un avantage, il suffit de regarder IBM ! C'est d'ailleurs un phénomène dont Sun est conscient et dont nous avons été préservés : nous sommes aujourd'hui la division intégration de Sun, mais nous avons conservé la même organisation que chez SeeBeyond, avec les mêmes équipes, celles de Sun travaillant jusqu'alors sur l'intégration ayant rejoint notre formation. Nous restons donc aussi efficaces qu'avant le rachat.

Programmez : le rachat va-t-il entraîner des changements dans votre calendrier ?

Ross Altman : Non. On va continuer à gérer les systèmes d'exploitation que l'on gérait auparavant et à supporter les trois grands serveurs d'application du marché (BEA, IBM et celui de Sun). Nous allons également nous focaliser sur l'implémentation de JBI (Java Business Integration) et profiter dans ce cadre, des avancées et travaux effectués par Sun sur la JSR (Java Specification Request) 208. Nous pensons qu'il s'agit vraiment d'un standard majeur qui va relancer le marché du middleware

Programmez : comment voyez-vous les architectures informatiques évoluer ?

Ross Altman : Nous allons vers de plus en plus de complexité que nous ne pourrions pas vraiment contrôler. Ou plus exactement, le succès de ces nouvelles architectures sera accidentel et occasionnel. La formule est à peine exagérée, car les standards actuels n'adressent qu'une partie infime de nos problèmes. En outre, il existe deux sortes de standard dans le domaine de l'intégration : ceux de l'interopérabilité et ceux de la portabilité. Côté interopérabilité, il se passera bien 10 ans avant que les Web Services ne soient en mesure de jouer leur rôle. Côté portabilité, nous sommes déjà confrontés à de gros problèmes de sémantique, chaque industrie et éditeur utilisant sa propre convention de nommage. En d'autres termes, il faut briser les frontières jusqu'à ce qu'un protocole SOAP, ou autre, fasse la même chose au niveau applicatif que TCP/IP a

fait au niveau réseau. Or, cette intégration « plug & play » dont on parle actuellement est loin d'être une réalité, le plus gros problème à résoudre aujourd'hui étant lié à la sémantique. Concrètement, le middleware a encore de beaux jours devant lui !

■ Propos recueillis par Marie Varandat

IBM rénove son ESB à la sauce SOA

La médiatisation de l'ESB (Enterprise Service Bus) n'ayant pas manqué de ricocher sur IBM, Big Blue vient d'annoncer la gamme SOA Foundation. Fortement estampillée Websphere, cette offre est articulée autour d'un ESB qui assure le routage et la transformation, avec une connectivité de type services web. Une version plus avancée comprend également Websphere Message Broker et de nombreux protocoles d'échange, dont HL7 (santé), SWIFT (banque) ou encore ebXML, Tibco Rendezvous, etc. Les deux versions regroupent de nombreux outils, dont l'atelier de modélisation UML Business Modeler, ou encore, le moteur d'orchestration de processus WebSphere Integration Services, un annuaire UDDI... et bien entendu le serveur d'application WebSphere, plus ou moins musclé selon la mouture.

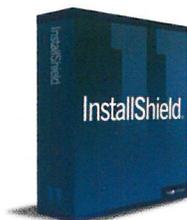
Progress met OpenEdge à l'heure de SOA

Dans sa version 10.1, la plate-forme OpenEdge de Progress intègre un nouvel environnement IDE basé sur Eclipse, qui incorpore de nouvelles structures de langage et mécanismes conçus pour faciliter la conception et la mise en œuvre de composants de service réutilisables, adaptés aux applications orientées services. Cette nouvelle mouture bénéficie également d'une meilleure intégration à l'offre de Sonic, filiale de Progress, afin de faciliter le travail des développeurs qui doivent mettre en place des process inter-entreprise notamment.



Les outils d'installation obsolètes peuvent provoquer des accidents.

28% de l'ensemble des pannes d'application sont liées à des erreurs d'installation et de configuration. Le meilleur moyen d'éviter cela est d'utiliser les tous derniers outils d'installation. Grâce à InstallShield 11, vous êtes à jour pour la prise en charge des applications Oracle, IIS6.0, MSI 3.1 et RPM, ainsi que des objets tiers. De plus, InstallShield 11 fait partie de la plate-forme FLEXnet™ de Macrovision, leader des solutions pour gérer et optimiser la valeur de vos logiciels.



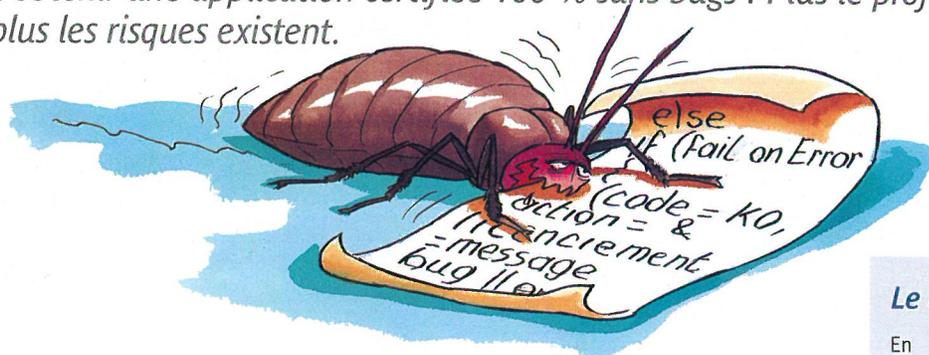
InstallShield®

Si vous achetez InstallShield 11 avec un accord de maintenance, les prochaines mises à jour seront incluses.

Pour en savoir plus, visitez le site www.installshield.com/current
ou appelez-nous dès aujourd'hui au numéro suivant :
0044 (0)117 9030650.

Tests : le défi de votre code

Depuis toujours, le bug, l'erreur, le dysfonctionnement existent. Il est impossible d'obtenir une application certifiée 100 % sans bugs ! Plus le projet se complexifie, plus les risques existent.



La notion de tests rejoint désormais celle du cycle de vie de l'application, de la qualité, du débogage, des règles de développement. Pour faire simple, on peut dire que plus le développement est structuré, avec une méthodologie de conception, une architecture fiable, des bonnes pratiques, et une préparation très en amont des tests, plus le projet a des chances de minimiser le bug. Un constat s'impose de facto : si l'on prévoit les tests tardivement dans un développement, il sera difficile de tester et de corriger l'erreur. Plus le bug est découvert tard, plus il coûte cher en temps et en argent ! Si aujourd'hui, on parle beaucoup des tests unitaires, le test couvre un spectre bien plus large. Les tests unitaires ne représentent qu'un maillon de la chaîne qui comprend en vrac : les tests manuels, fonctionnels, de non régression, de performances, de montée en charge, de couverture de code, etc. Bien entendu, un bon test s'accompagne d'un cycle de vie de celui-ci et de l'erreur découverte. Car, si on ne gère pas l'erreur - découverte, contexte, reproduction, localisation, priorité, correction, non régression, validation de la correction -, à quoi sert le test ?

Le rôle du testeur devient primordial. Celui-ci doit acquérir une vision globale du projet pour pouvoir bien tester. Malheureusement, le test continue à être parfois dévalorisé, car perçu comme ingrat et long. Cette phase représente entre 30 et 40 % du temps total d'un projet !

Dans ce dossier, nous avons voulu apporter des réponses concrètes, des idées réelles pour vous

aider à mieux appréhender le bug, sa recherche, sa correction et définir une stratégie de tests. Vous verrez que l'on tombe rapidement dans la notion de qualité, de qualité de code. Le tout avec un soupçon de théorie, d'abstraction, sur comment on crée un bug dans son code...

■ François Tonic

80% du temps de développement consacré aux erreurs

Selon une étude réalisée par le Ministère américain de l'Industrie et du Commerce, les bugs logiciels coûtent chaque année 59,5 milliards de dollars. Ce préjudice se répartit par moitié entre les clients et les sociétés de développement. Selon cette même étude, environ un tiers de cette somme pourrait être économisée, si des techniques de prévention et de tests adéquates étaient mises en œuvre de façon systématique.

La nature même des erreurs serait liée à la complexité. Quarante pour cent du temps de développement serait ainsi consacré à la détection des erreurs et à leur correction.

■ Jacques de Schryver

Le coût des erreurs

En fin d'année dernière, la France connaissait trois pannes géantes consécutives. D'abord chez France Télécom où une « anomalie logicielle » prive plusieurs régions de téléphone fixe pendant le week-end de la Toussaint. Cause de la panne : « une anomalie logicielle localisée dans un équipement de traitement de la voix sur IP situé à Reims, qui a provoqué des anomalies dans le formatage de la numérotation de certains appels, et déclenché les protections de sécurité sur des commutateurs ».

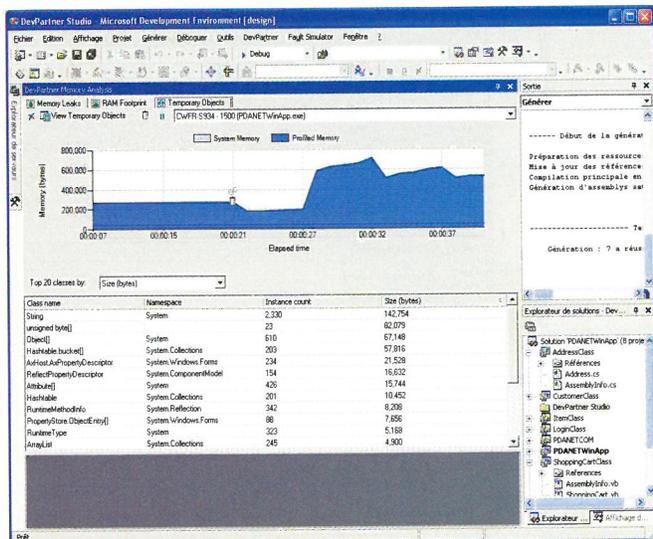
Le 17 novembre 2004, deux serveurs distants tombent en panne en même temps chez Bouygues Telecom, privant 7 millions d'abonnés de l'usage de leur téléphone. La panne informatique provient du dysfonctionnement de la base de données servant à repérer les mobiles des clients. Coût estimé de la panne : 16 millions d'euros.

Enfin, le 3 décembre, un millier de terminaux de vente de billets SNCF se trouvent paralysés. Cause déclarée : « un algorithme défectueux ». Un bug dans le même logiciel, Mosaïque, avait déjà fait disjoncter le service au cours de l'installation de sa nouvelle version, le 14 juillet, causant la suspension des réservations SNCF !

■ JK

1. MÉTHODES

Les tests et rien d'autre !



De prime abord, le test semble être quelque chose de simple à réaliser. On teste son application, on découvre un bug, on corrige, on reteste, etc. Avec la multiplication des technologies, des langages, des applications, l'apparition de nouvelles architectures, la génération automatique de code... Tout tend à rendre le test complexe. Que l'on soit un développeur amateur, un étudiant, en SSII, en entreprise, les tests font partie du quotidien qu'il faut savoir maîtriser et utiliser. Le sujet est tellement vaste qu'il est impossible de parler de tout. Essayons tout de même de poser les fondamentaux.

Le test sert à vérifier l'intégrité, le bon fonctionnement, les résultats, le comportement, la stabilité, les performances d'une application. Il sert à détecter les éventuelles erreurs, bugs, dysfonctionnements. Le débogage est l'action servant à corriger un problème de code. Ces deux notions sont liées et indissociables. On peut en ajouter une troisième : l'analyse du code (ex. : code coverage ou couverture du code). Tout cela contribue à la qualité du code, donc à la qualité du projet. Cette qualité est fortement liée à l'architecture définie, à la justesse des spécifications et des exigences, ainsi que des règles de programmation et l'organisation du projet (via une méthode agile de type Extreme Programming par exemple). L'architecture cause un souci particulier. Souvent, on utilise des design patterns. Or, si on s'aperçoit trop tard d'une erreur d'architecture (notamment grâce aux tests), la modification architecturale devient délicate, et l'usage d'anti-pattern (pour corriger une vraie fausse solution d'architecture) ne résout pas tout. C'est pour cela qu'il faut valider l'architecture par un maquettage, afin de tester et de valider celle-ci. Selon Michel Lara d'IBM Software France, 70-75 % des tests seraient encore des tests manuels ! Or, même avec un minimum d'automatisation (scripts, robots, user case), on gagne en efficacité et on optimise son temps pour réaliser d'autres tests. Le test demande une organisation spécifique, de la rigueur, des

outils adéquats. Faute d'argent ou de temps, de nombreux développeurs utilisent les fonctions du débogueur de leur IDE. « En France, nous ne sommes pas pire qu'un autre pays. Chez Microsoft, ils commencent seulement à avoir le plan de tests sur les spécifications. » précise Guy Cartigny, responsable Business Unit "développement" (Compuware).

Organiser, structurer, planifier les tests

Travailler en amont

Le test ne s'improvise pas à la dernière minute. Plus tard le bug est détecté, plus il devient difficile de le corriger. Plus tôt un bug est détecté dans la conception / programmation, moins il coûte cher à corriger. Son impact en est donc réduit.

Pour pouvoir appliquer ce dogme, il faut prendre le test au sérieux et l'implémenter dans le projet en amont. Il peut intervenir dès la fin de la définition des spécifications (= ce que doivent faire l'application, les fonctions, etc.). Lorsque les spécifications sont définies, on débute la phase de conception (architecture, choix des outils, des technologies, définition du processus de développement et des métriques, etc.). Ainsi, les tests unitaires peuvent se définir sur les spécifications, avant le moindre code. Quand le développement fabrique son code, il peut tout de suite vérifier avec le script de tests (tant que le code ne

passé pas le script, le développeur continue son travail). L'autre solution est de pouvoir écrire les scénarios de tests, les scripts de tests simultanément au code. L'avantage est de pouvoir valider au fur et à mesure les spécifications, les objets, classes et composants. On travaille au niveau du code.

Bien entendu, ces tests continus parallèlement au développement ne constituent qu'un premier niveau. Que vous travaillez en équipe, ou sur un projet indépendant / personnel, le test unitaire rendra bien des services...

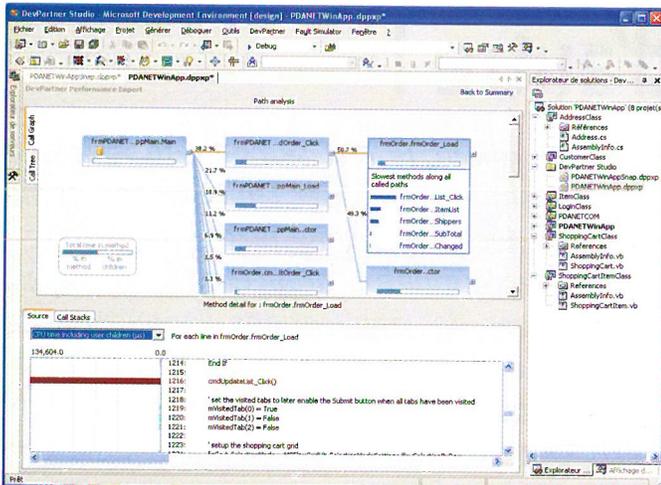
Tracer la vie du bug !

Pour bien traquer l'erreur, le test doit aboutir à son suivi. C'est-à-dire utiliser un système de suivi de bugs / des défauts, bref mettre en place une traçabilité du bug. Chaque bug est répertorié, décrit, avec le niveau de reproductibilité (et la procédure liée), le contexte, le niveau de priorité. Toute correction (la correction comportant le nom du développeur, le module, code modifié) doit être validée par de nouveaux tests. La traçabilité implique aussi une gestion d'historique.

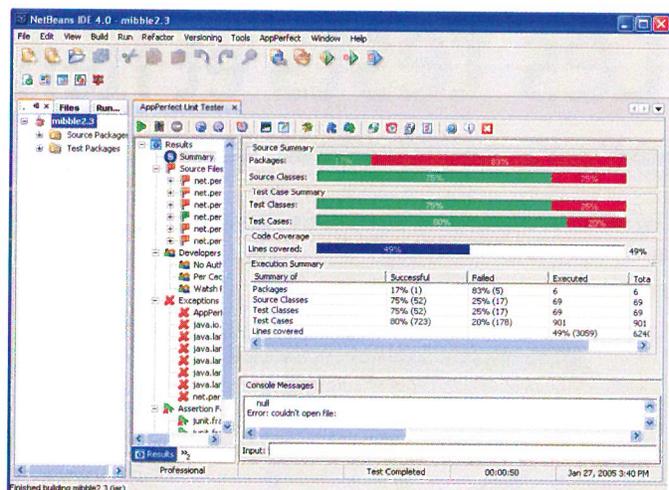
Cycle de tests ?

Le test constitue un cycle à part entière dans le cycle de vie de l'application. Car il ne faut pas parler du test, mais des tests. À chaque étape, à chaque intervenant, le test intervient tôt ou tard, plutôt tôt que tard... « Le problème

1. MÉTHODES



Les résultats de tests sous DevPartner



Les tests sous NetBeans

majeur est que les silos business, de développement et de production, sont étanches. Chez IBM, on unifie les 3 silos en proposant une vision globale des tests, et ce, dès le cahier des charges, jusqu'à la production. » précise Michel Lara (IBM Software France).

Voici une petite liste des tests possibles :

- tests unitaires,
- tests fonctionnels,
- tests manuels,
- tests de non régression,
- tests de couverture du code,
- tests d'intégrité,
- tests d'intégration,
- tests de performances, de montée en charge,
- tests utilisateurs.

bugs plus rapide, des tests plus approfondis, une meilleure qualité. Les tests unitaires (et d'autres basés sur le code) sont aussi nommés tests boîte blanche. Les tests fonctionnels (ou tests boîtes noires) ont pour objectif de vérifier (et de valider) les spécifications de l'application, tout en permettant de vérifier les performances, la mémoire, les ressources (attention aux fuites mémoires et à la mauvaise gestion de la mémoire et des ressources)... Il existe deux grands types de tests fonctionnels : tests nominaux (respect des spécifications dans le comportement normal), tests aux limites (on pousse l'application aux limites).

On pourrait aussi parler de tests de stress. Très intéressant, notamment dans une application web, pour tester la solidité et les performances d'un site, d'un composant. On simule les actions et les utilisateurs. Il s'intègre souvent dans les tests de charge (ex. : OpenLoad). Le test de stress peut aussi concerner l'application « normal » comme dans les données et la base de données. Les tests d'intégration seront très utiles en équipe. On intègre les différents modules, puis on applique un cycle de tests : unitaires, robots, fonctionnels, etc. Dans les projets offshore, c'est indispensable.

Les applications à risque

Les applications web

Cette catégorie d'application est l'une des plus difficile à tester, même si on peut y appliquer, peu ou prou, les méthodes classiques. Une application web dynamique (ASP, JSP...) sera d'autant plus difficile à tester car il faut travailler au niveau client et serveur et directement dans les conteneurs serveurs (ex. : les servlets, EJB

Une cellule de qualité / tests et use case

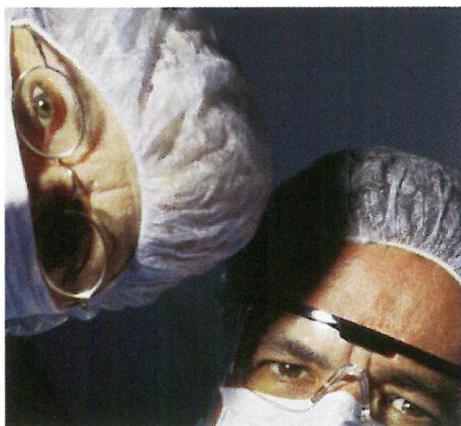
Sans aller jusqu'à la cellule qualité / test qu'une entreprise peut mettre en place pour définir, suivre et appliquer les différents tests, avec pour corollaire la qualité du projet et un rôle transversal par rapport à l'ensemble des intervenants (développeur, chef de projets, architecte, utilisateurs, direction, production, administrateur, etc.), on peut chez soi, dans une PME, une SSII mettre en place une structure identique : choisir les outils et définir l'ensemble des tests, créer les scénarios de tests (par rapport aux cas d'utilisation du projet – les use case). Il faut recréer au maximum les environnements d'utilisateurs possibles. Par exemple, pour un site web prendre en compte les différentes versions de systèmes et de navigateurs (Safari, Firefox, IE sous Windows XP / 98 / 2000, Linux, MacOS X...). Pour les applications de bureau, même chose. Les outils de virtualisation (ex. : vmware) permettent une très grande souplesse en virtualisant les systèmes cibles.

Vous DEVEZ appliquer l'ensemble de ces tests. Dans le cadre d'un gros projet, il est préférable de dédier une personne à l'établissement des tests, des protocoles, des procédures et d'en assurer le suivi et l'historique. Un développeur indépendant, étudiant, amateur, applique ces tests à une échelle moindre. Il peut le faire en manuel ou utiliser des outils open source afin de réduire le coût.

En XP, on remonte très en amont les tests, notamment les tests unitaires. On peut écrire les tests soit avant le codage des fonctions, soit en même temps. Mais, tous les tests ne peuvent pas être écrits avant. Les avantages sont multiples :

- on définit mieux les cas d'utilisation et on affine si besoin est les spécifications
- on aide à créer les entrées / sorties nécessaires

Disposer des scénarios de tests et d'utilisation vous permet de mieux coder le module. Tant que le code ne passe pas les scénarios, inutile de passer au suivant. Dans cette configuration, on est dans le développement piloté par le test (TDD). Basiquement, un test = une classe. Grâce à la méthode TDD, on aboutit à une traque de



Echange et stockage de messages, plus...



Développement d'Applications Composites, plus...



Orchestration des Processus Métiers, plus...



Observation de l'activité métier (BAM)

Intégrer vos processus d'entreprise. Enfin plus simple qu'une neurochirurgie !

Les innovations d'Ensemble vous permettent de réaliser facilement et beaucoup plus rapidement l'intégration des processus de votre entreprise. Ensemble est la première plateforme à offrir, au sein d'un même produit, toutes les technologies d'intégration et de développement, dont vous avez besoin pour mener à bien tout type de projet d'intégration.

La **Plateforme d'Intégration Universelle** Ensemble est le résultat d'une fusion innovante d'un serveur d'intégration, d'un serveur de données, d'un serveur d'applications, et d'un logiciel de développement de portails. Avec son environnement de développement et d'administration unifié, Ensemble réduira drastiquement vos délais de réalisation.

Et comme nous avons une pleine confiance dans notre solution, nous vous offrons une **garantie de remboursement** : jusqu'à un an après l'achat de votre licence Ensemble, nous vous remboursons 100% de son montant, si vous étiez insatisfaits, pour quelque raison que ce soit*.

InterSystems, votre partenaire technologique avec plus de 25 ans d'innovations réussies et N° 1 dans le monde de la Santé.

Rendez-vous sur le Forum de l'Intégration au CNIT/Paris les 23 & 24 novembre prochains au stand 25.
Conférence sur le salon le 23 novembre de 16h30 à 18h30 – Espace Darwin.
Plus d'infos sur : www.InterSystems.fr/forum

InterSystems
ENSEMBLE

Demandez votre Projet Pilote GRATUIT sur www.InterSystems.fr/Integration

* Consultez notre site web pour plus d'information.

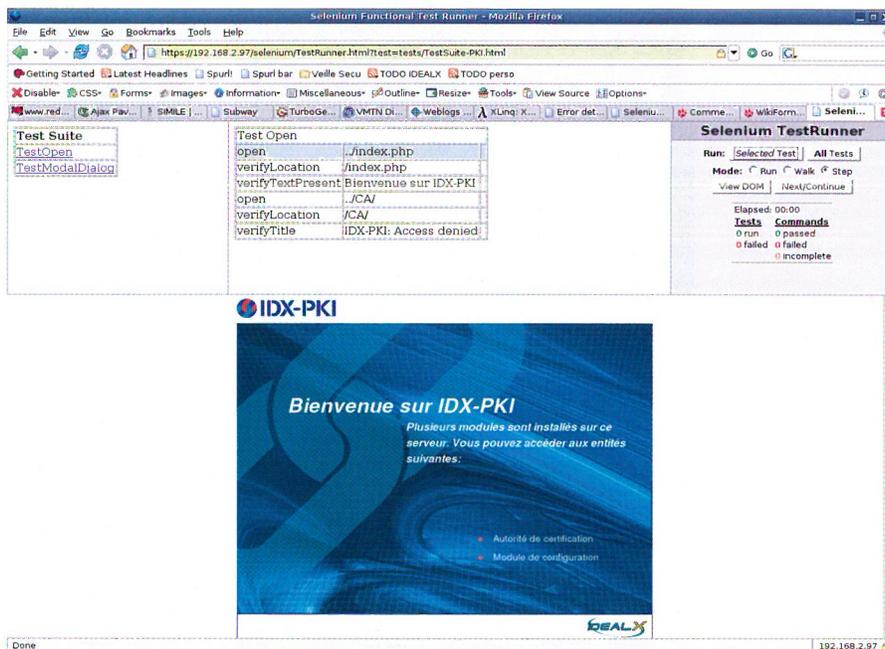
© 2005 InterSystems Corporation. Tous droits réservés. InterSystems Ensemble est une marque déposée d'InterSystems Corporation.

1. MÉTHODES

pour les applications entreprise, etc.). Surtout, on utilise différentes technologies nécessitant souvent des méthodes et outils de tests différents : Flash, XML, mapping, servlet, portlet, protocoles de sécurité, HTML, XHTML, CSS, pages dynamiques, modèle MVC pour la conception, framework, la base de données, le serveur web, etc. Plus on superpose les couches, les composants, plus on complexifie le cycle de qualité. Cependant, grâce au découplage des couches, on simplifie le développement et facilite théoriquement le travail du testeur. Mais l'emploi de framework introduit des boîtes noires dans son application, et *de facto*, on perd le contrôle. Heureusement pour vous, de nombreux outils de tests commerciaux et open source, dédiés au Web, sont disponibles. De plus, il n'est pas toujours aisé d'automatiser les tests web (pas d'automatisation de l'authentification). Cependant, les outils existants permettent d'analyser le code (sur la sécurité, le trafic http, les pages dynamiques, la montée en charge, etc.). Vous aurez le choix entre des frameworks de tests (Cactus, JwebTest, htmlUnit, httpUnit...) et les outils autonomes. Dans un autre domaine, les applications multi-threadées et parallélisées nécessitent une rigueur d'analyse et de tests de plus haut niveau.

Services Web, SOA...

Avec l'apparition et la mise en place des architectures services (SOA) utilisant les standards XML et des services web, le test se complexifie encore un peu plus. L'offre d'outils de test sur ces domaines est encore limitée mais s'étoffe.



L'outil de tests de IdealX

Par exemple, il faut pouvoir vérifier la conformité par rapport aux spécifications des web services, tester les WSDL, le protocole SOAP, etc. Parasoft dispose de SOA Test, pour aider à tester et déboguer en environnement SOA. L'éditeur propose en fait de suivre l'ensemble de l'architecture, du service web à la plateforme d'architecture. Quotium propose dans son outil Qtest des fonctions pour les services web. Webpartner propose un environnement de tests, orienté Web (Webpartner Test), prenant en charge XML et SOAP. L'une des difficultés est

de pouvoir tester un environnement applicatif distribué et interagissant avec d'autres services, d'autres applications internes ou externes. XML cause un souci, lié à sa complexité verbale et à la génération de plus en plus automatique de son verbiage dans les applications de création, de développement. Or, comment valider un schéma, un document, une transaction XML ? Comment valider et analyser la transformation ? Si les outils de tests commencent à prendre en compte ce problème, ils sont loin du niveau des outils « traditionnels ». Une des solutions sur les schémas est de pouvoir créer des jeux de tests sur des documents formés et validés afin de les appliquer aux tests schémas ! Pour la transformation, passez éventuellement par un outil de type XSLUnit ou XMLUnit. Altova propose peu ou prou la même chose avec XML Spy.

La génération de code : un autre problème ?

Le développement s'appuie de plus en plus sur les frameworks, les modèles (UML principalement), les générateurs de code. Si d'un côté, on simplifie le codage, de l'autre, on rend parfois le test et le débogage plus délicat, plus complexe. Et le risque est de se retrouver avec des boîtes noires, impossibles (ou quasiment) à déboguer après les tests. Il faut donc être vigilant sur la qualité des frameworks, modèles, générateurs.

■ François Tonic

Les bonnes pratiques selon Parasoft

Développement

- 1 Développement défensif : on anticipe les erreurs, les failles.
- 2 Revue du code : développeur et architecte se rencontrent pour discuter de la manière d'écrire le code.
- 3 Code standard : on utilise les spécifications et possibilités du langage utilisé. Rien d'autre.
- 4 Test unitaire : on commence très tôt à tester les morceaux de code.
- 5 Construire les tests (boîte blanche)
- 6 Tests fonctionnels
- 7 Analyse de couverture du code
- 8 Tests de régression
- 9 Tests d'intégration

Tests

- Comprendre l'architecture du projet avant de démarrer les tests
- Anticiper les abus potentiels et vérifier comment le logiciel répond
- Définir clairement la procédure de reproduction d'une erreur détectée
- Aider l'équipe à prévenir les erreurs

Maîtriser les Tests Unitaires

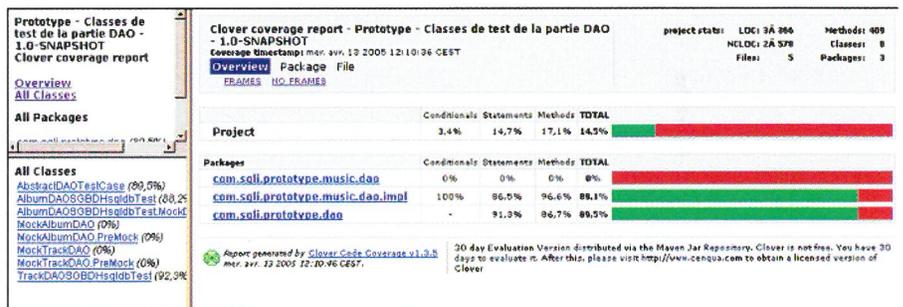
« Jamais aussi peu de lignes de code ont autant fait progresser le domaine du développement logiciel. »¹ *Martin Fowler. Les tests sont certainement une des phases les plus critiques du cycle de développement du logiciel. La mise en place d'une politique de tests unitaires au sein d'une méthodologie de développement devient un choix stratégique dans l'objectif d'amélioration continue de la qualité logicielle.*

Les tests unitaires sont liés aux réflexions plus globales autour du Design applicatif. Ils émanent également de méthodologies de développement, dites *agiles*, par opposition aux méthodologies classiques dites prédictives. Les méthodologies *agiles* modifient la manière de mener la réalisation de projets informatiques : elles privilégient un cycle de développement itératif et incrémental, et proposent d'assainir la phase de codage en appliquant un ensemble de pratiques, parmi elles : les tests unitaires. Ceux-ci bousculent ainsi quelques (mauvaises !) habitudes de développement, mais leurs enjeux sont tels qu'ils deviennent un élément central dans la réussite d'un projet informatique.

Test first, les enjeux du Test Driven Development

Le premier objectif des tests unitaires est de détecter un maximum de bugs très en amont de la mise en production, en l'occurrence, dès la phase de développement. Le *Test Driven Development*² (développement guidé par le test), pratique « pivot » des méthodologies agiles, révolutionne le processus de développement :

- Le développeur écrit le résultat attendu (l'exigence) de la fonction à tester, au sein de la classe de test.
- Puis il écrit la classe fonctionnelle en provoquant une erreur, ce qui permet de vérifier que la classe de test détecte bien les erreurs.
- Il finalise ensuite l'algorithme de la classe à tester.
- Il vérifie enfin que le test ne détecte plus d'erreur, il s'assure alors que le traitement développé vérifie l'exigence. Il revient à l'étape précédente si le test est en erreur. Si nécessaire, le développeur complète les tests



Reporting sur les couvertures de tests unitaires avec l'outil Clover

de la fonctionnalité avec les tests aux cas limites manquants.

Tester et développer devient une seule et même activité ! Le fait d'intégrer les tests au cœur du développement logiciel permet également de formaliser avec plus de précisions les exigences, les tests unitaires concrétisent en effet la compréhension fonctionnelle du développeur, ainsi que les cas limites envisagés. A noter qu'il est possible d'utiliser, pour vérifier la pertinence du test unitaire, des outils de couverture de code, citons par exemple l'outil Clover³ (possédant un plug-in maven⁴, voir copie d'écran ci-dessus).

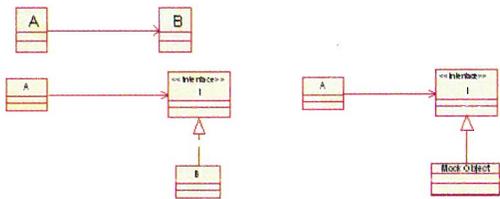
Les frameworks de tests

De nombreux frameworks dédiés aux tests unitaires ont vu le jour ces dernières années. JUnit⁵ est l'un d'entre eux, écrit par Kent Beck (un des créateurs de XP) et Erich Gamma (un des quatre auteurs d'un ouvrage de référence sur les *Design Patterns*⁵), il est considéré comme le framework de test de référence du monde J2EE. Il est basé sur quelques principes généraux :

- Une classe de test par classe développée. Chaque classe de test doit hériter d'une classe du framework : *TestCase*.
- Il est possible d'enchaîner des tests en utilisant la classe du framework : *TestSuite*.
- La vérification des tests est effectuée à partir de méthodes d'assertions proposées par le framework : *assertEquals(message, valeur attendue, valeur calculée)*, *assertNotNull(message, valeur calculée)*, *assertTrue(message, valeur calculée)*, etc.
- Une méthode de test (au moins) par méthode à tester. Les méthodes de test doivent être préfixées par « test » et indépendantes entre elles.
- Les données créées doivent toujours être réinitialisées pour chaque méthode testée. La méthode *setUp()* est appelée avant chaque appel d'une méthode de test, elle peut contenir les initialisations d'objets.
- La méthode *tearDown()* est appelée en fin de chaque méthode de test, elle peut contenir par exemple la fermeture des objets ouverts dans la méthode *setUp()*.

```
SHAPE \* MERGEFORMAT public class TestMonDaoImpl extends TestCase
{
// interface du DAO a tester
MonDao monDao;
```

1. MÉTHODES



Les interfaces, l'inversion des dépendances et les Mock Objects

de I et non plus de l'implémentation B (voir figure ci-dessus).

Le deuxième concept du pattern d'inversion de contrôle est l'injection de dépendance¹¹, décrit par Martin Fowler (principe sous-jacent aux conteneurs légers tels que Springframework¹² ou Hivemind¹³). Il s'agit d'injecter les dépendances d'un composant plutôt qu'il ne les instancie lui-même. Les objets sont dépendants des interfaces des classes qu'ils utilisent et non plus des implémentations, et celles-ci leur sont « injectées ». L'implémentation, si tant est qu'elle respecte cette interface, peut varier et être fournie par exemple uniquement à l'exécution. Ceci garantit ainsi un faible couplage entre composants, notamment entre composants de couches applicatives distinctes, le faible couplage étant un des buts à atteindre du Design. Relativement à notre sujet, en appliquant ce principe d'injection de dépendances, il devient plus aisé de tester un composant de manière réellement isolée. Les classes peuvent en effet être testées unitairement lorsque des « Mock Objects » leur sont fournis comme implémentations des objets dont elles dépendent. Les Mock Objects sont des objets spécifiquement conçus pour les tests : il s'agit d'implémentations postiches d'interfaces utilisées par la classe fonctionnelle à tester (voir figure ci-dessus). Il existe plusieurs frameworks Mock Objects, entre autres : MockMaker¹⁴, MockCreator¹⁵, EasyMock¹⁶, Jmock¹⁷.

Tests unitaires et Intégration continue : pour un développement dans la sérénité

Le lancement du test et l'analyse du résultat du test sont automatiques, donc intégrables dans un processus d'intégration continue. Ce type de processus permet, conjointement à l'écriture de tests unitaires, de construire un projet de manière incrémentale : chaque jour, il est possible de vérifier que l'application compile, est distribuable, déployable et fonctionne sur la base des cas de tests unitaires écrits. En lan-

çant régulièrement l'intégralité des tests unitaires, l'intégration continue donne ainsi un feedback rapide sur les éventuelles régressions et permet de tester régulièrement un maximum de livrables (scripts sql, scripts de construction). L'intégration continue peut être implémentée simplement à partir d'un script cron type ant¹⁸, encapsulant toutes les phases de l'in-

tégration sous forme de « target » et utilisant des « tasks » spécialisées (cvs, junit, junitreport). Il peut être préférable d'utiliser un outil dédié à l'intégration continue, comme CruiseControl¹⁹, Luntbuild²⁰ ou Anthill²¹, associé à un outil de gestion de projet type maven²². Les tests unitaires responsabilisent l'équipe de développement : ils garantissent que les développeurs font bien des tests (puisqu'ils sont écrits !). Ils permettent fortement de conserver au logiciel sa capacité à répondre au changement et donnent confiance dans l'avancement du projet. Ils permettent d'aborder des phases

de remaniement de code avec sérénité. Lors d'une évolution ou d'une correction, le fait de rejouer l'ensemble des tests unitaires permet d'identifier immédiatement un éventuel effet de bord et de découvrir au plus tôt les anomalies. Plus une anomalie est détectée tôt, moins sa résolution coûtera ! Le développement piloté par les tests favorise ainsi le refactoring de code, principe sur lequel s'appuie le *design dit évolutif*, l'un des enjeux des méthodologies agiles. Celui-ci consiste à pouvoir faire évoluer et modifier potentiellement l'architecture applicative à chaque itération, à minimiser l'anticipation d'éventuels besoins futurs qui pourraient complexifier inutilement le Design... Nous retrouvons là encore un des corollaires d'XP, le principe « YAGNI » (You Ain't Gonna Need It), sous-jacent à une valeur essentielle dans XP : celle de toujours préférer la simplicité et de privilégier une approche du Design pragmatique.



■ Agnès CREPET, Sqli

Références :

- Article de Ron Jeffries sur l'intérêt des tests unitaires : http://www.xprogramming.com/publications/software_testing.htm
- <http://www.extremeprogramming.org/rules/unittests.html>
- Tests unitaires en Java, les tests au cœur du développement, Johannes Link, Dunod, 2003.

¹ « Never in the field of software development was so much owed by so many to so few lines of code. » Martin Fowler, à propos du framework de tests unitaires JUnit

² Egalement appelé Test Driven Design

³ <http://www.cenqua.com/clover/>

⁴ <http://maven.apache.org/reference/plugins/clover/>

⁵ <http://www.junit.org>

⁶ « Design Patterns. Elements of Reusable Object-Oriented Software » - E. Gamma, R. Helm, R. Johnson, J. Vlissides - 1994, Addison Wesley

⁷ Un DAO, « Data Access Object » en anglais, est une classe responsable de la persistance (en base de données, fichiers XML, ...) des objets

⁸ <http://dbunit.sourceforge.net/>

⁹ <http://www.beust.com/testng>

¹⁰ Avec J2SE 1.5, mais également compatible avec J2SE 1.4 avec des annotations à la xDoclet

¹¹ L'injection de dépendance : <http://martinfowler.com/articles/injection.html>

¹² <http://www.springframework.org>

¹³ <http://jakarta.apache.org/hivemind/>

¹⁴ <http://mockmaker.sourceforge.net/>

¹⁵ <http://mockcreator.sourceforge.net/>

¹⁶ <http://www.easymock.org/>

¹⁷ <http://jmock.org/>

¹⁸ <http://ant.apache.org>

¹⁹ <http://cruisecontrol.sourceforge.net>

²⁰ <http://www.pmease.com/luntbuild/>

²¹ <http://www.urbanocode.com/projects/anthill/default.jsp>

²² <http://maven.apache.org>, depuis peu maven propose également sa propre version en beta test : Continuum

Ne pas sous-estimer les tests de non régression

On entend beaucoup parler d'usines à logiciels, de tests unitaires automatisés, et l'on y voit quelque chose d'extraordinaire. L'idée d'automatiser les tests n'est pas nouvelle, et il y a longtemps que la phase des tests de non régression est automatisée dans la plupart des équipes de développement. Jusqu'à l'arrivée des méthodes agiles, ces tests constituaient la part la plus importante de la vérification du bon fonctionnement des applications.

Ces tests garantissent une continuité dans les résultats au cours des livraisons successives. Deux approches sont généralement considérées. La première consiste, à chaque anomalie détectée, à créer un test permettant de la mettre en évidence, et de l'ajouter à la batterie des tests de non régression. Cette approche présuppose que la portion de code correctif, ainsi que son voisinage, a une probabilité assez forte d'être génératrice d'autres anomalies dans le futur. En effet, un code correctif a plus tendance à déroger aux diverses règles et conventions de codage, et à être, de ce fait, moins compréhensible que le code dans lequel il s'insère.

La seconde approche, plus répandue, consiste à s'assurer de l'intégrité de l'application après toute intervention sur son code. Cette approche est très suivie dans le cas des applications lourdes, ayant un fort historique, ou, de façon plus générale, dans les projets écrits en C++ pour lesquels la mise en place de tests unitaires n'est pas toujours facile. On peut rencontrer une troisième approche : la comparaison des résultats produits par la version de test avec ceux produits par une autre application. Cette approche est certainement une des meilleures, et l'on peut regretter qu'elle soit rarement mise en œuvre.

Quand exécuter les tests de non régression

La phase des tests de non régression suppose que l'on dispose d'une version aboutie de l'application. Un plan de test classique comprend les tests unitaires, les tests d'intégration, les tests système, les tests de performance et de charge, et, pour finir, les tests de non régression, suivis des bêta-tests. Les tests de non

régression arrivent tardivement dans le cycle de développement. Qu'il s'agisse de corrections, d'ajouts de fonctionnalité ou d'optimisations, ceux-ci sont considérés comme terminés. L'objectif est bien de s'assurer que tous ces développements n'ont pas introduit de régressions par rapport à la version précédente. Naturellement, comme dans toute phase de test, il faut prévoir plusieurs itérations. La première itération révèle des anomalies dont les corrections sont validées par la seconde. Certaines corrections peuvent engendrer de nouvelles régressions comme dans le cas où la correction révèle une autre anomalie masquée jusque là. Les tests de non régression sont qualifiés de test de boîte noire : on teste l'application de l'extérieur.

Déroulement des tests de non régression

On retrouve toujours les mêmes étapes. A ces étapes s'ajoute la phase de correction des régressions détectées.

Définition du jeu d'entrées

Les tests de non régression sont par nature des tests massifs. Idéalement, on souhaiterait tester le comportement de l'application sur toutes les valeurs possibles que peuvent prendre les entrées. En pratique, on établit au cours du temps, différents jeux de tests plus ou moins représentatifs du périmètre de l'application. Lorsque cela est possible, on utilise un automate pour générer les jeux de test.

Exécution de l'application

Si l'application s'intègre au sein d'un système complexe, l'usage de bouchons ou autres mock objects peut se révéler très utile.

Collecte des résultats

Le format "fichier plat" reste toujours le plus adapté à des comparaisons.

Comparaison avec la référence

La qualité du rapport de test repose en grande partie sur l'outil de comparaison des résultats. On doit être en mesure de déterminer très précisément la nature des écarts. L'usage d'une base de données dédiée comme outil de comparaison peut être idéal sur des volumes importants.

Définition de la référence

La définition de la référence est un autre aspect important des tests de non régression. Deux grandes approches. La première, la plus contraignante, consiste à stocker quelque part des jeux de tests, ainsi que les résultats attendus. La seconde consiste à ne stocker que les jeux de tests, et à exécuter, d'une part, la version de référence de l'application (la version en production ou la dernière version ayant passé avec succès les tests) et, d'autre part, la version de test. Cette approche a l'avantage de ne pas introduire de biais éventuels dus à des environnements d'exécution ou à des configurations évoluant au cours du temps.

Automatisation

En utilisant des scripts divers, beaucoup d'équipes de développement ont en partie automatisé cette phase. Beaucoup de tests unitaires sont d'ailleurs écrits et utilisés comme tests de non régression. Au lieu d'écrire de véritables tests unitaires validant la fonction à tester, le test unitaire repose sur la lecture d'un fichier de données d'entrées, l'exécution de la fonction sur ces entrées et la comparaison du

1. MÉTHODES

résultat à une sortie de référence associée à ces entrées.

Du fait de la nature massive des tests de non régression, leur intégration à l'usine à logiciel, après le build quotidien par exemple, ne peut être que partielle. Un délai de plusieurs heures entre le lancement du test et la production du rapport n'est pas exceptionnel. Mais comme l'on souhaite détecter les anomalies très rapidement, on peut tout de même effectuer un test portant sur un périmètre restreint judicieusement constitué.

Une fois le rapport de test établi, reste à réaliser le "dépilage", dont le but est de cataloguer chaque écart. Il s'agit d'une tâche fastidieuse pouvant être automatisée en créant, pour chaque anomalie, une méthode permettant de ranger un écart identifié dans une catégorie

connue. A l'issue de la phase de comparaison, on invoque toutes les méthodes sur chaque écart de façon à n'avoïr à "dépiler" que les écarts inconnus.

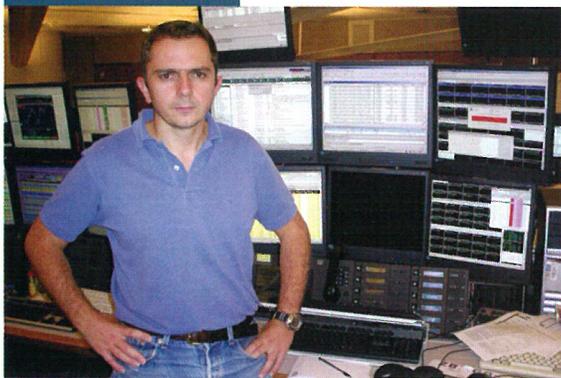
Les tests de non régression seuls ne sont pas suffisants.

Incontournables avec la livraison d'une application, ces tests ne sont pas le vaccin miracle contre les anomalies : comme tous les autres tests, leur but est de déceler des anomalies et non de valider que l'application en est exempte. Ils ne sont d'aucune utilité pour valider les nouvelles fonctionnalités ou les correctifs. C'est ainsi que l'on peut arriver à la situation (authentique) suivante : une fonctionnalité est restée en production boguée pendant plusieurs mois, car non ou mal réfé-

rencée lors de son introduction ; et une fois corrigée, les tests de non régression ont échoué et l'équipe de développement a dû expliquer la régression. Gérer des "écarts corrects" est désagréable et pénible, mais il y a plus grave : ces écarts peuvent masquer de nouvelles anomalies.

Bien souvent, on constate que la taille de la batterie des tests de non régression (et donc leur durée et leur charge sur le planning) est d'autant plus grande que la couverture du code par les tests unitaires est faible. Il en va de même pour la criticité : en l'absence de tests unitaires (et *a fortiori* de l'absence des autres types de tests) ils deviennent le seul rempart contre l'apparition d'anomalies en production derrière lequel les équipes peuvent s'abriter.

INTERVIEW



Un robot pour améliorer les tests

L'équipe R&D de la Société Générale banque d'investissement (SGIB) a mis en place un "robot de développement" maison. La cellule R&D a la charge du développement d'une bibliothèque de calcul de prix utilisée par les salles de marchés sur produits exotiques de taux, de change et de crédit. Vincent Mahon, l'expert technique de l'équipe qui a mis en place le robot nous en parle.

Quand avez-vous mis en place votre robot ?

En 1994, nous ne disposions pas d'outils de test. Les développeurs validaient leurs modifications à la main. Évidemment, ce n'était pas satisfaisant : il fallait ne pas oublier de tester, connaître l'impact des modifications pour savoir quoi tester (effet battement d'aile du papillon), ne pas se tromper en testant, et cela prenait du temps. Au début, le robot a été écrit en Delphi, comme la bibliothèque de calcul. Aujourd'hui celle-ci est écrite en .NET (C#) et le robot en Java.

Quelles sont les principales fonctions de ce robot ?

Le logiciel cœur du robot est conçu pour gérer une file d'attente de tâches avec

gestion des priorités, communiquer avec ses utilisateurs (par email), pour lancer des tâches et pour utiliser notre système de gestion de sources. Il est soumis à une forte contrainte de robustesse. Ce logiciel cœur exécute des tâches qui vont elles mêmes lancer des programmes externes. Par exemple, lorsqu'elle est lancée sur le tronc commun, la tâche principale de test effectue les opérations suivantes : obtenir les sources à tester (temps négligeable), poser un label sur les fichiers sources utilisés, qui permettra d'identifier formellement ce qui a été testé (en 10 minutes), compiler notre programme (en 6 minutes), lancer le programme qui fait tourner les tests unitaires en parallélisant sur un parc de machines (en 14 minutes), lancer



Des outils pour bâtir.
Des outils pour affiner.
Des outils pour transformer.

Vous construisez des applications logicielles qui affecteront la façon dont le monde travaille, joue et agit. Nous sommes là pour vous accompagner, avec des produits qui aident les applications à mieux tourner et plus rapidement. Compilateurs. Tuners. Outils multi-filaires. Debogueurs. Et beaucoup plus. Nos produits vous aident à intégrer de l'innovation logiciel avec les technologies propres aux plates-formes Intel—depuis l'idée de base jusqu'à l'optimisation.

Intel[®] software network

Pour tout savoir, allez sur www.intel.com/software/products

intel[®]

1. MÉTHODES

le programme qui fait tourner les tests de non régression en parallélisant sur un parc de machines (en 15 minutes), lancer le programme qui fait tourner des tests d'interaction avec d'autres programmes en parallélisant sur un parc de machines (en 1h30), lancer des programmes d'analyse statique du code (en 3 minutes), identifier la liste complète des différences par rapport au label précédent et contacter les développeurs concernés pour leur demander la doc utilisateur à ajouter au site web, prévenir les développeurs s'ils ont intégré du code dans une version qui ne passe pas les tests (5 minutes), documenter le label posé sur les sources afin qu'il soit simple de savoir si ce label est validé (temps négligeable), et enfin, prévenir tous les développeurs de l'état du tronc commun.

Le robot consulte à intervalles réguliers le système de gestion de sources afin de déterminer si quelque chose a changé depuis les derniers tests sur le tronc commun. Dans l'affirmative, il relance une validation afin que la dernière version du tronc commun soit au plus vite dans un état connu, permettant à un développeur d'utiliser le label posé comme point de départ d'une branche. On peut

spécifier au robot, différents niveaux de validation (par exemple seulement les tests rapides). Le robot s'occupe également, sur demande, de déployer les versions du logiciel après validation, d'ajouter de la documentation sur le site web.

Au quotidien, comment le développeur utilise-t-il le robot ?

Le développeur demande au robot de lancer des tests en lui envoyant un courrier électronique qui respecte une grammaire précise. Typiquement, le développeur demande au robot de valider ses modifications lorsqu'il a atteint un point stable où il pense et souhaite que les tests passent. Le développeur doit aussi obligatoirement faire valider ses modifications par le robot avant de les merger vers le tronc commun de l'arborescence des versions. Le robot répond par mail après exécution de la tâche.

A quelle fréquence, le robot est-il sollicité ?

Nous sommes aujourd'hui quinze développeurs. Chacun travaille sur son chantier et donc sur sa branche. En temps normal le robot reçoit une quinzaine de

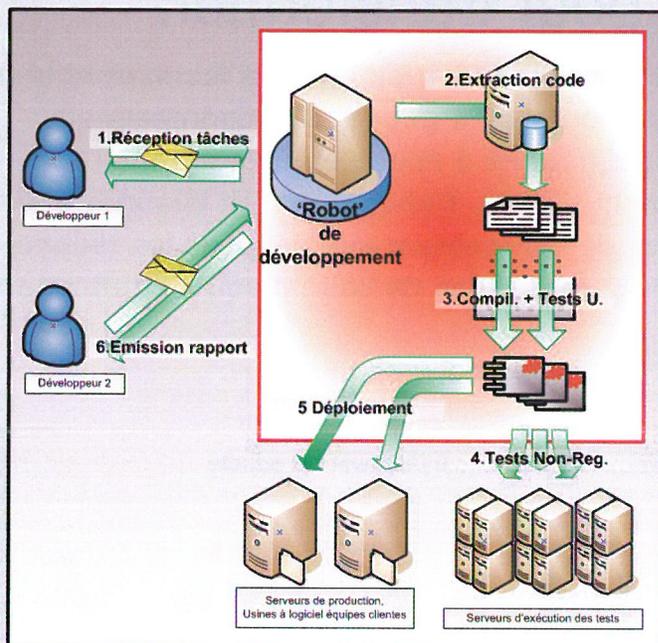
demandes de tests par jour. Un test prend environ 2h30 tout compris quand il va jusqu'au bout. Mais si par exemple les tests unitaires ne passent pas, les tests de non régression ne sont pas exécutés.

Nous avons trois robots qui ont chacun un parc d'une douzaine de machines. Par ailleurs, le robot scrute toutes les dix minutes les sources du tronc commun pour voir si quelque chose a changé. Dans l'affirmative il pose un label et teste. Nous avons environ en moyenne trois labels de ce type par jour.

Quelles fonctionnalités manquent ?

Le problème principal est la durée des tests et nous comptons ajouter encore des tests. Les 2h30 posent principalement le problème de l'engorgement de la file d'attente du robot. Ce délai est peu compressible, à cause de la taille de l'atome de parallélisation (dans la partie listée ci-dessus qui prend 1h30, nous dépendons d'autres équipes). Par ailleurs, la parallélisation sur parc hétérogène est difficile à optimiser. L'utilisation de trois robots pour diminuer le temps d'attente ne fait que diminuer le problème : il arrive que le robot numéro 1 soit surchargé, alors que le numéro 2 est inoccupé. Il y aurait à gagner de ce côté là. La consultation que fait régulièrement le robot des sources sur le tronc commun pourrait être remplacée par un système d'événements. Les API modernes permettent de rêver qu'un jour nous serons capables de déterminer automatiquement quels tests il faut lancer lorsque l'on modifie du code. Mais il y a du temps à passer sur le sujet ! Par ailleurs, nous n'avons pas pour l'instant de tests de performances, la parallélisation multi-thread et multi-machine sur parc hétérogène complique les mesures.

Yann A.Oudghiri est consultant chez NEOXIA, intégrateur de systèmes spécialisé dans les nouvelles technologies de l'information.



Pour les initiés.

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

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

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

John Appleyard, Polyhedron*

Performances

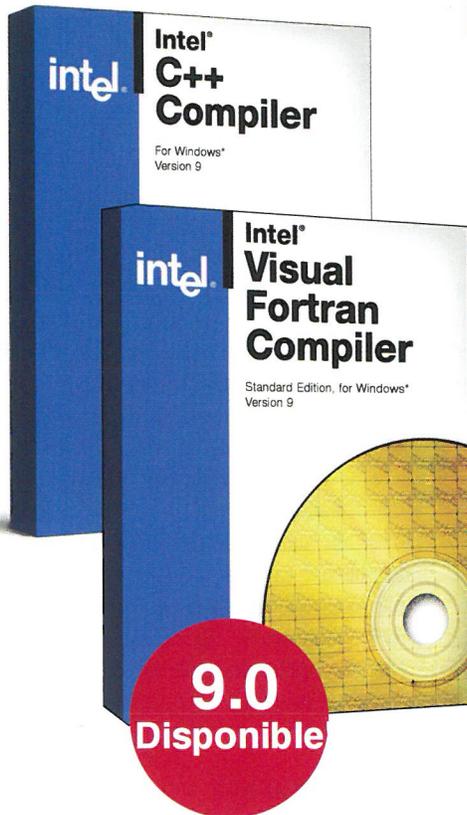
Les outils de développement Intel® dotent les plus grands développeurs d'une connaissance d'initié aux plates-formes informatiques hautes performances. Leur quête vers des niveaux de performances exceptionnelles est simplifiée grâce à des caractéristiques comme l'auto-parallélisation, OpenMP, des mises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

Compatibilité

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

Support

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



Intel® Thread Checker 2.1

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

Intel® Integrated Performance Primitives Library 4.1

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

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

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

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

L'art subtil du test

Bugs de conception, bugs de codage, bugs système, la panoplie est vaste. Elle définit un contexte dont les implications sont économiques, puisqu'elles déterminent la dérive des délais et des coûts. Une bonne gestion de la prévention, puis de l'éradication des bugs favorisera l'harmonisation et la rentabilité des projets.

Pour comprendre la complexité de l'art de tester, reportons-nous à l'exemple donné par Glenford J. Myers dans son ouvrage classique, *The art of software testing*. Il s'agit de tester trois valeurs données, correspondant à la longueur des côtés d'un triangle, et de déterminer si le triangle est scalène (quelconque), isocèle ou équilatéral. La plupart des programmeurs échouent lamentablement à percevoir la nécessité de tests tels que :

- un des côtés est-il égal à zéro ?
- un des côtés a-t-il une valeur négative ?
- la somme de deux côtés est-elle inférieure à la valeur du troisième ?
- la somme de deux côtés est-elle égale à la valeur du troisième ?

Sur les quatorze tests nécessaires, les meilleurs programmeurs en trouvent entre sept et huit. On imagine par conséquent la difficulté à tester un logiciel conséquent de façon simplement raisonnable. On ne peut cependant pas s'en dispenser, sauf à le payer fort cher lorsque l'application entame son cycle de vie normal.

Définir les bons et mauvais tests

Pour beaucoup de chefs de projet, un test est ce qui détermine l'absence d'erreurs. Par conséquent, un bon test ne mettra pas en évidence les failles d'un programme. En effet, un bon programme fonctionne sans surprises. Bons tests et bons programmes, de ce point de vue, participent de la même illusion.

Psychologiquement, ils évitent les cas extrêmes et se fauillent au travers des cas déjà testés, qui correspondent à l'utilisation normale des éléments testés. L'utilisateur étant par nature imprévisible, il découvrira à ses dépens ce que les tests n'auront pas mis à jour. Dans le cadre de formules financières, les valeurs de sorties d'argent seront précédées du signe moins, du simple fait que c'est ainsi qu'on manipule une calculatrice financière. Ce qui sort de la poche est négatif, ce qui entre dedans est positif. Cette erreur repose sur une logique totale. Le testeur optimiste n'y pense pas. Son programme marche tel qu'il l'a écrit, et cela lui suffit.

Quand un bon test est un mauvais test

Au contraire, la bonne attitude à l'art de tester consiste à se comporter en chasseur. Un bon test capture des bugs. Il réalise alors sa vocation. Toute erreur constitue un gibier. Un test intelligent utilise des stratégies concourantes, afin d'organiser des battues efficaces. Aucun bug n'est supposé y échapper. Il en découle que ce changement de mentalité considère tout bug capturé comme une victoire du testeur. Plus le testeur est compétent et plus il s'avère l'ennemi du développeur. Comme l'affirme le dalaï-lama, "ton pire ennemi est ton meilleur ami". Il t'aide à progresser, pourrait-on ajouter. Il en découle que, de façon évidente, le testeur ne doit jamais être l'auteur du programme. On ne peut être à la fois le parent des bugs et leur prédateur. L'inconscient s'y oppose. Ces deux rôles sont antithétiques. Ils s'excluent mutuellement. Au pire, dans le cas d'un développeur unique, c'est à l'utilisateur final de maltraiter le programme, en anticipant les erreurs de manipulation qui ne manqueront pas de surgir dans la vie réelle. On observera que face à une équipe de tests indépendante, le développeur aura tendance à anticiper les cas qu'il considèrerait

CHECK LIST

La prévention des bugs et erreurs de conception débute en même temps que le projet, avec l'écriture du cahier des charges plus particulièrement dans le détail des spécifications. Une liste limitée de savoir-faire pourrait comprendre les éléments suivants :

En phase de démarrage :

- Demander des spécifications écrites par le client
- Obtenir le cas échéant que chaque page soit signée par le demandeur, qui s'en trouve ainsi responsabilisé.

- Appliquer la même méthode au niveau des chefs de service et des différents détenteurs du savoir-faire.
- Faire relire les résultats obtenus pour confirmation, au demandeur du projet.

En phase de codage

- Paramétrer l'éditeur en mode sévère, c'est-à-dire avec détection de toutes les erreurs.
- Utiliser des noms de variables longs, afin de faciliter la phase de maintenance, lorsque les spécifications évolueront et que de nouveaux programmeurs

prendront le relais.

- Intégrer des remarques décrivant ce que fait chaque module.
- Intégrer des remarques sur le pourquoi de chaque action.
- Documenter les particularités du codage
- Limiter à moins de trente lignes la taille de chaque procédure.
- Intégrer, sous forme d'objets indépendants, ses propres outils de vérification et de débogage, aux programmes en cours. Par exemple, si le système le permet autoriser les tests dynamiques en cours.

auparavant comme impossibles ou improbables, et qu'il évitait comme un skieur de slalom, avec une habileté instinctive.

Les enjeux économiques d'une bonne stratégie de prévention et d'exclusion des bugs sont importants. L'industrie américaine évalue le coût de leurs maléfices à cinquante neuf milliards de dollars annuels, soit 0,6% du produit national brut. Cependant, les interactions entre les erreurs de spécifications et les erreurs de code transforment peu à peu ce qui devrait rester une simple affaire de méthode et de rigueur, en une quête du Saint-Graal. Comme le disait avec bon sens le grand-père de ma fille, l'informatique rend fou !

■ Jacques De Schryver

La psychologie du test

La vocation des tests est économique. Elle consiste à réduire les coûts et les délais. Si comme le suggère Glenford J. Myers, le but d'un test est de démontrer qu'un programme ne comporte pas d'erreurs, alors on tendra inconsciemment vers ce but, avec les oublis catastrophiques qui en résultent. Il s'agit en quelque sorte de définir une programmation neurolinguistique telle que l'on ressent une satisfaction pour chaque bug ou erreur de conception trouvés. Un peu comme un enfant qui ramasse sur la plage un ensemble de coquillages et crustacés que ses parents s'empresseront de jeter, le testeur se trouve confronté à cette situation ludique et dramatique tout à la fois. Les belles erreurs qu'il a mises à jour seront détruites impitoyablement. Paradoxalement, il est à la fois un constructeur, puisqu'il augmente la fiabilité des logiciels, et un destructeur, puisqu'il contribue à l'éradication des morceaux de code imparfaits, buggés ou insuffisamment spécifiés, qu'il a lui-même découverts.

L'art de détecter les bugs introuvables

Certains bugs manifestent une nature paradoxale qui les classe dans la catégorie des introuvables. Plusieurs jours, voire plusieurs semaines sont parfois nécessaires à leur éradication. Bugs systèmes, bugs structurels, bugs combinatoires, bugs de non-dit s'en donnent à cœur joie.

Certains projets capotent parce que le client a perdu confiance. Déçu par son système de développement, découragé par la dérive des délais et des coûts, il abandonne. Le responsable de ce désintérêt est parfois un simple bug invisible, qui hypothèque lourdement un travail pourtant avancé. Amertume, procès et rancœurs peuvent cependant être évités. Des principes et méthodes existent, qui aident à retrouver l'aiguille dans la meule de foin, sans pour autant y mettre le feu.

Utiliser les forums

Certaines erreurs exigent à la fois de l'expérience et du savoir-faire. Le débutant et le programmeur moyen n'auront acquis l'expertise suffisante qu'après un certain nombre de mois. La solution rapide passe éventuellement par les forums. Certains sont tellement fréquentés que les bonnes réponses s'y affichent rapidement. Outre le gain de temps, leur lecture favorise le développement de connaissances précieuses qui, le moment venu, favoriseront l'instinct. Comme le soulignait Louis Pasteur, la découverte ne favorise que l'esprit déjà préparé. Ayant soudain commencé à perdre des fiches sur un gros programme qui évoluait pourtant bien, je posais la question, j'interrogeais le forum Windev de PCSoft. La réponse que je reçus était plutôt vague, de mon point de vue. « *Regarde le contexte hyperfile indépendant* », me conseillait un participant. De fait, j'ignorais tout de cette notion qui contenait cependant la solution. Certes, ce point était peut-être peu documenté à l'origine, mais il m'avait coûté cher. Après avoir coché l'option correspondante, tout rentra dans l'ordre. Parfois, on réalise à la lecture des questions posées que l'on se trouve dans le même cas

que le demandeur, à ceci près qu'on l'ignorait. Chaque langage possède ses particularités. De façon naturelle, intuitive et implicite, le développeur présume à tort que ses habitudes et son savoir-faire ancien s'appliquent à la situation nouvelle. C'est ainsi que des pointeurs mal positionnés occasionnent des drames en écrasant une fiche par une autre. Cela survient sournoisement après une amélioration. Une zone de liste, une combo, une table ou un objet arbre se réaffichent pour tenir compte d'une mise à jour. Trop tard pour l'ignorant innocent, les pointeurs ne correspondent déjà plus à la fiche en cours. Il existe désormais un décalage entre le fichier et l'écran. L'écran affiche des données de rang n , tandis que les pointeurs se trouvent sur la dernière fiche lue... L'action préventive de la lecture des forums n'est donc plus à démontrer.

La nature du non-être

Lorsqu'on recherche un bug introuvable, on risque d'émettre à plusieurs reprises la même hypothèse et donc de tourner en rond. Il existe une méthode pour éviter cela. On crée un petit tableau dans lequel on notera successivement et sur deux colonnes ce qu'est le bug et ce qu'il n'est pas :

Ce qu'il n'est pas, par exemple :

- Ce n'est pas une faute de syntaxe
- Ce n'est pas une mauvaise affectation de variable
- Ce n'est pas une erreur de type (entier au lieu de décimal)
- Ce bug n'existe pas dans la version collaborative, ni dans la version de test (il pourrait donc être lié au serveur)
- Il n'est pas présent dans la version de test (seulement dans l'exécutable serveur)

1. MÉTHODES

Depuis quand le problème survient

- Depuis une ou plusieurs modifications
- Depuis un changement de version du programme
- Depuis un changement de la version du langage de développement
- Depuis un changement de fournisseur d'accès
- Depuis l'installation d'une nouvelle imprimante
- Depuis une modification d'interface (une option d'un objet a été modifiée accidentellement ?)
- Depuis un changement de paramètres dans l'éditeur (une police de caractère provoque des confusions entre l,i et 1)

En quelles circonstances

- Lors de l'impression
- Jamais lors de la première impression
- Sur toutes les impressions, mêmes lancées depuis un autre programme tel que Word.
- Lorsque les valeurs dépassent un certain seuil.
- En cas de valeurs négatives.
- Lorsqu'une personne précise utilise le programme.
- Sur un poste précis et un seul.
- Au bout d'un certain temps
- Après une opération précise et à ce moment

là seulement.

- Lorsque l'ordinateur est resté allumé trop longtemps.

La puissance de cette méthode réside dans le fait de ne pas explorer les branches inutiles de ce que le bug n'est pas.

La liberté de chercher

Le stress, ennemi du programmeur, aveugle de façon sélective. On ne voit plus ce qui est devant soi. On présume que l'erreur n'est pas à l'endroit où l'on vient de vérifier. On tourne en rond. Il existe à ceci des remèdes techniques et psychologiques. Dans la première catégorie, l'éditeur joue un rôle capital. Parmi les qualités d'un bon environnement, citons les caractéristiques suivantes :

- Utilisation de couleurs différentes selon le type d'objet. On pourra ainsi éviter les confusions entre rubrique d'enregistrement et zone de texte.
- Interception immédiate des boucles non fermées.
- Surveillance des conflits entre variables locales et globales.
- Assistants en ligne, avec rappel de la syntaxe et de l'ordre des paramètres. On évite ainsi les pièges des syntaxes anti-intuitive, par exemple FichierVersEcran(Ecran, Fichier).
- Saisie assistée. On évite de se tromper d'une

lettre dans les noms de variable longs, ou de confondre lorsqu'il existe des variables proches. En effet, le système affiche alors une boîte de liste dans laquelle choisir.

■ Jacques De Schryver

La représentation mentale

Le bug introuvable naît souvent d'une représentation mentale insuffisante ou erronée. Dans le premier cas, le flou règne en maître grâce à une absence de repères. Dans le second, on s'installe dans la certitude, mais à tort, sans pertinence dans les repères. Une bonne représentation mentale suppose la mémoire du problème et à tout le moins, de sa logique. Or les êtres humains sont inégaux devant la mémoire. Selon des statistiques anglaises réalisées dans les années soixante, une personne sur deux mille possède la mémoire photographique. Une personne sur cinquante mille possède une mémoire synesthésique, c'est à dire avec mélange des sens. Aux championnats de France d'échecs, en discutant avec les Maîtres français, j'appris que certains pouvaient jouer en aveugle sept parties simultanées. Et surtout, sans effort. Nicolas Tesla, contemporain d'Edison et spécialiste du fonctionnement des ondes longues, visualisait les détails des machines qu'il allait construire, au point de pouvoir prédire leur degré d'usage. Il les faisait tourner dans sa tête !

En termes de logiciel, on pourrait définir la mémoire, de façon tautologique, comme l'absence d'oubli. Cela signifie que le développeur ne se fera pas d'illusions sur le code qu'il a écrit. En le repassant dans sa tête, il retrouvera les erreurs de structure. Il se souviendra des détails de syntaxe. La difficulté demeure que les grands mnémotechniciens ne possèdent pas toujours le mode d'emploi de leurs dons, et que celui-ci ne s'accompagne que rarement de créativité. Enfin, lorsqu'un petit génie a fait ses preuves dans l'écriture et le débogage de programmes, le principe de Peter entre en action. On le nomme chef de projet en lui supposant des qualités de gestionnaire et d'animateur d'équipe.

L'erreur sans trace

Cet exemple a été résolu sur le forum Windev, après un travail d'équipe. Après une erreur sur une transaction, ordre est donné d'opérer la transaction inverse. L'opération réussit, et un enregistrement est ajouté dans le fichier de la gestion d'erreur, où sont stockés les problèmes rencontrés, avec l'heure et la date. Dans ce cas précis et dans ce cas seulement, le fichier d'erreurs reste vide. L'erreur est enregistrée comme le mode Trace le vérifie. Cependant, sans raison traçable, elle disparaît aussitôt. Dans tous les autres cas elle demeure et perdure, comme dans tout bon fichier qui se respecte. Ce n'est pas une erreur de syntaxe. Ce n'est pas une erreur de code. Néanmoins, le compte rendu de l'erreur persiste à refuser obstinément de s'enregistrer durablement, et ceci, uniquement dans le cas de l'erreur de transaction.

Solution

Une fois acquis qu'il ne s'agit pas d'une erreur liée au code, il reste à analyser le contexte, au niveau des paramètres. En relisant le mode d'emploi, il est spécifié que le fichier sur lequel s'opère la transaction doit être précisé, faute de quoi, le mécanisme opère sur l'ensemble des fichiers concernés. Dans ce cas, l'absence de paramètres active la seconde option. Il en résulte que lorsque la transaction du fichier principal est annulée par une opération de transaction inverse, l'opération consistant à écrire dans le fichier d'erreurs est également annulée aussitôt écrite. Il faut donc renseigner le paramètre pour que tout rentre dans l'ordre. On peut présumer qu'une remise à zéro de l'erreur, juste après avoir effectué la transaction inverse, et avant d'enregistrer l'erreur, résoudrait également le problème.

France Telecom Une cellule de tests mutualisés

L'opérateur a privilégié la démarche proactive en passant tous ses développements au peigne fin, afin de limiter les dysfonctionnements et anomalies, synonymes de coûts. La société a mobilisé d'importants moyens en mettant en place une cellule de tests composée de 200 personnes, avec une politique de mutualisation qui décuple ses moyens.

France Telecom pratique des tests depuis 1996, mais c'est seulement en 2000 que le groupe met en place une cellule dédiée, regroupant près de 200 personnes. « C'est la plus grande équipe en Europe, souligne Lionel Verlainne, directeur de l'intégration et de l'évaluation des performances de France Telecom et co-auteur de l'ouvrage "Tests de Performance des applications Web", paru aux éditions Eyrolles. En 2000, quand j'ai pris la direction de ce pôle de Test, France Telecom a souhaité placer beaucoup d'énergie en amont du déploiement des applications, afin de faire du préventif plutôt que du curatif ». Sur les 200 personnes de la cellule, 150 réalisent des tests d'intégration technique à l'aide des outils Test Director et WinRunner de Mercury, les 50 restants s'appuyant sur LoadRunner pour les tests de performance.

Mutualiser pour réutiliser

La réussite de cette politique repose sur un principal facteur : la création d'une cellule favorise en effet la mutualisation des ressources et la création d'un stock de tests sur lequel l'opérateur ne cesse aujourd'hui de capitaliser. « Nous avons choisi des outils qui nous permettent de stocker les tests pratiqués pour les réutiliser plus tard sur d'autres applications, explique Lionel Verlainne. Aujourd'hui, le nombre de tests créés est en diminution constante, tandis que la réutilisation de ceux qui existent ne cesse d'augmenter. Nous capitalisons ainsi sur nos acquis et gagnons en réactivité ». Pour autant, l'opérateur s'est très vite aperçu que le test systématique, notamment de performances, n'était pas forcément la meilleure approche en termes économiques : « Nous avons donc élaboré une méthodologie

basée sur la plus value que nous pouvons apporter, en l'appliquant aux gros projets ou à ceux qui sont sensibles, tels que les applications du Web dont la qualité peut avoir des répercussions sur notre image », précise Lionel Verlainne.

Au total, les tests de performances sont pratiqués sur 50% des applications conçues par France Telecom, soit approximativement une centaine d'applications par an, chacune pouvant comporter un ou plusieurs tests. Les tests de non régression et fonctionnels sont en revanche réalisés systématiquement. France Telecom vérifie ainsi la compatibilité des nouveaux développements avec les systèmes sous-jacents avant leur déploiement, contrôle la cohabitation entre nouvelles et anciennes applications et vérifie, par exemple, qu'un nouveau développement n'efface pas par mégarde une dll indispensable à une autre application, que les interfaces ou écrans sont conformes au cahier des charges, etc.

Des bénéfices concrets

En termes de bénéfices, la création de cellule et la mutualisation opérée permettent déjà à l'équipe d'assumer une charge de travail plus importante. Mais pour Lionel Verlainne, les avantages se situent surtout au niveau de l'exploitation et des investissements en matériel : « mieux contrôlées, les applications nécessitent moins d'interventions. Sur le poste client, par exemple, avant la mise en œuvre de ce pôle, nous avions un taux moyen d'indisponibilité des applications au dessus de la dizaine d'heures par mois. Aujourd'hui, nous sommes tombés à moins d'une heure ». Les économies en matériel, pour leur part, sont réalisées grâce au contrôle de conformité des applications par rap-



« Nous n'avons pas calculé exactement le retour sur investissement, mais il est plus qu'évident en termes de non investissement en matériel et de disponibilité des applications. Si les utilisateurs ont des applications exemptes de bugs avec de meilleurs temps de réponse, ils peuvent travailler plus vite et donc être plus efficaces pour traiter les commandes des clients »

Lionel Verlainne, directeur de l'intégration et de l'évaluation des performances de France Telecom

port au cahier des charges initial, des mesures du comportement de l'application en termes de temps de réponse et de gestion de la charge ayant permis de détecter et corriger des anomalies. « Certaines applications sont tout simplement mal développées, mal paramétrées ou n'ont carrément pas été prévues pour le système d'exploitation qui les héberge, explique Lionel Verlainne. Nos processus de tests permettent de détecter ces anomalies et nous réalisons alors des optimisations qui permettent par exemple de passer de 100 utilisateurs, avec un temps de réponse de 2 secondes, à 1000 utilisateurs avec toujours le même temps de réponse. Nous évitons ainsi les extensions matérielles inutiles ». Qui a dit que le test coûtait plus cher qu'il ne rapportait ?

■ Marie Varandat

Activité : opérateur

Chiffre d'affaires : 47,2 milliards € en 2004

Nombre d'employés : 240 145

Outils utilisés : WinRunner (tests fonctionnels), TestDirector (solution globale de gestion des tests), LoadRunner (tests de performance)

Tests automatiques - tests manuels : choisir une méthode adaptée

Tout le monde dans l'industrie IT se remet en question à un moment ou un autre et réfléchit au meilleur moyen de tester une application. Est-ce que le test manuel

suffit ? À quel moment l'équipe de développement doit-elle

faire appel aux outils et automatiser les tests ? Est-ce qu'il est plus effi-

cace d'utiliser à la fois le test manuel et le test automatique pour vérifier

qu'une application est complètement fonctionnelle ? La réponse à ces questions réside dans la complexité de l'application mais aussi finalement dans la créativité de l'équipe de développement. Cet article explorera les bonnes pratiques permettant de déterminer le meilleur process de test d'une application logicielle.



Évaluer la complexité d'une application

L'architecture d'une application logicielle est constituée de nombreuses parties et éléments. Elle peut se composer de serveurs d'applications, bases de données, services web, anciens systèmes, clients, sites web ou toute sorte d'autres composants, chacun d'entre eux interagissant avec les autres, de façon encore plus complexe et avec des variations infinies.

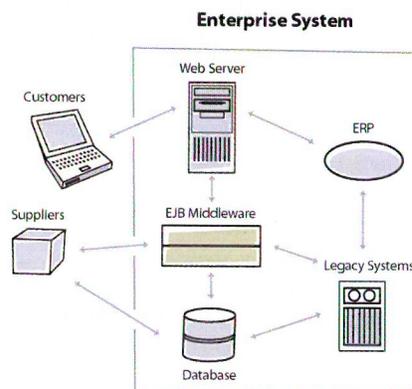
Afin de tester de telles applications complexes, le meilleur scénario pour les équipes de développement serait de capturer le comportement de l'application et de le reproduire automatiquement. Mais c'est plus facile à dire qu'à faire. En effet, créer des tests réutilisables dans ce contexte est un processus difficile et pénible. La première chose que l'on aurait tendance à faire est de transférer l'application vers une zone virtuelle neutre pour réaliser plus de tests.

Une zone virtuelle est une version privée de l'application complète. Elle peut consister en un système multicouche compliqué, mais elle peut aussi être un système très simple. Le but de l'opération est d'avoir à disposition une zone sécurisée où les modifications réalisées sur l'application sont testées, avant de passer en production. De cette façon, les erreurs peuvent être détectées

et corrigées avant d'atteindre le public. Mais le système virtuel a ses limites : il implique des tests réalisés dans une situation irréaliste, avec des résultats de tests forcément peu fiables et une maintenance difficile. C'est une raison suffisante pour éviter le système virtuel et adopter d'autres techniques toujours dans le but d'automatiser des tests.

Simplifier avant d'automatiser les tests

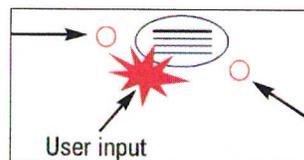
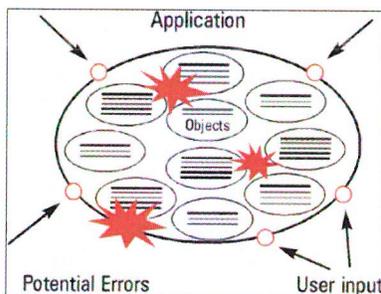
Pour pouvoir automatiser leurs tests, les équipes de développement doivent d'abord parvenir à simplifier l'application. Pour cela, elles doivent rendre les composants individuels moins dépendants les uns des autres, puis exécuter les tests sur une seule machine plutôt que sur un système virtuel. Si chaque élément de l'application peut être testé séparément et donc éviter la maintenance d'éléments multiples dans l'infrastructure, l'automatisation des tests devient beaucoup plus simple. Autrement dit, lors de tests sur une application complexe,



la clé est finalement de simplifier celle-ci en extrayant des composants et en utilisant des outils automatiques pour tester ces composants plus petits et plus simples.

Lors des tests sur un composant, il faut aussi pouvoir simuler le comportement du reste du système pour garantir que la partie testée se comporte comme prévu. Par exemple, lors de tests sur un service web, celui-ci peut avoir à traiter des requêtes clients, des informations issues de bases de données, des réponses clients et bien d'autres cas d'utilisation. Pour chacun de ces cas, il faut décider des différentes entrées, sorties et spécifications qui vont créer le flux logique d'informations.

Prenons, par exemple, un système bancaire en ligne



qui permet de transférer de l'argent d'un compte à l'autre en passant par un service web. Si vous souhaitez tester cette partie d'application, vous devez d'abord l'amener à l'état le plus proche de l'état définitif pour s'assurer qu'il se comporte comme prévu. Pour cela, il faut veiller à ce qu'un compte puisse être créé en passant par l'interface Web, l'argent transféré au moyen du service web et que cet argent soit disponible dans la base de données. Vous devez donc concevoir des tests unitaires pour chaque module de l'application et vous assurer que l'application peut traiter les entrées, qu'elles soient positives ou négatives. Automatiser les tests unitaires de cette façon est une tâche simple puisqu'il s'agit ici de parties et process réutilisables et non de l'application entière.

Le rôle des tests manuels

L'automatisation des tests de process réutilisables semble maintenant beaucoup plus simple, cependant, il n'en est pas de même de la vérification du process de test et des résultats. Les scripts et outils automatiques permettent effectivement d'automatiser des process à

l'intérieur de l'application mais les résultats livrés par le système peuvent être tellement nombreux qu'ils rendent l'automatisation impossible. Si l'on reprend l'exemple de service web utilisé ci-dessus, la base de données, l'interface web, le service sont probablement remplis de résultats de comptes et d'informations, et pourtant vous devez vérifier que ce sont des données correctes. Tester ce genre de données devient très vite compliqué à répéter automatiquement. Parfois, c'est tellement compliqué que la seule bonne méthode est de faire ces tests à la main. C'est pourquoi il est recommandable de réaliser les tests manuellement dans les cas où l'application change lentement et les outils automatiques ne peuvent produire les résultats d'eux-mêmes, ou répéter les étapes adéquates. Dans ces cas, le développeur devient l'outil de reconnaissance des patterns et doit tester l'application manuellement.

Conclusion

Les applications logicielles peuvent présenter des variations infinies et il y a encore plus de façons d'utiliser le test automatique et le test

manuel pour vérifier leur fonctionnalité. Mais tout ne peut pas être testé automatiquement et il arrive que l'automatisation n'ait pas de raison d'être dans le cas de systèmes complexes. En fait, tout dépend de l'application, des circonstances et de la quantité de travail nécessaire à investir dans l'application.

D'autre part, nous l'avons vu, il y a différentes façons de tester une application, mais au bout du compte ce sont les compétences du développeur qui décideront des techniques utilisées. Tester une application n'est pas un travail adapté pour quelqu'un qui n'est pas créatif. En effet, concevoir une expérience sous contrôle qui peut être réutilisée et testée de nombreuses fois nécessite une intelligence vive et des énergies créatives.



■ Dr. Adam Kolawa

Adam Kolawa est Président et cofondateur de Parasoft.

Docteur en Sciences

Physiques, il est co-auteur du Bullet proofing Web Applications (Hungry Minds 2001).

Ak@parasoft.com

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

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

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

Profitez-en !

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



Nat System
Stratégiquement vôtre

1. MÉTHODES

Nuxeo pratique le test en open source

Editeur d'une plate-forme de gestion de contenu open source s'appuyant sur Zope, Nuxeo pratique avant tout des tests unitaires pour vérifier son code module par module. Pour l'éditeur, l'application exempte de bugs est avant tout une question de bonnes pratiques et de discipline.

Très présent sur le développement et les services autour du serveur d'applications Zope, Nuxeo est à l'origine d'une plate-forme open source de gestion du contenu en entreprise. Baptisée CPS (Collaborative Portal Server), elle regroupe des fonctions de bases telles qu'un portail, le stockage unifié de documents, la gestion des types de documents, la recherche plein-texte, le versioning, le multilinguisme, etc. ainsi que des fonctions additionnelles telles que forum, calendrier partagé, webmail, chat interactif, lettres d'informations et listes de discussions, etc.

En constante évolution, cette plate-forme est le fruit de développements internes et d'une collaboration avec une communauté de développeurs open source externes à la société et fait l'objet d'un grand nombre de tests réalisés à l'aide d'outils open source. Mais pour Stéphane Fermigier, PDG de la société, « une application exempte de bugs, c'est d'abord une question de bonnes pratiques et de discipline. Aussi bons soient-ils, les outils ne sont pas magiques : ils ne garantissent rien. D'ailleurs, il est connu dans le monde du génie logiciel que l'adoption de nouveaux outils entraîne au début une perte de productivité. Et, de fait, le test a un coût : les créateurs d'eXtreme Programming et du développement dirigé par les tests estiment en effet qu'il faut écrire environ deux fois plus de lignes de code pour les tests que pour l'application testée ! Mais le retour sur investissement, en termes de qualité du code produit, compense largement ce coût initial ».

Des tests unitaires systématiques

De la discipline, Nuxeo n'en manque pas. Très tôt, la société a en effet imposé à ses développeurs des tests unitaires systématiques avant chaque "commit" ou, en d'autres termes, la sauvegarde de son code sur le dépôt central

de la société. « Autrefois, nous avons effectivement des développeurs qui, sous prétexte d'être pressés par le temps, ne réalisaient pas leurs tests, explique Stéphane Fermigier. Mais cette mentalité tend à disparaître : aujourd'hui, un bon développeur sait très bien que pour faire du travail de qualité, il doit tester ». Les tests unitaires sont réalisés à l'aide de PyUnit, un dérivé, pour le langage Python que Nuxeo a retenu pour sa plate-forme, de JUnit, framework de tests unitaires en environnement Java.

Le processus de développement dirigé par les tests

En parallèle de cette approche, Nuxeo a également adopté, pour certaines applications, le processus de développement dirigé par les tests où, en d'autres termes, les tests sont écrits avant l'application. Une démarche facilitée par un outil assez original, propre à l'univers de développement Python : Doctest. « Cet outil est d'abord prévu pour écrire la documentation, explique Stéphane Fermigier, mais il permet également de dérouler un scénario d'application avec le résultat attendu, d'insérer le code et de le vérifier dans la foulée. C'est vraiment un outil étonnant qui permet à la fois d'écrire les spécifications, les tests et la documentation technique ».

Un contrôle méthodique

Enfin, en bout de chaîne, Nuxeo pratique des tests d'intégration, écrits avec le framework PyUnit et automatisés à l'aide de BuildBot, un outil open source d'intégration continue. Ils permettent de vérifier que les modules conçus par les développeurs s'intègrent correctement et, également, que l'application fonctionne sur différents environnements, comme par exemple Linux, Windows et MacOS X. Stéphane Fermigier reconnaît ne faire que depuis peu des tests



« Les communautés open source n'ont pas toujours eu bonne réputation en matière de tests, mais c'est bien fini. Aujourd'hui nous attribuons une grande importance aux tests de manière générale, y allouons des ressources et les outils se développent progressivement pour couvrir les parties jusqu'alors un peu négligées de ces besoins ».

Stéphane Fermigier, PDG de Nuxeo

fonctionnels : « Le test unitaire est rapide comparé au test fonctionnel et l'on manquait il y a encore six mois d'outils open source pour réaliser ces tests ». Une absence d'outil à laquelle Nuxeo a commencé à remédier avec le projet Funkload, une solution complète destinée aux tests fonctionnels, de non-régression et de performance HTTP.

■ Marie Varandat

Activité : éditeur open source

Création : 2000

Chiffre d'affaires : 1,5 M €

Nombre d'employés : 25

Estimation des développements pour CPS : 180 000 lignes de code Python, 40 développeurs, plusieurs centaines d'installations à travers le monde

Principaux outils utilisés : PyUnit pour les tests unitaires et d'intégration, BuildBot pour l'automatisation de tests d'intégration, Doctest pour le développement orienté par les tests, Funkload pour les tests fonctionnels et de performance.



vosre nom vaut bien un oui ! nouvelle extension : .eu



100 domaines .eu offerts !
attribués par tirage au sort**

Pré-enregistrez votre domaine
N'attendez pas qu'il soit trop tard : protégez vos noms et vos marques sur Internet ou le nom de votre futur site !

SERVEUR À LA UNE : PACK SERVEUR PRIVÉ LINUX OU WINDOWS

INNOVATION TECHNOLOGIQUE !

Toutes les fonctionnalités d'un serveur dédié Linux ou Windows au prix d'un mutualisé !

DÉTAILS DU PACK SERVEUR PRIVÉ AMEN
Technologie Virtuozzo • Adresse IP fixe
• Accès root ou administrateur • Interface d'administration PLESK • 400 Mo (extensible à 2 Go) • Multi-sites • Multi-domaines • Emails illimités • 10 bases MySQL • Trafic illimité

à partir de 10 € HT/MOIS¹ soit 11,96 € TTC/mois

AMEN, Registrar Officiel auprès de l'EURID pour l'enregistrement du .eu, gère déjà plus de 200 000 noms de domaine.

Avec le Pack Web Nom, **pré-enregistrez dès maintenant** vos noms de domaine en .eu au prix exceptionnel de **12 € HT/AN*** et **beneficiez de nombreux services associés OFFERTS** : emails, anti-virus-anti-spam, hébergement, outil de création de site...

NOUVEAU ! Pack Web Nom .eu

Enregistrement de votre nom de domaine .eu + service DNS + redirection emails illimitée + redirection web + hébergement 2 Mo + anti-virus/anti-spam + WEB SITE CREATOR : outil de création de site + **AVANTAGE AMEN .EU** : remboursement en cas de non aboutissement de la demande !*

12€ HT/AN* soit 14,35 € TTC/AN

Nos 10 engagements*** : • satisfait ou remboursé • votre pack en 1 H • surveillance réseau 24x7 • mise en service offerte • haute disponibilité 99,9 % • monitoring proactif 24x7 • interface d'administration en ligne • réseau redondant • bande passante garantie • aucun frais caché

www.amen.fr
.eu registrar

0892 55 66 77
0,34 € TTC/mn depuis la France 9H-19H

AMEN RCS PARIS: B 421 527 797. IN WEB WE TRUST - Nous croyons au web. * hors frais de dossier (AMEN et EURID) applicables pendant la période de soumission qui seront communiqués ultérieurement par EURID et qui seront accessibles sur notre site Internet. L'EURID impose de percevoir des frais de dossier en cas de non aboutissement de la demande d'enregistrement, soit du fait du pré-enregistrement du nom de domaine par un tiers, soit en cas de non aboutissement du fait du client. ** tirage au sort effectué par l'Office Français de la Propriété Intellectuelle à la fin de la période de soumission. Les noms de domaine qui seront attribués par tirage au sort seront attribués par ordre alphabétique. Les noms de domaine qui ne seront pas attribués par tirage au sort seront attribués par ordre alphabétique. *** selon nos statistiques mensuelles et nos CGV. 1 Prix pour un serveur Linux, à partir de et pour un contrat de 12 mois. Prix total sur la période : 120 € HT (143,52 € TTC). Conditions générales de vente sur www.amen.fr. Prix au 01/11/2015 pour un paiement annuel, modifiables sans préavis. Offre valable dans la limite des stocks disponibles.

2. OUTILS

Les outils de tests au cœur du développement

Le marché des outils de tests évolue pour couvrir tout le cycle de développement et faciliter les interactions à chaque étape du processus entre les développeurs, les utilisateurs et les testeurs.

Le test a toujours existé. Mais, faute de temps et de moyens, il était souvent relégué au second plan. « En moyenne, on estime que pour 100 euros de lignes développées, il faut compter 30 euros de tests » souligne Stéphane Dulor, Directeur commercial de Metaware, entreprise d'ingénierie qui a fait du test une méthodologie de développement. Un coût dont beaucoup d'entreprises ont cru pouvoir se passer, pour s'apercevoir au final que l'absence de tests coûtait parfois beaucoup plus cher : applications non conformes au cahier des charges, bugs et autres anomalies, sans compter l'impact sur l'image de marque et le chiffre d'affaires d'une société, comme le souligne François Guérin, Responsable Performance de cc-Hubwoo, plate-forme d'e-procurement : « les performances de notre plate-forme ont un impact direct sur notre chiffre d'affaires, notre business model est basé sur les transactions qui y transitent. C'est donc, non seulement une question d'image, mais également de disponibilité : tout arrêt du système, aussi infime

soit-il, équivaut à autant de commandes qui ne passeront pas par notre plate-forme ».

Intégration avec les outils de développement

Face à la pression du Web et aux exigences de qualité logicielle, notamment, on assiste à un revirement du marché. L'offre de produits de tests de code est en effet progressivement intégrée aux ateliers de développement de Borland ou Microsoft, ou encore, de Rational. Le test unitaire et de code est pratiqué de façon systématique. Intervenant en amont, ces fonctions embarquées dans les ateliers de développement sont complétées de trois principales catégories d'outils de tests : performance, fonctionnel, intégration/non régression, les deux derniers pouvant être regroupés dans des offres globales. Les premiers permettent, à l'aide de scénarios, de simuler des utilisateurs, afin de vérifier la capacité de l'application et de l'architecture matérielle sous-jacente à monter en charge. La majorité des solutions s'intéressent aux architectures trois-tiers du Web mais il

existe aussi des produits pour applications sur mainframe, tel Certify de Worksoft et, bien entendu, pour des développements de type client/serveur.

La seconde catégorie est là pour vérifier la conformité de l'application au cahier des charges. Le principe même de ces outils est moins de vérifier la programmation intrinsèque d'une application que d'analyser les spécifications initiales, pour ensuite contrôler que les interfaces utilisateurs, les API, la gestion des bases de données, ou encore de la sécurité, sont conformes.

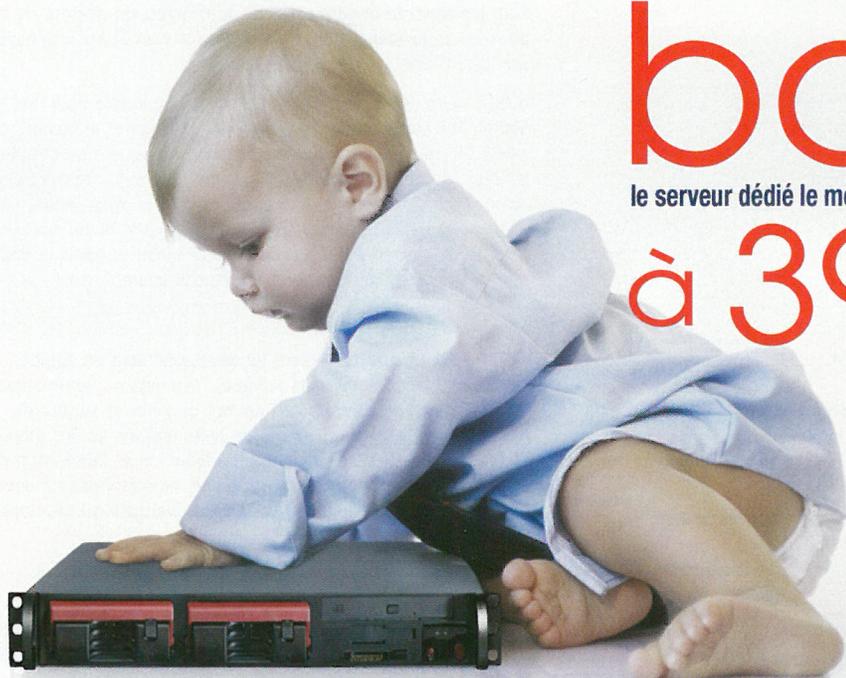
Enfin, les tests d'intégration et de non régression permettent de vérifier que le code nouvellement déployé ne nuit pas à l'existant en effaçant une dll essentielle au fonctionnement d'une autre application, par exemple. Enfin à ces trois principales catégories, il faut encore ajouter des consoles qui automatisent l'exécution de tests telles, Test Director ou QACenter, de respectivement Mercury et Compuware, les deux leaders du marché du test.

■ Marie Varandat

Principaux éditeurs du marché

Editeur	Offres	Caractéristique
Compuware www.compuware.com	Application Reliability Solution "CARS" (fonctionnel) QACenter Entreprise Edition (intégration) TestPartner (non régression) QACenter Performance Edition et QALoad (performance)	Editeur qui domine le marché du test avec Mercury disposant d'une offre très complète supportant les environnements client / serveur, L4G, Java et Web.
IBM/Rational http://www-306.ibm.com/software/awdtools/tester/robot/	Rational Robot	Rational n'est pas présent sur le test de performance, Robot automatisant les tests de régression et fonctionnels pour applications Web et client/serveur
Mercury www.mercury.com/fr/	QuickTest Professional, WinRunner et Business Process Testing pour la partie fonctionnelle, LoadRunner pour la performance, Test Director pour l'automatisation	Probablement une des offres les plus complètes du marché s'appuyant sur une méthodologie de gouvernance de la qualité (BTO)
Radview www.radview.com	TestView, WebFT (fonctionnel) WebLOAD (Performance)	Offre destinée uniquement aux applications Web permettant de définir des scénarios de tests automatisés et de centraliser les campagnes de tests.

A la naissance de vos projets, il y a le serveur dédié AMEN...



baby

le serveur dédié le moins cher du marché

à **39 €** HT/MOIS*
soit 46,64 € TTC/MOIS

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



- > AUCUN FRAIS DE DOSSIER
- > AUCUN FRAIS DE MISE EN SERVICE
- > HAUTE DISPONIBILITE : 99,9 % !
- > SUPPORT TECHNIQUE PAR MAIL ET PAR TELEPHONE
- > SATISFAIT OU REMBOURSE***

Serveurs Dédiés AMEN en version  Linux ou  Windows

SMALL	MEDIUM	LARGE
Athlon XP 2400 ou Sempron 2400 IDE 80 Go / RAM 512 Mo 2 adresses IP Amen Data Backup 2 Go Amen Reboot instantané Interface Plesk 7 Trafic illimité ¹	Duron ou Sempron 2800 IDE 120 Go / RAM 768 Mo 4 adresses IP Amen Data Backup 4 Go Amen Reboot instantané Interface Plesk 7 Trafic illimité ¹	Athlon XP 3000 RAID 2x160 Go / RAM 2 Go 4 adresses IP Amen Data Backup 6 Go Amen Reboot instantané Interface Plesk 7 Trafic illimité ¹
69 € 59 € HT/MOIS* 70,56 € TTC/MOIS	99 € 89 € HT/MOIS* 106,44 € TTC/MOIS	169 € 139 € HT/MOIS* 166,24 € TTC/MOIS

Nos 10 engagements*** : • satisfait ou remboursé • votre serveur en 2 H
 • surveillance réseau 24x7 • mise en service offerte • haute disponibilité
 99,9 % • monitoring proactif 24x7 • interface d'administration en ligne
 • réseau redondant • bande passante garantie • aucun frais caché

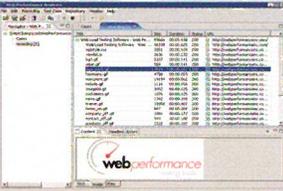
www.amen.fr

.EU registrar

0892 55 66 77

0,34 € TTC/mn depuis la France 9H-19H

2. OUTILS

Editeur	Offres	Caractéristique
Segue www.segue.com	SilkTest (fonctionnel)	Pas d'outil de test de la performance, mais Segue possède en revanche une solution de tests fonctionnels et de non régression pour les environnements Web, Java, client/serveur complètement automatisée
Worksoft www.worksoft.com/	Certify (fonctionnel)	Un des rares éditeurs à adresser directement l'univers du mainframe avec les deux leaders du marché.
Parasoft www.parasoft.com	Jtest .Test WebKing C++ Test	Jtest permet d'automatiser les tests unitaires Java et génère automatiquement les tests Junit. Permet aussi de vérifier la validité du code en incluant 500 règles de développement pour détection des violations de codage. C++ Test et .TEST sont les pendants de Jtest en C++ et .Net. Pour les tests de performances et de charges, on dispose de WebKing pour les applications web et les architectures services, l'éditeur propose SOA Test.
Empirix www.empirix.com 	E-TEST Suite E-Tester E-LOAD E-LoadExpert e-Manager Enterprise	e-TEST suite est un cycle de vie du test, disponible pour .net, pour J2EE et Siebel. Elle inclut e-Manager Enterprise pour gérer le process de tests, e-TESTER pour les tests automatiques et de régression et enfin e-LOAD pour les tests de performances et de montée en charge (ces deux outils sont disponibles séparément). Une déclinaison pour Web Services est aussi disponible. L'éditeur propose aussi e-LoadExpert dédié aux tests d'applications web à distance. Il s'utilise avant et après le déploiement, pour les tests de performances sur l'architecture.
Agitar Software www.agitar.com	Agitator Agitar Management Dashboard 	Agitator est un environnement de développement en équipe, inclut des fonctions de tests unitaires, manuels ou automatiques. Agitar Management Dashboard permet de gérer et surtout de monitorer les tests et les efforts de tests réalisés sur les projets. Il inclut les fonctions de rapports, la définition et l'utilisation de métriques de développement. On définit les méthodes de développements et chaque violation est dûment indiquée au développeur.
Webperformance www.webperformanceinc.com	Web Performance Trainer Web Performance Analyzer 	Web Performance Trainer est un outil de montée en charge incluant un modeleur de cas d'utilisation. Il permet de tester l'application web sur différents navigateurs et selon des contraintes business. En complément, l'éditeur vient de sortir Web Performance Analyzer pour mesurer les performances des pages web. Idéal pour le développeur web.
Telelogic www.telelogic.com	Logiscope TAU	Logiscope teste les applications C, C++, Ada et Java. Il vérifie le code selon les règles définies et facilite la qualité du code. Il s'intègre à l'outil TAU. TAU intègre notamment des fonctions de tests d'intégration.
Microsoft www.microsoft	Software Testers	Visual Studio 2005 Team System contient une section dédiée aux testeurs (Software Testers). Deux types de tests sont inclus dans Software Tester et Software Developer : tests unitaires et de couverture de code. La partie purement testeur comprend en plus, les tests manuels, les tests de charge (pour les applications web) et la gestion des scénarios de tests.
Borland www.borland.com	Optimizeit Core ::Tester	Optimizeit se décline en trois versions : Enterprise Suite pour les applications J2EE, le profiling, le tuning, la couverture de tests, ServerTrace s'utilise dans les équipes de pré-déploiement et de production pour vérifier les performances et découvrir les problèmes, et Profiler pour .Net. Dans la nouvelle stratégie de l'éditeur, Core SDP, on dispose de Core ::Tester. Il s'intègre avec les principaux outils de tests de montée en charge du marché et met les fonctions de tests et de qualité dans le cycle de vie lié à Core SDP.
Quest Software www.quest.com	Jprobe 	Dédié aux applications Java et J2EE, Jprobe permet de tester facilement et rapidement son projet afin d'en améliorer la qualité. Il s'intègre avec les principaux IDE Java du marché, que ce soit en 32 ou 64-bit. Il s'utilise aussi bien avec un serveur d'applications qu'avec un IDE.
Qualité First Software www.qfs.de/	qftest UI	Outil pour tester (automatiquement) les interfaces Java réalisées avec Swing. La complexité de Swing constitue une source de bugs. Il fonctionne à base de scénarios de tests et est compatible de Java 1.1 à 5.0.

Les outils de tests Open Source

Choisir un outil de test c'est une chose, mais selon quels critères ? Plusieurs facteurs doivent entrer en considération, comme le taux d'activité du projet (si un projet est peu actif, ou sur le point d'être abandonné, passer son chemin), le niveau de la documentation (ne rêvez pas trop d'une documentation en français), ou encore le type de support sur lequel vous pourrez vous appuyer (par email ou par l'intermédiaire de forums). Voici notre sélection.

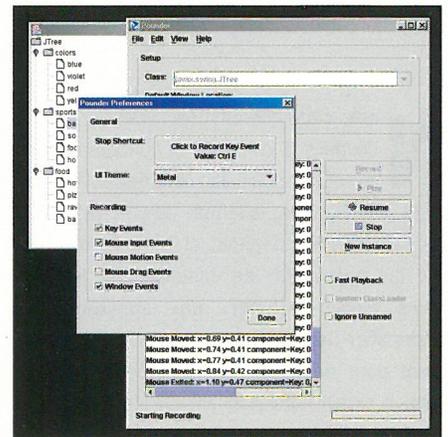
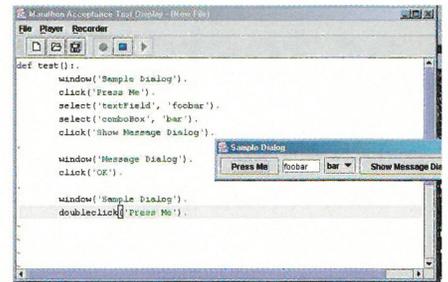
Dans le monde Open Source, un projet libre se base souvent sur une démarche qualité passant par un système de traque de bogues (*Bug Tracking System*). Par exemple, Bugzilla est un logiciel libre écrit en langage Perl, développé et utilisé par Mozilla. Il implémente le suivi de bogues (ou de « tickets ») via une interface web. Il est utilisé par de nombreuses organisations (GNOME, KDE, RedHat, Mandriva) pour suivre le développement de nombreux logiciels, tant propriétaires qu'Open Source.

En C on peut réaliser des tests « de taux de couverture » à l'aide de gcov (http://gcc.gnu.org/onlinedocs/gcc-3.0/gcc_8.html). Pour travailler avec gcov, les fichiers sources doivent être compilés avec les options `-fprofile-arcs` et `-ftest-coverage` de gcc afin de générer des fichiers d'extension « .da ». Il reste alors à lancer gcov qui utilise les fichiers .da pour donner des informations sur le taux de couverture. DejaGNU est un framework qui permet d'automatiser le test de programmes C et C++ (<http://www.gnu.org/software/dejagnu/>) en n'exposant qu'une seule interface unifiée au développeur. Il s'installe en parallèle avec Expect qui permet d'automatiser les entrées d'un logiciel. Sous Windows, vous pouvez l'installer via l'environnement cygwin. DejaGNU est très stable et très puissant. Dans la même veine il existe DART (<http://public.kitware.com/Dart/HTML/Index.shtml>) qui est nettement plus facile à mettre en œuvre tout en proposant une interface conviviale pour les résultats des tests.

Pour tous les goûts

En Java, il existe un très grand nombre d'outils de tests. Abbot est l'un de ceux-ci (<http://abbot.sourceforge.net/doc/overview.shtml>). Il a pour lui de pouvoir réaliser des tests en environnement graphique. Vous devez décrire les tests dans des fichiers XML. Il inclut également un éditeur de script (Costello) qui enregistre les actions des utilisateurs. Ce framework peut ensuite être invoqué directement du code java (via Junit, un framework Open Source de tests unitaires : <http://www.junit.org/index.htm>), ou en passant par les scripts. Ajoutons que l'équipe de développement semble très dynamique. Pounder (<http://pounder.sourceforge.net/>) est un autre outil de test Java, valable également en environnement graphique. Il autorise le chargement dynamique des composants, enregistre des scripts et les rejoue via Junit. D'autres outils de tests, connus pour Java sont JavaTest Harness (<http://sourceforge.net/projects/harness/>), Jfunc (<http://jfunc.sourceforge.net/>), Marathon (<http://marathonman.sourceforge.net/>) et QAT (<http://qat.sourceforge.net/>). Il y a aussi TestNG, s'inspirant de JUnit et NUnit, qui supporte les méthodes dépendantes, est configurable et possède un modèle d'exécution très puissant (<https://testng.dev.java.net/>). La liste est loin d'être exhaustive.

Xml Test Suite est, comme son nom l'indique, une suite de test pour le XML (<http://xmltest-suite.sourceforge.net/>). Il est basé sur le principe du test empirique, on teste et on continue, sinon on reteste. Xml Test Suite a



l'avantage de séparer la structure des tests, du code des tests et des données des tests. Un autre produit du même style est JwebUnit (<http://jwebunit.sourceforge.net/>), ou encore HTTPUnit (<http://httpunit.sourceforge.net/>), mais qui sont des outils beaucoup plus difficiles à appréhender.

Du côté des bases de données, Check_mysql (http://groovis.net/projects/check_mysql.html) est un script PERL de tests qui protège MySQL des requêtes qui pourraient planter. Un timeout de 60 secondes est fixé. Si le script ne rend pas la main dans ce délai, un courrier électronique est envoyé à l'administrateur. Le rapport inclut la requête et la sortie EXPLAIN pour déboguer et optimisation.

Enfin, comment ne pas citer le projet LTP (The Linux Test Project test suite : <http://ltp.sourceforge.net/>). LTP représente un ensemble de tests automatiques qui permettent notamment de trouver les régressions d'une nouvelle version du noyau. Les résultats dépendent de la configuration matérielle et logicielle de l'environnement de test (architecture, nombre de processeurs, quantité de mémoire, etc.). Vous trouverez certains de ces résultats à l'adresse <http://developer.osdl.org/bryce/ltp/>.

■ Xavier Leclercq

Xavier.Leclercq@programmez.com

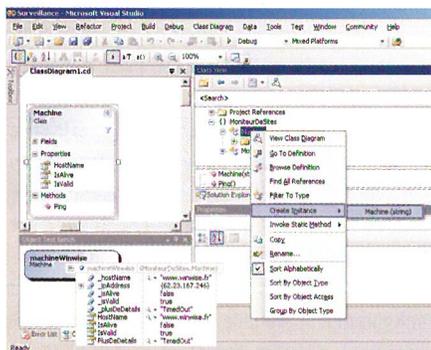
Les tests dans Visual Studio 2005

Demandez à un développeur les tâches qui l'exaspèrent le plus au quotidien, et il vous répondra dans cet ordre : la documentation et les tests. La réticence à écrire des tests est plus liée à un manque d'outils et de pratique qu'à une incompatibilité fondamentale avec le métier d'informaticien, comme en témoignent les courants récents (eXtreme Programming, MSF, et Test Driven Development).

Nous distinguerons les tests unitaires, destinés à contrôler le fonctionnement aux limites et la non-régression, les tests d'intégration, orientés vers la validation de scénarios fonctionnels, et les tests de charge, qui permettent de dimensionner les serveurs devant héberger une application et d'identifier les principaux points d'étranglement pour optimiser intelligemment l'application. Il existe naturellement d'autres types de tests, mais les trois besoins fondamentaux sont définis : vérifier l'adéquation aux spécifications techniques pour les tests unitaires, vérifier l'adéquation aux spécifications fonctionnelles pour les tests d'intégration, valider le fonctionnement en production pour les tests de charge.

Gestion des tests unitaires

Ce sont les tests les plus simples à rédiger pour un développeur. La technique la plus simple est destinée au débogage, il s'agit de l'Object Test Bench, qui permet de manipuler des instances des classes écrites sans lancer l'application.



Classe Machine permettant de vérifier que des machines sont actives.

Cette première astuce rend quelques services, mais un test, pour avoir de la valeur, doit être reproductible. Il est possible de capturer les

tests unitaires dans des projets de tests, en utilisant l'assistant Create Unit Tests.

Ces tests ne sont naturellement pas complets, voici par exemple le code qui teste la méthode Ping :

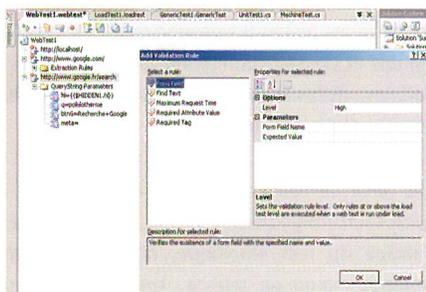
```

/// <summary>
/// A test for Ping ()
/// </summary>
[TestMethod()]
public void PingTest()
{
    string adresse = null; // TODO: Initialize to
    an appropriate value
    Machine target = new Machine(adresse);
    target.Ping();
    Assert.Inconclusive("A method that does not
    return a value cannot be verified.");
}
    
```

Autres types de tests

Tests Web

Ce test permet d'enregistrer une consultation web, puis de reparamétrer les requêtes enregistrées, par exemple, pour varier les valeurs saisies dans les champs de formulaire et éviter ainsi que la base de données ou le serveur Web ne mettent en cache certains résultats ou pages. Naturellement, on peut également définir des règles de validation pour définir que le test a échoué ou réussi, par exemple un temps maximal d'exécution de requête, ou la présence d'une valeur particulière dans la page de



retour Tests manuels

Certains tests sont impossibles à automatiser, par exemple, parce qu'ils impliquent un logiciel impossible à piloter automatiquement, ou une vérification de bon fonctionnement après restauration d'un backup. Pour vous permettre d'intégrer quand mêmes ces données au sein de vos tests, Visual Studio dispose de la notion de test manuel. Un test manuel est un document (texte ou Winword) où vous décrivez pour un testeur les opérations à effectuer. Il peut ensuite saisir le résultat du test (échec ou réussite) ainsi qu'un commentaire.

Séquences de tests

Ces tests permettent d'organiser d'autres tests dans un ordre pré-établi, notamment pour créer des tests d'intégration.

Test de charge

Ce test permet d'organiser plusieurs autres tests, notamment Web, de les jouer en simulant une charge prédéfinie, et de collecter un ensemble d'indicateurs, dont les compteurs de performance. Ces tests sont intéressants, mais demandent de bonnes connaissances et une bonne infrastructure matérielle pour être valables.

Avec Visual Studio, il est possible de classifier et d'organiser l'ensemble des tests que vous avez créés, de rejouer automatiquement une sélection de ces tests, y compris en ligne de commande (par exemple, pour ajouter les résultats à un processus d'intégration continue), et que ces tests se marient bien avec les autres outils de qualité logicielle de Visual Studio 2005 (couverture de code, analyse statique).

Ces outils de test sont un très bon premier pas, et il est probable que Microsoft propose dans les versions suivantes d'autres améliorations.

■ Pierre Couzy

Responsable de l'activité conseil
Microsoft Regional Director - France
www.winwise.fr

Testeur, un métier en pleine revalorisation

Au cœur de la qualité logicielle, le testeur fait enfin partie des leviers acceptés dans le cycle de développement pour réduire les coûts liés aux dysfonctionnements et autres anomalies qui reviennent si cher aux entreprises.

Longtemps, le métier de testeur, quand il existait dans l'entreprise !, a été dévalorisé. D'ailleurs, pour Stéphane Dulor, Directeur commercial de Metaware, entreprise d'ingénierie au forfait, spécialisée dans la Rénovation Logicielle des Systèmes d'Information et qui a fait du test une méthodologie de développement, « *autrefois, on plaçait aux postes de testeurs les gens qui n'arrivaient pas à développer. Mais aujourd'hui, c'est bien terminé et on assiste à une véritable valorisation du métier* ». Un point de vue confirmé par Yves Gandon, testeur chez Compuware, éditeur de logiciels de développement et de tests : « *Cela fait environ deux ans que l'on constate une véritable évolution dans*

le milieu informatique. Avant, nous étions les parents pauvres. Aujourd'hui, les entreprises ont pris conscience de l'importance du test et elles créent des cellules et des process pour industrialiser la démarche ».

Un généraliste de l'informatique

En tenant compte des coûts de l'application « buggée » ou non conforme au cahier des charges, les entreprises ont effet donné des galons à ce testeur qui désormais se hisse à la hauteur du développeur et peut même prétendre, en termes de perspectives d'évolution, aux postes de management de projet ou responsable qualité et méthodes, voire même directeur technique au sein de SSII ou petites entreprises. Car, si on ne requiert généralement pas du testeur de performances, ou fonctionnel, qu'il soit doté d'une grande expertise informatique, il doit en revanche disposer d'un bon vernis informatique et, à défaut, faire preuve de suffisamment de curiosité pour pouvoir couvrir le large spectre d'applications et de technologies qu'il va être amené à tester. D'ailleurs, pour Stéphane Dulor, « *un profil d'utilisateur est souvent préférable à celui d'un développeur, ne serait-ce pour éviter que le testeur soit à la fois juge et arbitre* ». La formation aux outils de tests est souvent dispensée par les éditeurs qui incluent en général une dimension méthodologie (CMMI, ITIL, Cobit, Six Sigma, etc.) plus ou moins poussée selon les besoins et la taille de l'entreprise.

Des relations pas toujours aisées avec le développeur

Par définition, le rôle du testeur est de détecter les anomalies... et de les signaler. Il n'est pas toujours évident d'aller dire à un développeur qu'il a commis une erreur. Raison pour laquelle, selon le Syntec, le testeur, ou plus



Le temps perdu du test, c'est de l'efficacité gagnée sur la suite.

Pour un développeur, c'est plus pénible de faire des tests sur une application finie que du test unitaire, parce que c'est toujours plus compliqué de revenir sur un produit quasi fini. Mais si on travaille sur du test unitaire, c'est relativement simple de mettre en place des procédures systématiques. C'est vrai que dans la première phase de développement, on perd jusqu'à 50% du temps à écrire du test. Mais c'est un temps qu'on récupère facilement par la suite, ne serait-ce que parce qu'on évite les bugs».

Julien Anguenot, développeur/testeur chez Nuxeo



Stages et formations

Outre les formations de courte durée sous forme de stages proposées par des sociétés telles que Orsys, Qualium, Valtech formation ou encore Precilog, il existe aujourd'hui des formations universitaires dispensées notamment par l'université d'Angers sous forme de MASTER 2 Professionnel Qualité & Sûreté de Fonctionnement - parcours QUALité et Sûreté de fonctionnement des Systèmes Informatiques. Certaines de ces formations portent avant tout sur les méthodes de qualité (CMMI, ITIL, Cobit, Six Sigma, etc.) et le coût de la formation varie entre 1 000 et 3 000 euros pour une durée allant de la 1/2 journée à la semaine.

exactement « l'ingénieur de validation » doit impérativement posséder des qualités de communication et savoir faire preuve de diplomatie. Enfin, pour ce poste, dont la rémunération annuelle peut varier de 27 000 à 37 000 euros, selon l'observatoire des salaires de cadreonline, il faut ajouter la rigueur et l'organisation au chapitre des qualités humaines, sans oublier les aptitudes rédactionnelles souvent indispensables pour, ne serait-ce que formaliser les recommandations après une série de tests.

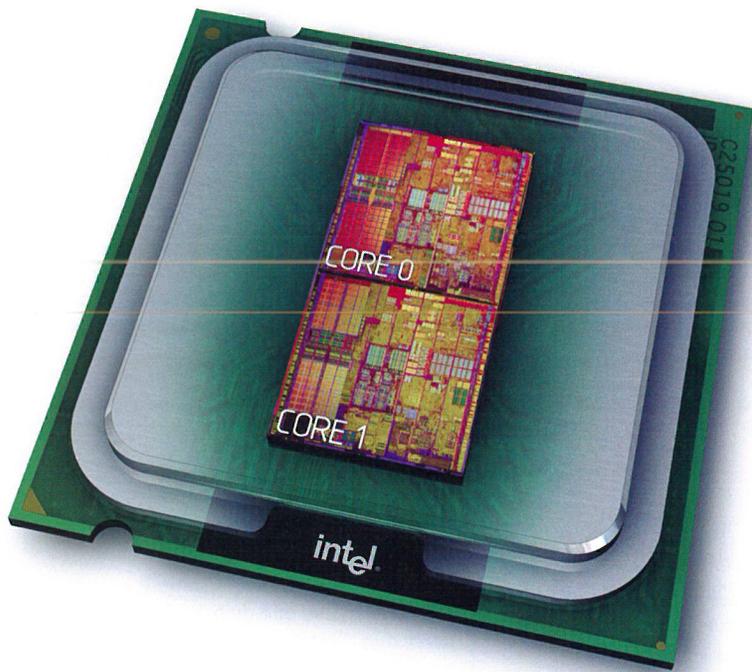
■ Marie Varandat

Pour en savoir plus

www.passinformatique.fr/40-metiers/60-20-10_fichemetier.asp?fiche=ingvalid: fiche du Syntec sur l'ingénieur de validation

- « Tests de performance des applications Web » de Lionel Verlainne – Edition Eyrolles

- www.univ-angers.fr/formation.asp?ID=GDSQUA1&langue=1: publications sur le métier de testeur et retours d'expériences d'élèves de l'Université d'Angers.



64-bit :

Après le 16-bit et le 32, voilà le 64-bit ! Les processeurs 64 se généralisent rapidement sur les serveurs et même sur le PC de tous les jours, même si ce n'est pas encore visible pour tout le monde. Apple avait montré la voie avec le processeur G5. Pour l'utilisateur lambda, le 64-bit ne lui signifiera pas grand' chose. D'ailleurs, les éditeurs et constructeurs ne tentent pas d'expliquer les réels bénéfices du 64.

Le marché du 64-bit

Longtemps atone et discret, le marché du 64-bit existe depuis des années. Essentiellement réservés aux serveurs, les processeurs 64 connaissent aujourd'hui un « engouement » auprès du public, l'argument marketing passant par là. Car, pour le moment, si les processeurs existent, les logiciels jouent les absents.

Le 64-bit constitue un triptyque : processeur, système, logiciel. Jusqu'à présent, le marché logiciel 64 se limite essentiellement aux applications professionnelles, scientifiques et serveurs. Le grand public est peu ou prou concerné. Le 64 pose un problème aux fondeurs. Comment vendre une technologie apportant peu de nouveautés ? Le seul réel avantage d'un processeur 64 est de supprimer la limite des 4 Go de mémoire vive. En termes de performances pures, bien entendu, il y a amélioration, mais pas significative. Ce type de processeur est idéal pour les gros consommateurs de mémoire et de calculs. Mais, qu'on le veuille ou non, le processeur 64 devient la puce de référence. Notez qu'à la sortie des premiers G5, Apple mettait en avant le 64 comme

argument. Aujourd'hui, le constructeur le mentionne à peine...

Le système 64

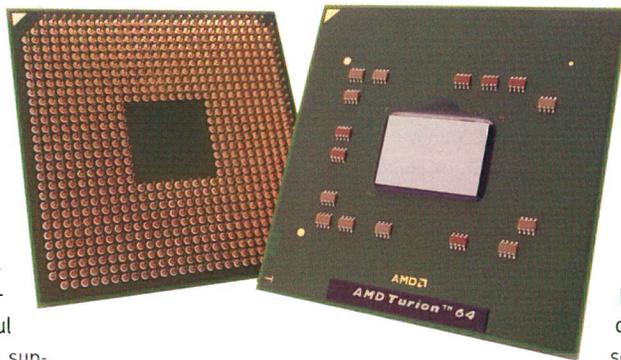
Le marché attendait une annonce forte de Microsoft autour du 64 et Windows. Les principales distributions Linux existent déjà en version 64 mais cela n'a jamais tiré le marché, hor-

mis peut-être côté serveur. Les principaux logiciels d'entreprise disposent aussi d'une édition 64.

Microsoft a, au printemps dernier, réveillé le marché en annonçant les éditions 64 de Windows XP et Windows Server 2003. Windows Vista sera, à n'en pas douter, disponible en 64-bit. Ces versions supportent les processeurs AMD et Intel. Cette annonce devrait booster le marché et favoriser les portages d'applications.

Les logiciels à la traîne

Voilà bien le problème. Aujourd'hui, mis à part côté serveur et quelques grosses applications de calcul et de gestion, les applications 64 n'existent pas. Avouons que l'intérêt de disposer d'un OpenOffice ou MS Office 64-bit semble bien limité pour 99 % des utilisateurs. C'est là que le marché est pervers. Car, en installant par défaut des processeurs 64 dans les ordinateurs, les utilisateurs, tôt ou tard, doivent s'équiper en système et applications 64. Or, rien n'est prêt. Outre le fait d'installer un système 64, il faut utiliser des pilotes 64. Et dans ce domaine, le travail à accomplir est énorme, il faut tout réécrire. C'est pour cela



soyez prêts pour le futur !

Bien entendu, qui dit nouvelle architecture, nouvel adressage, dit nouvelles applications. Pour pouvoir réellement tirer profit du 64-bit, il faut le matériel adapté, mais surtout les applications et pilotes adaptés et recompilés, ou entièrement réécrits. Si pour les grosses applications d'entreprises et les logiciels scientifiques et celles nécessitant d'importants volumes de mémoire (compression, décompression, 3D, etc.) l'intérêt du 64 est réel, pour le grand public il demeure encore limité.

La généralisation du 64 auprès du public et des entreprises passe par une mise à niveau du parc informatique et des applications. Le système doit lui aussi être 64. Microsoft a annoncé début 2005 sa stratégie autour de Windows 64, Linux possède déjà plusieurs distributions adaptées et MacOS X fonctionne partiellement en mode 64.

Au-delà du 64, le développeur (et encore plus les éditeurs) est confronté à une délicate transition technique et technologique. Le 64 ne représente qu'une partie des évolutions des

processeurs. Le HyperThreading et surtout le multicœur deviennent des éléments vitaux des nouvelles générations de puces. Revers de la médaille, adapter ou concevoir une application optimisée multicœur nécessite une compétence en programmation parallèle de haut niveau... Quand on sait que le multi threading commence juste à être utilisé optimalement (là aussi à condition d'être bon), la parallélisation du code n'est pas pour demain.

■ François Tonic

qu'à l'heure actuelle, le marché grand public du 64 n'est pas viable. Et en entreprise, à part le serveur et quelques postes très précis, le 64 n'a pas non plus grand intérêt.

Techniquement, le portage d'une application 32 en 64 ne pose pas de soucis (à condition de disposer d'un code source propre et structuré).

Quelles architectures prendre en compte : Intel, AMD, les deux ? Théoriquement, les deux solutions sont compatibles entre elles. Mais les incertitudes autour d'Intel demeurent, AMD étant en avance. Mieux vaut, dans l'immédiat, disposer de deux binaires.

Les outils pour le 64 existent. Les compilateurs sont disponibles depuis longtemps. Plusieurs JVM existent et par défaut, .Net 2.0 inclut une version 64 du framework. Tout est fait pour aider le développeur à porter rapidement son code. Il doit se tenir prêt à travailler en 64, même si pour cela, il doit acheter au moins deux machines : une AMD, une Intel, voire les

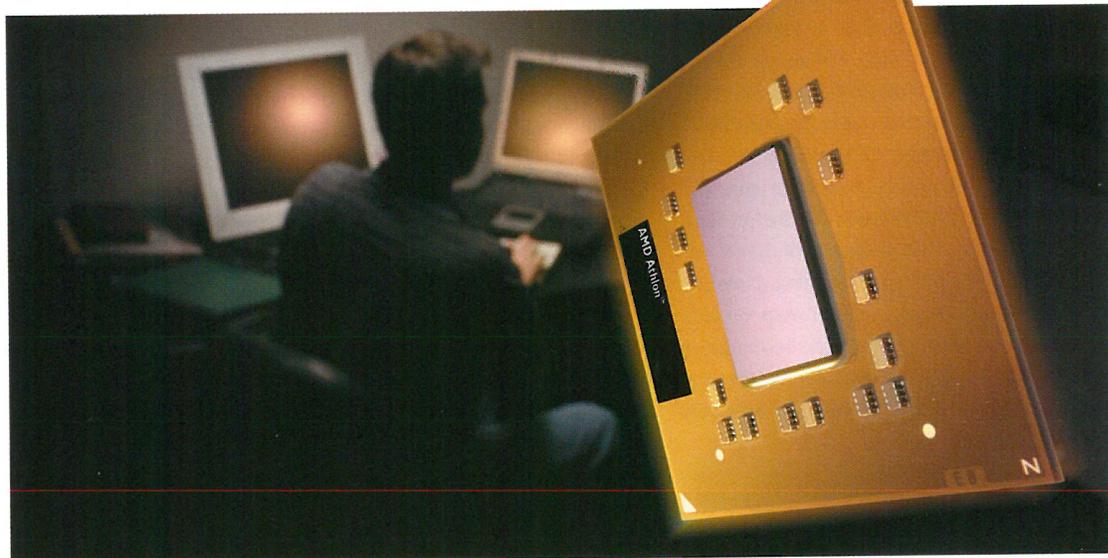
déclinaisons portables. Et en installant les différents systèmes. Tout en continuant à supporter les processeurs 32-bit...

Car, n'oublions pas que les applications 32 (et donc les processeurs actuels) demeureront encore des années et qu'il faudra gérer, maintenir, et porter, le parc applicatif existant. La migration d'une plate-forme à un autre ne va pas de soi. Car, il faut migrer l'ensemble du système informatique : système, machine, applications. Bien entendu, les applications 32 peuvent fonctionner en environnement 64, mais avec des restrictions (ex. : les applications fortement liées au matériel deviendront inutilisables).

Où est le véritable défi ?

Le 64 n'est finalement qu'un des aspects du processeur. Sans doute le moins important. Le véritable défi se situe désormais autour du multicœur, avec le fameux dual-core. Les gains de performances se situent là. Cependant, là encore, un problème de logiciels se fait jour. Pour exploiter le dual-core, il faut des applications parallélisées et parfaitement multithreadées, capables de fonctionner en parallèle. Or, cette technique de programmation exige une grande maîtrise du développement processeur et une réécriture du code.

■ F.T.



Comment bien tirer parti des processeurs double cœur



Placer deux processeurs sur la même puce pour accélérer les performances est une bonne idée. Malheureusement, seules les applications multi threads et parallélisées peuvent en profiter. Aux développeurs d'effectuer tout le travail d'optimisation du code.

Pendant longtemps, la solution de facilité a simplement consisté à augmenter la fréquence d'horloge. Malheureusement, même si les limites technologiques sont encore loin d'être atteintes, cette augmentation devient de plus en plus difficile à réaliser. Les fondeurs ont donc choisi une autre voie qui consiste, non plus à exécuter les instructions plus vite, mais à en exécuter plusieurs en même temps. Après l'HyperThreading d'Intel, la nouvelle étape consiste à placer deux processeurs dans une même puce. Cette technique a été appliquée par AMD avec ses Athlon X2, Intel avec ses Pentium D et enfin IBM avec son PowerPC 970 MP.

Découper les applications en threads

L'avantage d'une telle technologie est évident lorsque plusieurs programmes indépendants tournent au même moment. Un des deux cœurs va traiter une application bureautique, par exemple, pendant que l'autre s'occupera des téléchargements, de l'antivirus et du firewall. Maintenant, s'il s'agit d'accélérer l'exécution d'un seul programme, le problème est plus épineux. Si le programme ne comporte qu'un seul thread, seul un cœur pourra s'en charger et le bénéfice obtenu sera nul. Il est donc impératif de réécrire les programmes, afin de les diviser en un certain nombre de threads indépendants. C'est plus facile à dire qu'à faire. Dans certains cas, le découpage est impossible. Chaque séquence de code a besoin des résultats de la précédente pour s'exécuter et il n'y a rien à

faire. Dans d'autres cas, le découpage est facile. Dans la création d'une image par exemple, chaque partie peut être construite indépendamment des autres. Il y a enfin le cas général, qui représente la majorité des applications, où le découpage est possible, mais généralement difficile. Il faudra alors, dans un premier temps, déterminer quelle est la partie de l'application qui est la plus grande consommatrice de ressources. C'est surtout sur cette partie qu'il faudra travailler, l'optimisation du reste du programme n'offrant que peu de bénéfices. Il ne faut pas non plus chercher à créer le maximum de threads, leur création prend du temps et il faut que le jeu en vaille la chandelle. Le parallélisme dans l'exécution des tâches n'est pas quelque chose de totalement nouveau. Les développeurs qui travaillent sur les supercalculateurs et sur les systèmes multiprocesseurs sont habitués à traiter ce type de problème depuis longtemps. Ils ont, au fil du temps, créé des ensembles de modèles qui représentent des solutions à certains problèmes et plutôt que de réinventer la roue, il pourra être intéressant de s'y référer.

Utiliser des API spécialisées

Ces travaux qui sont souvent très « universitaires », sont décrits dans de très nombreux ouvrages, mais les développeurs en C/C++ ou Fortran, pourront également se tourner vers OpenMP, une API spécialisée dans la programmation en parallèle en mémoire partagée (www.openmp.org). Open MP supporte les plates-formes Unix et Windows, et est accepté par de nombreux compilateurs, tels que Visual Studio par exemple. Il existe d'autres API, plus

difficiles à utiliser et qui nécessitent plus de réécriture du code. Parmi celles-ci, on peut citer thWin32 pour Windows, Pthread pour Unix (POSIX). Si les outils et les modèles sont d'une aide précieuse, ils sont loin de résoudre tous les problèmes. Le premier étant le débogage de l'application. Il n'est possible de débogger qu'un seul thread à la fois, et de plus, la distribution des threads entre les processeurs est laissée à l'appréciation du système d'exploitation. Il devient alors extrêmement difficile de prédire le fonctionnement du programme, car on ne sait plus qui s'exécute et quand. Une solution consiste à utiliser des points de synchro qui permettent de repasser en mode séquentiel pour savoir où en sont les différents threads. En plus du problème lié au parallélisme, il faut éviter bien des pièges, tels que les locks sur les variables, ou tous les problèmes liés à l'utilisation d'un même espace mémoire par différents threads.

Il existe actuellement une pression importante des clients sur les éditeurs pour que les processeurs double cœurs soient pris en compte dans les produits. Cela signifie que beaucoup d'applications devront être réécrites, mais ce gros travail ne devra être fait qu'une fois. Les applications ainsi optimisées pourront ensuite tourner sans problème sur des processeurs multicœurs des prochaines générations (la compatibilité entre les processeurs multicore Intel et AMD sera-t-elle totalement assurée ? Ou faudra-t-il deux binaires par plate-forme matérielle ? ndr).

■ Alain KERHUEL

.NET, Java et le 64-bit

Intel, Microsoft, Sun, les distributions Linux... ont tous à proposer un système d'exploitation, un processeur, un logiciel 64-bit. L'offre est finalement pléthorique, sans qu'on puisse toujours en percevoir le réel intérêt. Quels sont en fait les apports du 64-bit ? Comment cette technologie est-elle supportée par les deux principales plates-formes d'entreprise que sont .NET et Java / J2EE.

Pour pouvoir fonctionner en mode 64-bit, une application .NET ou Java doit s'appuyer sur un environnement d'exécution spécifique 64-bit, lui-même nécessitant un système d'exploitation 64-bit, lequel requiert enfin un processeur 64-bit. On le voit, le fonctionnement en mode 64-bit nécessite le passage de toute la chaîne au 64-bit.

IL et bytecode : une portabilité maximum

Bonne nouvelle cependant, .NET et Java reposent tous deux sur un langage intermédiaire : IL (Intermediate Language) pour .NET, et le bytecode pour Java. Sur les deux plates-formes, le langage intermédiaire est exécuté au sein d'un environnement d'exécution : le CLR

(Common Language Runtime) pour .NET, et la JVM (Java Virtual Machine) pour Java. Ainsi, pour ces deux environnements, la représentation en mémoire vive des types simples reste la même, que l'on soit en 32-bit ou en 64-bit. Seule la représentation en mémoire vive des références (pointeurs) change, passant de 32-bit à 64-bit, pour profiter de la plus grande capacité d'adressage spécifique du 64-bit. Ces propriétés permettent d'exécuter le même code intermédiaire, soit en mode 32-bit, soit en mode 64-bit, sans modifications, et donc sans recompilation du code source. Il suffit juste d'opter pour une version 64-bit du CLR, ou une version 64-bit de la JVM. Dans la pratique, il subsiste quelques problèmes à partir du moment où l'application utilise du code non-

managé, pour .NET, ou du code natif interfacé via JNI (Java Native Interface) pour Java.

Les systèmes d'exploitation 64-bit

Dans la famille Windows, le support du 64-bit est disponible au sein de Windows Server 2003, avec le support à la fois de l'Itanium et des processeurs x64. Windows XP x64 supporte également le 64-bit, uniquement pour les processeurs x64, comme son nom l'indique. Linux n'est pas en reste et propose des versions 64-bit spécifiques. Quant à Solaris, le support du 64-bit ne date pas d'hier, puisqu'il remonte à Solaris 8 (la version courante est Solaris 10). Mac OS X 10.4 est partiellement 64-bit (ndlr).

Les environnements d'exécution

.NET 2.0 disposera, à sa sortie, d'une version 64-bit du CLR. Quant à Java, les premières versions 64-bit de la JVM datent du JDK 1.4, mais uniquement pour le processeur Sparc sous Solaris. Le JDK 1.4.2 apporte le support de l'Itanium sous Windows 64-bit. Quant à lui, le JDK 1.5 fait machine arrière, en renonçant au support de l'Itanium, pour lui préférer les processeurs x86 sous Windows 64-bit et Linux 64-bit.

Tout n'est pas si simple

Une application compilée avec .NET 1.x ne peut pas tourner directement en 64-bit. En fait, même sur un Windows 64-bit, l'application tourne en mode 32-bit au sein de WOW64 (Windows On Windows 64). En revanche, sauf contre-indication explicite, une application .NET 2.0 peut tourner soit en mode 32-bit, soit en mode 64-bit, selon les possibilités offertes. Avec le code non-managé, la situation se complique un peu. Si l'application accède à des DLL ou des composants COM 32-bit, l'ensemble

.Net 2 aime bien le 64-bit

Comme dans la version 32-bit de .Net, la déclinaison 64 possède une CLR, un Framework Class Libraries et un kit de développement. Basiquement, le code .Net est rapidement portable de 32 en 64. Tout le code appelant des DLL, du COM InProc Servers... nécessite une phase de réécriture et d'adaptation. En effet, on ne peut pas invoquer une DLL 32 à partir d'un code 64, et vice versa. Cependant, on utilise les mêmes outils de développement pour coder : Visual Studio 2005. VS 2005 propose trois options : seulement MSIL (option par défaut), mode 32, mode 64. VS 2005 permet de coder, compiler et déployer des applications 32 et 64. Le SDK .net 64 comprend les compilateurs en ligne C# et VB.Net et un debug. J# ne fait pas partie des langages supportés en 64-bit. Pour DirectX, une

version 64 est en cours de conception, mais aucune date de disponibilité n'est encore annoncée. Si vous utilisez tout de même du code .Net 1.x sur un Windows 64, par défaut, il tournerait en mode WoW, donc non natif. Bien entendu, il faut disposer d'une machine et d'un Windows 64 (2003 Server et XP). Les applications 32 peuvent fonctionner (à quelques exceptions) en environnement 64 via un WoW64 (Windows on Windows). Mono est depuis plus d'un an disponible sur différents processeurs 64 grâce au portage de Mono JIT : AMD 64 (uniquement Linux), Sparc (sous Solaris), IA64 (uniquement Linux pour le moment). Actuellement, aucun support Windows 64 n'est annoncé. Il faudra attendre la disponibilité effective de ces systèmes et du portage de Mono. F.T.

de l'application doit obligatoirement tourner en mode 32-bit, au sein de WOW64. Pour fonctionner en mode 64-bit, les DLL doivent être recompilées. De plus, le code 'unsafe' doit éventuellement être modifié pour tenir compte du changement de taille des pointeurs. Le framework .NET fournit une fonction qui permet de connaître la taille des pointeurs. Il n'est donc pas très compliqué d'écrire du code portable à la fois en 32-bit et en 64-bit.

En ce qui concerne Java, on retrouve un peu les mêmes problèmes avec le code natif. Mais la situation change quelque peu selon le système d'exploitation 64-bit. De manière générale, tout code natif interfacé via JNI (Java Native Interface) doit idéalement être porté en 64-bit. Le code Java client doit également tenir compte du changement de taille des pointeurs. Il s'agit cependant de problèmes très relatifs : l'usage de code natif est assez rare en Java, fortement découragé par l'aridité de JNI.

Note de la rédaction : en Java 64, le modèle du data-type est le LP64 et non le ILP32. Il faut faire attention dans la migration à la taille des long et pointer. Une des solutions est d'utiliser lint afin de vérifier le code si celui-ci est à la fois 32 et 64. Vous pouvez spécifier de vérifier le mode 64 et de générer les avertissements 64 (-errchk=longptr64). L'une des principales difficultés dans une conversion 32 - 64 d'un projet Java se focalise sur la taille du int. Dans une conversion, il faut savoir que seuls les types long et les types de pointer changent en taille. Les int demeurent en taille 32-bit. Voir notamment : <http://developers.sun.com/prodtech/cc/articles/ILP32toLP64Issues.html>

Le 64-bit en perspective

Le 64-bit permet de briser la fameuse barrière des 4 Go et d'accéder ainsi à une capacité

d'adressage beaucoup plus importante. Les SGBD sont naturellement les premiers bénéficiaires, et peuvent ainsi gérer des bases de données de très grande taille, en profitant de buffers gigantesques. Il y a également fort à parier que les applications d'entreprise nécessiteront de plus en plus le franchissement de cette limite de 4 Go. De telles capacités de mémoire sont nécessaires pour effectuer des calculs numériques sur des données de très grande taille, ou encore pour gérer des caches sur des applications d'entreprise hautement scalables. Cependant, dans ce dernier cas, on peut se demander si l'implémentation de caches distribués n'est pas une approche plus

scalable et plus apte à traiter le fail-over. Dans ce cas, la fonction de cache est implémentée par une batterie de processus, chacun n'ayant pas forcément besoin de dépasser la fameuse limite des 4 Go. Quant au garbage collector de .NET et Java, la pratique montrera s'ils sont capables de gérer efficacement de très grands espaces mémoire.



■ **Nicolas Dasriaux**
Directeur technique de Neoxia
nicolas.dasriaux@neoxia.com



Windows 64 : le nouveau marché ?

Avec la sortie de Windows XP et Server 2003 en édition 64-bit, le marché devrait dans les 18 mois se dynamiser. Outre l'aspect matériel, il faudra mettre à jour une importante partie de son parc de progiciels et applicatifs. Cela signifie migration, portage des applications 32 en 64 si on souhaite réellement tirer parti du système et du processeur. Si pour Java, il suffit d'une JVM 64 et de tests de vérifications, peu de changements à l'horizon. Pour Windows, la situation n'est pas identique, surtout pour les applications Win32, certaines ne fonctionneront pas (même en mode WoW). Cependant, si votre code est propre, peu de soucis hormis dans le cas d'applications distribuées utilisant du RPC avec ou sans DCOM. Dans tous les cas, testez l'application portée ou fonctionnant en WoW.

Voir notamment : <http://msdn.microsoft.com/visualc/using/building/64bit/default.aspx>

Suggestion d'étapes :

- › identifier le code 32 pouvant poser problème ou à modifier obligatoirement
- › compiler le code 32 avec le compilateur 64 pour lister les problèmes
- › porter l'intégralité de l'application en 64-bit. Ne jamais porter des sections !
- › Identifier les technologies obsolètes ou non, porter en 64 pour trouver des substituts.
- › Séparer les sources et build 64 de l'arbre 32

Avant tout portage, migration, mettez-vous à niveau : formez-vous, étudiez les changements de programmation.

F.T.

Au service du développeur

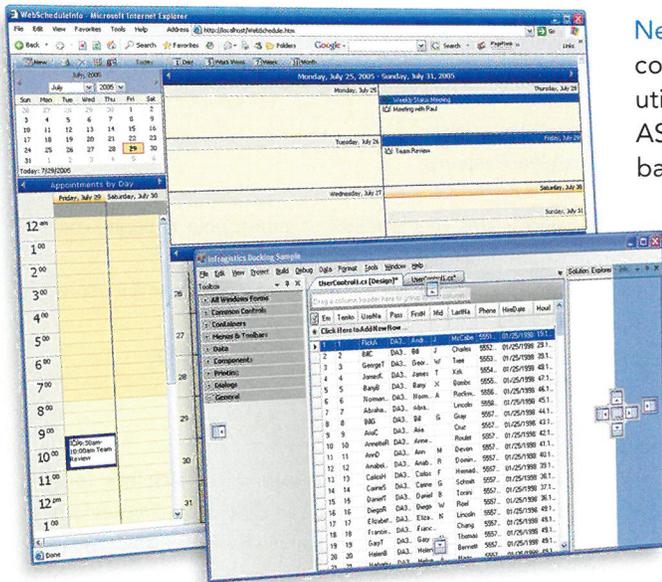
- L'actu au quotidien
- La news letter
- L'emploi, les forums
- Les archives de Programmez
- L'abonnement en ligne

www.programmez.com

Nouvelle version!

NetAdvantage™ 2005 Vol. 3

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



NetAdvantage 2005 Volume 3 est la boîte à outils complète pour créer rapidement l'interface utilisateur de vos applications Windows Forms, ASP.NET et Tablet PC. A travers nos UIElements™ basés sur notre Presentation Layer Framework™, nous fournissons la capacité d'ajouter facilement les derniers looks and feels style Microsoft ainsi qu'une extensibilité sans parallèle à vos applications .NET.

Nos presets style QuickDesign et support continu de la section 508 (accessibilité) à travers nos éléments continuent de faire de NetAdvantage le numéro un des ventes des boîtes à outils pour interface utilisateur sur le marché!

Prennez connaissance de la puissance d'AJAX en utilisant notre WebGrid et améliorez ainsi la performance de votre application.

Avec trois nouvelles mises à jour par an, Infragistics continue de simplifier le développement .NET. Toujours innovateur, Infragistics livre des outils de développement vous permettant d'augmenter l'extensibilité de Visual Studio 2005!

Cliquez ici afin de visionner une liste complète des nouvelles caractéristiques et éléments améliorés dans cette nouvelle version: www.infragistics.com/whatsnew



TestAdvantage™ 2005

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



Training

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

www.infragistics.com

sales-europe@infragistics.com

N° Vert 0800 667 307

Optimized for

Microsoft
Visual Studio.net

Infragistics®
Powering The Presentation Layer

Interopérabilité .NET - PHP

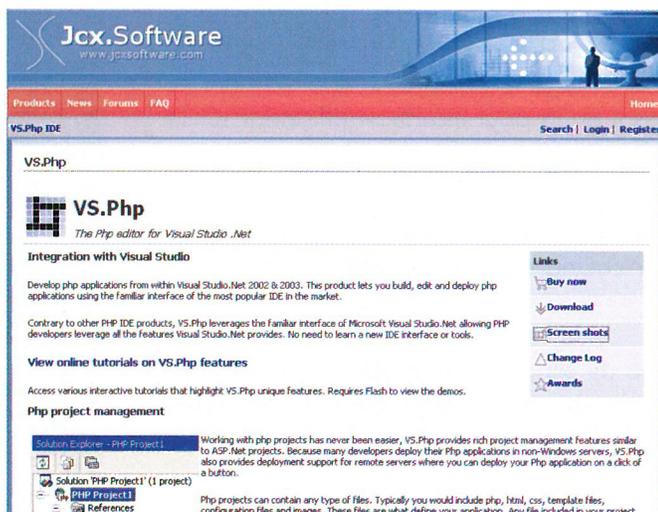
2^e partie

Continuons notre découverte de l'interopérabilité .NET / PHP.

Comme vous le remarquerez au sein du tableau parcouru préalablement, j'ai volontairement référencé l'ensemble des éléments applicables à la relation d'interopérabilité entre ces deux langages. Vous comprendrez cependant que j'éviterai de vous faire un copier-coller brut des informations, préférant vous relayer vers la partie précédente « De .NET à PHP » et ce, particulièrement sur des points tels que PHP# ...

1. Couplage Fort : Mono 1.0 - NET pour Unix et Linux

Bien que Mono ne soit pas tout à fait classable dans un article tel que l'interopérabilité et plus particulièrement dans le couplage Fort, je me suis permis de vous le présenter succinctement sous cette rubrique. Merci aux avertis pour leur compréhension au sujet de cette prise de liberté !



Mono est un projet qui a pour but de développer un framework .NET alternatif complet, disponible pour les plates-formes Linux et Windows (Mac et BSD par extension). Il a été lancé par l'équipe Ximian, le 9 juillet 2001 et géré par Miguel de Icaza, vice-président du développement chez Novell, qui a créé Mono (et co-fondateur Ximian). Mono 2.0 inclura un support natif des WinForms et du Visual Basic.Net, ainsi qu'une mise à jour des API .Net. Une application .Net créée sous Linux ou Windows tournera indifféremment sur l'un ou l'autre OS.

Les sites indispensables :

- <http://www.go-mono.org>
- <http://www.gotmono.com>

2. Couplage Lâche : XML-HTTP

Le protocole XML-HTTP est un protocole permettant d'invoquer un objet

distant en fournissant les paramètres nécessaires à l'appel dans un message au format XML et en utilisant le moyen de communication HTTP. La réponse est également renvoyée dans un format XML.

Voici les classes au sein du .NET Framework facilitant l'utilisation de ce protocole :

- System.Net.HttpWebRequest
- System.Net.HttpWebResponse

Quelques liens nécessaires vers la MSDN et autres, pour la bonne compréhension de ce protocole :

- <http://msdn.microsoft.com/library/fre/default.asp?url=/library/FRE/cpreft/html/frlrSystemNet.asp>
- <http://www.dotnet247.com/247reference/msgs/25/125779.aspx>
- <http://www.xmlforasp.net>

3. Couplage Lâche : Services Web

Par Services Web, nous entendons de manière assez simpliste toute application web pouvant interagir dynamiquement avec d'autres programmes en utilisant des protocoles d'échanges basés sur XML comme SOAP, XML-RPC ou XMLP.

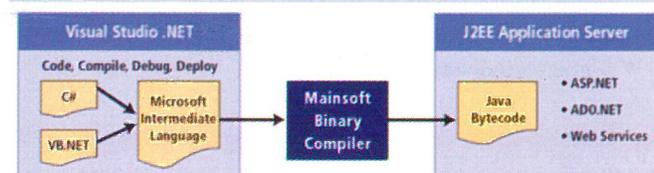
Nous retrouvons ainsi au sein du .NET Framework l'espace de nom « System.Web.Services », celui-ci est largement décrit au travers de la MSDN. En voici les liens :

- <http://msdn.microsoft.com/library/fre/default.asp?url=/library/FRE/cpreft/html/frlrSystemWebServicesProtocols.asp>
- <http://www.microsoft.com/france/msdn/technos/webservices.mspx>
- <http://www.xmlforasp.net/CodeBank.aspx?codeType=webservices>

Voyons ainsi un exemple caractéristique de flux de données, par voie XML, d'une application PHP vers une solution ASP.NET. Nous n'entrons pas pour autant dans la gestion via Webservice.

Observons préalablement le code PHP permettant d'exposer le flux XML dynamique à notre application ASP.NET qui se chargera de le consommer.

```
<script>
<?
// vérification de la présence du fichier
if (file_exists('rss.xml')) {
```



```
// simplexml_load_file convertit un fichier XML en objet > PHP5
$xml = simplexml_load_file('rss.xml');

// vérification de la présence de la variable date dans le tableau get
if(isset($_GET['date'])) {

    // c'est le cas, création du document
    header("Content-Type: text/xml");
    $docxml = '<?xml version="1.0" encoding="UTF-8"?><news>';

    //on vérifie la date de chaque news et on les affiche
    si la date correspond à la var date (get)
    for ($i=0; $i<4; $i++) {
        if($xml->channel->actu[$i]->date==$_GET
        ['date']) {
            $docxml .= '<actu>';
            $docxml .= '<titre>'.$xml->
            channel->actu[$i]->titre.</titre>';
            $docxml .= '<desc>'.$xml->
            channel->actu[$i]->desc."</desc>";
            $docxml .= '<src url="">'.$xml->
            channel->actu[$i]->src['url'].>
            '.$xml->channel->actu[$i]->src.</src>';
            $docxml .= '</actu>';
        }
    }
    $docxml .= '</news>';
    echo $docxml;
} else {
    exit('Erreur !');
}
?>
</script?>
```

Vous l'aurez compris, je ne fais que taper dans un fichier xml dont vous retrouvez la structure ci-dessous à partir d'une requête GET sur la « date » ! Rien de bien complexe, mais c'est efficace !

```
<channel>

  <actu>
  <date>20/11/2004</date>
  <titre>actu 1</titre>
  <desc>resume de l'actu 1</desc>
  <src url="http://www.google.fr/">Google</src>
  </actu>
  ...

</channel>
```

Observons tranquillement le code que nous devons implémenter côté

ASP.NET afin de faire interopérer ces deux structures !

D'abord la partie HTML, je glisse tout simplement un WebControl ASP.NET Calendar ainsi qu'une DataGrid. Nous allons, en fonction du choix de la date dans le Calendar, engendrer un réaffichage dans la DataGrid des news du jour en question. News qui nous proviennent d'une plate-forme tournant en PHP !

```
<form id="Form1" method="post" runat="server">
  <asp:Calendar id=Calendar1 style="Z-INDEX: 101; LEFT: 32px;
  POSITION: absolute; TOP: 64px" runat="server"></asp:Calendar>
  <asp:DataGrid id=DataGrid1 style="Z-INDEX: 103; LEFT: 304px;
  POSITION: absolute; TOP: 72px" runat="server" Width="376px">
  </asp:DataGrid>
</form>
```

J'ai volontairement développé cette partie en VB.NET en vue de faciliter sa compréhension par une majorité de lecteurs, ce qui ne veut pas dire que le VB.NET est plus simple que le C#, ceci est un lourd débat dans lequel nous pourrions passer trop de temps !

```
'gestion sur l'évènement SelectionChanged de l'objet Calendar
Private Sub Calendar1_SelectionChanged(ByVal sender As System.
Object,
    ByVal e As System.EventArgs) Handles Calendar1.Selection
    Changed

    'Gestion légère de l'erreur au travers de la méthode Try ... Catch
    ... End Try
    ' nous retrouvons également l'approche Try ... Catch au sein du
    PHP5, celui-ci est
    ' cependant moins complet
    Try

        'instanciation d'un objet de type DataSet
        Dim newsVSDs As New DataSet

        'utilisation de la méthode ReadXML de notre objet DataSet en
        vue de charger
        ' dynamiquement le flux xml au travers de la requête GET en lui
        transmettant
        ' la variable Date reprise directement du calendar
        newsVSDs.ReadXml("http://localhost/php5/DemoXMLHTTP.php?
        date=" &
        Calendar1.SelectedDate.ToShortDateString, XmlRead
        Mode.Auto)

        'définition de la source de l'objet DataGrid par le conteneur
        Tables(0)
        ' du DataSet qui s'avère contenir les news
        DataGrid1.DataSource = newsVSDs.Tables(0)

        'liaison entre la source et l'objet DataGrid (DataBind)
        DataGrid1.DataBind()
```

```

'affichage de l'objet DataGrid (passage de sa propriété visible
à true)
    DataGrid1.Visible = True

Catch

'retour d'un message d'erreur, je l'ai volontairement voulu non
complet,
'rend non visible l'objet DataGrid
    DataGrid1.Visible = False

End Try
End Sub

```

Le code se passe de commentaires, il l'est déjà suffisamment ;) !

Solutions exotiques émergentes !

1. Add-In J2EE VS.NET MainSoft

Ce concept part de l'approche de convertir une simple application ASP.NET en servlet Java debuggable, le tout devant ainsi être déployé sur une plate-forme Tomcat.

Voici un lien de mon blog relatant un peu plus l'approche que j'ai eu l'occasion de tester et ma conclusion serait de dire qu'elle est des plus intéressantes, d'autant plus que l'Add-In en question s'intègre parfaitement au sein de votre VS.NET 2003 :

<http://blogs.developpeur.org/redo/archive/2004/07/01/1945.aspx>

Pour faire bref, vous compilez votre code au sein de votre Assembly en IL et l'Add-In se chargera de recompiler votre Assembly IL en servlet Java. Que du Bonheur !!!

2. Add-In VS.PHP (Intégration à VS.NET) & VS.PHP Debugger

Contrairement aux autres IDE du marché, VS.PHP fournit l'interface familière de Visual Studio.net aux développeurs PHP, avec l'accès à l'ensemble des fonctionnalités. Il n'est pas utile d'apprendre un nouvel outil. Les sites :

<http://www.jcxsoftware.com/jcx.corporate6/modules/wfsection/index.php?category=12>

<http://www.jcxsoftware.com/jcx.corporate6/modules/wfsection/article.php?articleid=35&PHPSESSID=4eddb2285a1449aef8ab89c6d13e44>

Conclusion

Au travers de cet article centré sur les concepts et préceptes de l'interopérabilité, j'espère vous avoir fait toucher du doigt le bonheur et la solution, ou tout du moins un début de solution à certaines de vos problématiques !

Mais si de l'interopérabilité, il ne fallait retenir que quelques bribes, je vous proposerais :

- Évolution constante de PHP
- Sur extensions et connecteurs
- Sur le support des Web Services
- .NET Interop en mode Natif
- Évolution du projet Mono
- Simplification de la communication entre systèmes hétérogènes, éliminer les complexités et les diversités technologiques.

Et surtout ...

« Exploitez le meilleur des deux technologies »

Enfin, je conclurai sur le fait que la globalité de cette approche est applicable à d'autres plates-formes et langages. Comme vous l'aurez compris !

Références & Liens :

PHP5

- <http://www.php.net/downloads.php>

Manuel PHP4 – PHP5

- <http://www.php.net/manual/fr/index.php>

- <http://www.php.net/download-docs.php>

Install de php5 sous IIS

- http://www.asp-php.net/ressources/bouts_de_code.php?codid=60

Projet Mono

- <http://www.go-mono.org>

Faire un agrégateur en PHP 5

- http://www.asp-php.net/scripts/asp-php/php5-agregateur_rss.php?page=2

Communautés

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

- <http://www.codes-sources.com>

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

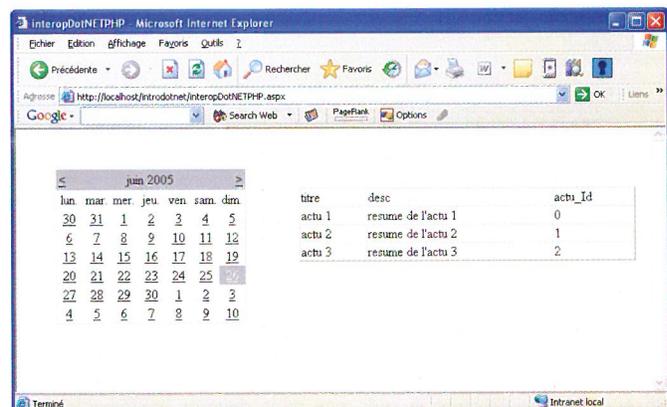


■ Grégory Renard

Directeur Développement Wygwam sarl www.wygwam.com

Microsoft MVP Visual Developer ASP - ASP.NET, Microsoft CodeZone Community Member

Blog : <http://blogs.developpeur.org/redo> - Auteur de nombreux articles .NET sur www.asp-php.net (Alias Rédo)



XFire : développer des services web

Les Web Services sont incontournables. Après quelques années de gestation, de nouvelles implémentations légères, rapides et open source, développées en Java, apparaissent. Le projet Xfire, de la communauté CodeHaus propose une solution élégante qui pourrait connaître le succès.

Le service web regroupe un ensemble de normes définies pour l'échange de données entre applications. A l'origine, le concept utilisait des protocoles de communication Web comme HTTP (ou FTP, SMTP etc.) et XML pour l'écriture des messages. Au fur et à mesure, plusieurs normes sont venues enrichir l'idée de base :

SOAP : pour structurer les messages XML échangés

WSDL : Web Service Description Language : pour décrire les services web, leur opération, les messages échangés, le type de données et les 'bindings' (protocoles et encodages) utilisés.

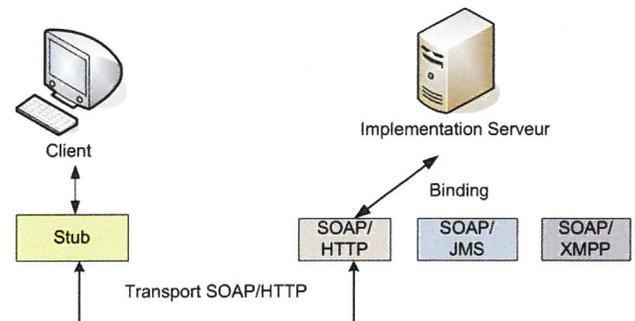
UDDI : norme pour définir et accéder (par l'intermédiaire de Web services) des annuaires pouvant recenser, entre autres, des Web Services. Actuellement, le nombre de normes évoluées gravitant autour des Web Services prend des proportions démesurées, au point que certains utilisateurs remettent en question le modèle et prônent un retour aux sources.

Présentation de XFire

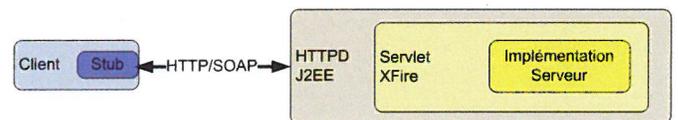
XFire est un framework implémentant les différentes normes pour le développement de Web Services en Java. XFire se présente comme une alternative légère, rapide et open source (licence similaire à BSD) aux solutions existantes telles qu'Apache Axis, ou les produits commerciaux équivalents. Actuellement, le produit n'a pas encore atteint un degré de maturité digne d'Apache Axis son principal concurrent, mais le code évolue constamment pour atteindre la stabilité d'une version 1.0.

XFire s'appuie sur StAX, une API de Pull Parsing XML qui a pour principe de consommer les éléments un à un dans un flux de données entrant (à la demande de l'application), contrairement à l'API SAX qui génère des événements qui doivent être immédiatement traités. L'application peut donc mieux gérer et contrôler ses ressources (principalement au niveau de la mémoire) et traiter les informations utiles, contrairement aux frameworks reposant sur SAX qui 'subissent' les demandes de traitement de données XML. Il est à noter d'ailleurs, que cette approche qui s'appuie sur une implémentation StAX, est également le choix qui a été retenu pour le développement de la seconde génération d'Apache Axis (encore instable).

XFire a été développé dans un souci de simplicité et de légèreté afin de permettre de l'embarquer facilement dans des serveurs ou des « conteneurs » tels que Spring. Il s'appuie naturellement sur les techniques d'introspection de Java pour découvrir les méthodes des classes à partager sous forme de Web Services ainsi que les messages qui seront échangés. De plus, XFire propose une prise en compte des annotations introduites dans le JDK 5 avec le support actuellement partiel des attributs définis via la norme JSR181. Cette norme propose un ensemble d'attributs pour spécifier à même le code Java, le comportement de certaines classes et méthodes.



Détail amusant, XFire permet d'utiliser comme transport le protocole XMPP (Jabber), plus communément utilisé par la messagerie instantanée et qui s'appuie sur l'échange de messages XML. Actuellement, le développement s'est particulièrement penché sur la partie serveur, laissant de côté la partie cliente qui en est au stade expérimental.



A l'instar d'Apache Axis, XFire s'appuie sur une Servlet Java lorsqu'il utilise le transport HTTP. Cependant, XFire est bien plus souple dans sa configuration. L'utilisateur pourra, soit utiliser une implémentation de Servlet fournie par XFire, soit étendre une Servlet fournie par XFire et utiliser l'API de XFire pour enregistrer les différentes classes Java comme des services web.

Déploiement d'un service

Exemple simple de service web

Nous allons illustrer l'utilisation d'XFire à l'aide d'un exemple d'application simple. Nous proposons de développer un service de gestion de personne avec une classe de Service et une classe pour l'entité 'Personne'.

```
public class Personne implements Serializable {

    private String nom;
    private String prenom;
    private int age;

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getNom() {
```

```

        return nom;
    }
    public void setNom(String nom) {
        this.nom = nom;
    }
    public String getPrenom() {
        return prenom;
    }
    public void setPrenom(String prenom) {
        this.prenom = prenom;
    }
}

```

```

public class PersonneService implements PersonneServiceInterface
{
    // code d'initialisation de la liste des Personnes
    // ...

    public void addPersonne( Personne p )
    {
        m_personnelList.add( p );
    }

    public Personne[] getAllPersonnes()
    {
        return (Personne[])m_personnelList.toArray(new
        Personne[0]);
    }

    public Personne getPersonne( int indice )
    {
        return (Personne)m_personnelList.get( indice );
    }
}

```

Dans le cas du déploiement pour l'utilisation d'un transport HTTP, nous utilisons la Servlet fournie par XFire, que nous référençons dans le fichier de configuration web.xml de l'application web.

```

<?xml version="1.0" encoding="ISO-8859-1" ?>

<!DOCTYPE web-app
    PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>

<servlet>
<servlet-name>XFireServlet</servlet-name>
<display-name>XFire Servlet</display-name>
<servlet-class>
org.codehaus.xfire.transport.http.XfireConfigurableServlet
</servlet-class>
</servlet>

<servlet-mapping>

```

```

<servlet-name>XFireServlet</servlet-name>
<url-pattern>/services/*</url-pattern>
</servlet-mapping>
</web-app>

```

La Servlet que nous utilisons s'appuie sur un fichier de configuration 'services.xml' qui doit être situé dans le répertoire /META-INF/xfire/ du class-path.

```

<xfire>

<services>
<service>
<!-- Nom du service -->
<name>PersonneService</name>

<!-- L'espace de nommage du webservice -->
<namespace>urn:objdirect:PersonneService</namespace>

<!-- Document, Wrapped ou RPC? -->
<style>wrapped</style>
<use>literal</use>
<!-- La classe qui implémente le service -->
<serviceClass>com.objdirect.ws.server.PersonneService</service
Class>
</service>
</services>

</xfire>

```

A travers ce fichier, nous indiquons notamment les classes qui implémentent des services web, le type de message SOAP (ici 'wrapped/literal') utilisé pour les échanges.

Utilisation des annotations

Les utilisateurs de Java 5 sont probablement familiarisés avec l'utilisation des annotations qui permettent d'enrichir le code Java à l'aide d'attributs interprétés à la compilation et pouvant modifier le comportement de la classe. L'utilisation des marqueurs de type Xdoclet, ou plus simplement des instructions Javadoc est ce qui pourrait se rapprocher le plus des annotations.

XFire prend en compte les annotations, via la plate-forme Java 5, mais il assure aussi une compatibilité pour les JDK antérieurs à l'aide des bibliothèques Common Attribute ou BackPort 175. Les spécifications pour les annotations sur les Web Services sont regroupées dans la JSR181.

Dans le cas d'une utilisation avec Java 5 notre service pourra avoir la forme suivante :

```

@WebService ( name="PersonneService" )
public class PersonneService implements PersonneServiceInterface
{
    // ...

    @WebMethod
    @SOAPBinding( style=SOAPBinding.Style.DOCUMENT,
    use=SOAPBinding.Use.LITERAL )
    public void addPersonne( @WebParam( name="Personne", header=true

```

```

) Personne p )
    {
        m_personnelList.add( p );
    }

@WebMethod
@SOAPBinding( style=SOAPBinding.Style.DOCUMENT,
use=SOAPBinding.Use.LITERAL )
    public Personne[] getAllPersonnes()
    {
        return (Personne[])m_personnelList.toArray(new
Personne[0]);
    }

@WebMethod
@SOAPBinding( style=SOAPBinding.Style.DOCUMENT,
use=SOAPBinding.Use.LITERAL )
    public Personne getPersonne(@WebParam(name="indice") int indice )
    {
        return (Personne)m_personnelList.get( indice );
    }
}

```

L'attribut @WebService marque la classe comme étant un service web. Les attributs @WebMethod indiquent les méthodes accessibles sous forme d'opérations web avec le binding SOAP défini à l'aide l'attribut @SOAPBinding (ici document/literal).

Pour indiquer l'utilisation des annotations JSR181, nous ajustons le fichier service.xml comme suit :

```

<xfire>
<services>
<service>
<name>PersonneService</name>
<namespace>urn:objdirect:PersonneService</namespace>
<serviceClass>com.objdirect.ws.server.PersonneService</serviceClass>
<annotations>org.codehaus.xfire.annotations.jsr181.Jsr181WebAnnotations
</annotations>
</service>
</services>
</xfire>

```

Développement Client

L'API cliente de XFire est encore expérimentale, c'est pourquoi nous préférons utiliser la librairie Apache Axis pour illustrer l'appel à un Web Service XFire. A travers les lignes de code ci-dessous, nous récupérons dynamiquement un code client implémentant l'interface 'PersonneServiceInterface' qui va se charger d'effectuer les appels Web Services en toute transparence.

```

// Définition des emplacements physiques du document WSDL
URL url = new URL("http://localhost:8080/mywebservice/services/
PersonneService?wsdl");
QName serviceName = new QName( "urn:objdirect:Personne
Service", "PersonneService");

```

```

// Récupération dynamique d'une instance du stub de notre service
ServiceFactory factory = ServiceFactory.newInstance();
Service service = factory.createService(url, serviceName);
PersonneServiceInterface ps = (PersonneServiceInterface)service.getPort(
PersonneServiceInterface.class);

// Enregistrement des désérialiser pour les types de données utilisées
// ...

// Appel des webmethods
Personne[] personnes = ps.getAllPersonnes();
System.out.println( Arrays.asList( personnes ) );

```

Dans notre exemple, nous utilisons des types de données 'Personne' et 'Personne[]' qui doivent être convertis en message. Nous enregistrons le serializer/deserializer pour le type de données complexes à l'aide du code suivant :

```

Class classz = Personne.class;
QName beanName = new QName( "http://server.ws.objdirect.com",
"Personne" );
TypeMapping tpMap = (TypeMapping)service.getTypeMappingRegistry().get(
DefaultTypeMapping());
tpMap.register( classz, beanName, new BeanSerializerFactory(
classz, beanName), new BeanDeserializerFactory(classz, beanName) );

```

XFire et Axis

Comparaison des fonctionnalités de XFire et d'Apache Axis :

Fonctionnalité	XFire 1	Axis 1	Axis 2
Soap 1.1	x	x	x
Soap 1.2	x	x	x
Soap avec attachments		x	x
StAX	x		x
WSDL 1.1	x	x	x
WSDL 1.2			x
WSDL -> code (Client)	expérimental	x	x
WSDL -> code (serveur)		x	x
JAX-RPC		x	
POJO -> Web service	x		
JSR 181	x	x	
Microsoft DIME		x	



Alexis Agahi
Expert XML/J2EE
Objet Direct, filiale de Homsys Group



Créé en 1991, Homsys Group est spécialisée autour de la Business Intelligence (Homsys) et les technologies Objet et Internet (Objet Direct) ; il est implanté à Paris, Marseille, Lyon, Toulouse, Bordeaux, Grenoble et Rennes. www.homsysgroup.com

C# 3.0 : attention, chaud devant !



Microsoft a rendu public la troisième mouture des spécifications de C# lors du dernier PDC. Ces notions de typage implicite, de méthode d'extension, d'expressions lambda, d'initialiseur d'objet et de collection, de type anonyme, sont indispensables à la compréhension du projet LINQ.

Attention, les spécifications de C# v3 ne sont encore ni définitives, ni complètes (ne couvrant pas par exemple les «expressions trees»). L'ensemble des exemples qui suivent ont tous été testés en installant une preview de LINQ comprenant un compilateur test C# v3. En outre, cette préversion ne fonctionne qu'avec la bêta 2 de Visual C# 2005 (version Express comprise). Pour ceux qui désirent tester, vous devez d'abord vérifier que votre version de Visual C# est bien la 8.0.50215.44 (bêta2). Puis, vous devez exécuter le fichier de commandes «Install Visual Studio IDE Support.cmd» situé sous le répertoire bin de la preview de LINQ.

1 Le typage implicite

En C# lorsque vous déclarez une variable vous devez préciser son type comme ceci :

```
int i = 12;
string s = "Bonjour chers lecteurs de Programmez!";
double d = 1.5;
```

C# v3 introduit la possibilité de déclarer une variable locale qui sera typée implicitement :

```
var i = 12;
var s = "Bonjour chers lecteurs de Programmez!";
var d = 1.5;
Console.WriteLine(i.GetType());
Console.WriteLine(s.GetType());
Console.WriteLine(d.GetType());
```

Ce qui donne :

```
System.Int32
System.String
System.Double
```

Nous ne sommes pas en présence d'un langage de script, mais bien de C# v3 ! C'est le compilateur qui automatiquement déclarera une variable du type de la valeur de l'initialiseur. Si vous n'indiquez pas d'initialiseur, le compilateur générera une erreur de syntaxe Error 1, Syntax error, '=' expected :

```
var i; // provoque une erreur
```

Soulignons qu'il s'agit bien de typage implicite et non de type variant, ainsi...

```
var i = 12;
i = "Bonjour chers lecteurs de Programmez!";
```

ne compile pas... Le type implicite est nécessaire pour LINQ mais de notre point de vue, cette manière de procéder en dehors du contexte de LINQ rend le code source moins lisible. Remarquez que la déclaration d'une variable locale implicitement typée est une inférence basée sur l'assignation de variable. Par contre, lorsqu'une méthode générique est appelée sans spécifier d'argument, l'inférence de type retrouvera les différents paramètres à inférer. Il s'agit alors d'une inférence basée sur l'utilisation des variables "à inférer".

2 Les initialiseurs d'objets

En C# v3 vous pouvez déclarer un objet par typage implicite, tout en initialisant directement ses propriétés. Exemple. Nous avons une classe ATP :

```
class ATP {
    public string Nom;
    public string Nationalité;
    public int Points;
    public int classement;
}
```

Et au lieu d'écrire :

```
ATP ClassementATP = new ATP();

ClassementATP.Nom = "FEDERER Roger";
ClassementATP.Nationalité= "SUI";
ClassementATP.Points=1210;
ClassementATP.classement = 1;
```

Nous pouvons écrire :

```
var ClassementATP = new ATP {Nom = "FEDERER Roger", Nationalité=
    "SUI", Points=1210,classement = 1};
```

Ce qui réduit le nombre de lignes à encoder tout en introduisant plus de flexibilité.

3 Les initialiseurs de collection

Il s'agit d'une extension naturelle aux initialiseurs d'objets, qui consiste à dresser une liste d'initialiseurs pour chaque élément d'une collection, séparé par une virgule et enfermé par {}.

En voici un exemple :

```
var i = new int[] {1,2,3,4,5};
var j = new int[] {6,7,8,9,10};
```

```
var ij = new List<int[]> { i,j,i };
Console.WriteLine(ij[2][0]);
```

Entre parenthèses la première ligne ne peut s'écrire :

```
var i = {1,2,3,4,5};
```

Car l'initialiseur d'une collection n'est pas autorisé (ni le null d'ailleurs). Par contre, l'initialiseur de ij est un initialiseur spécifique aux collections, ce qui permet au programmeur d'ajouter des éléments très rapidement.

N'oubliez pas d'inclure l'espace de nom System.Collections.Generic. Avec notre exemple Client/Telephone cela donne :

```
using System.Collections.Generic;
...
List<Client> listeclient = new List<Client> {
    new Client {nom = "Xavier Leclercq",
               adresse = "rue de C SHARP v3",
               telephone = new Telephone {codeinternational = 33,
                                           coderegional = 27, numero = 123456}},
    new Client {nom = "FEDERER Roger",
               adresse = "rue de FEDERER Roger",
               telephone = new Telephone {codeinternational = 33,
                                           coderegional = 27, numero = 123455}},
    new Client {nom = "NADAL Rafael",
               adresse = "rue de NADAL Rafael",
               telephone = new Telephone {codeinternational = 33,
                                           coderegional = 27, numero = 123454}};
};
```

Comme on le voit, les initialiseurs d'objets et de collections nous ferons gagner du temps. Quel dommage que cette fonctionnalité ne soit pas incluse avec C# v2 ! J'adore !

4 Les types anonymes

L'écriture d'un initialiseur d'objet peut encore être simplifiée. En effet, avec C# v3 vous n'êtes plus obligé de spécifier le type de l'objet. Le compilateur utilisera alors un mécanisme de typage anonyme :

```
var ClassementATP1 = new {Nom = "FEDERER Roger", Nationalité= "SUI",
                          Points=1210,classement =1};
var ClassementATP2 = new {Nom = ClassementATP1.Nom, classement =1};
```

Mieux (mais tout dépend du point de vue), le compilateur peut facilement déterminer le type de ClassementATP2.Nom en le déduisant du type de ClassementATP1.Nom :

```
var ClassementATP1 = new {Nom = "FEDERER Roger", Nationalité=
                          "SUI",Points=1210,classement =1};
var ClassementATP2 = new {ClassementATP1.Nom, classement =1};
```

Pour bien en comprendre le mécanisme, reprenons une syntaxe C# classique :

```
using System;
using System.Collections.Generic;

class Client
```

```
{
    private string nom;
    public string Nom { get { return nom; }
                    set { nom = value; } }
    private string adresse;
    public string Adresse { get { return adresse; }
                          set { adresse = value; } }
}
class MaClasse
{
    static void Main()
    {
        Client client = new Client();
        client.Nom = "Xavier";
        client.Adresse = "rue de C SHARP v3";

        Console.WriteLine(client.Adresse);
        Console.ReadLine();
    }
}
```

En C# v3 cela donne :

```
var client2 = new {Nom = "Xavier",Adresse = "rue de C SHARP v3"};
```

Comme cette ligne ne possède pas de type nommé, le compilateur générera une nouvelle définition de classe en lui attribuant d'office un nom (ici <Projection>f_o) qui restera inconnu du programmeur. Ce qui générera quelque chose comme ceci :

```
public sealed class <Projection>f_o
{
    // Méthodes
    public <Projection>f_o();
    public override bool Equals(object);
    public override int GetHashCode();
    public override string ToString();
    // Propriétés
    public string Adresse { get; set; }
    public string Nom { get; set; }
    // Champs
    private string _Adresse;
    private string _Nom;
}
```

Au même titre que le typage implicite, le typage anonyme n'est pas la panacée (en dehors du contexte de LINQ).

5 Les expressions lambda et les expressions de requêtes

Avec C# v2 la notion de méthodes anonymes a été introduite dans le but de créer facilement des blocs de fonctions in-line. Les expressions lambda en combinaison avec l'inférence de type permettent de déclarer et d'utiliser plus facilement ces méthodes anonymes. Elles sont basées sur les calculs (notations) lambda qu'utilisent déjà certains langages fonctionnels comme Lisp.

Considérons la ligne suivante :

```
string[] ListeMot = {"programmation", "pratique", "Hello", "C_SHARP",
"World", "LINQ"};
```

Si nous ne voulons afficher que les mots Hello World, nous pouvons choisir de programmer une alternative dans un foreach :

```
foreach (var mot in ListeMot) if (mot.Length == 5) Console.WriteLine(mot);
```

Mais nous pouvons aussi choisir avec C# v3 de filtrer la liste à l'aide d'une expression lambda, ce qui peut apparaître plus complexe, mais le compilateur va déduire le type de la variable :

```
var resultat = ListeMot.Where( s => s.Length == 5);
foreach (var mots in resultat) Console.WriteLine(mots);
```

Ainsi le typage de « s » se base sur la forme d'une expression lambda (une expression lambda est un délégué). De même, on peut aussi arriver au même résultat en réalisant une projection via une méthode Select, c'est ce que l'on appelle une expression de requête (« Query Expression » en anglais) :

```
string[] ListeMot = {"programmation", "pratique", "Hello", "C_SHARP",
"World", "LINQ"};
var resultat = from s in ListeMot where s.Length == 5 select s;
foreach (var mots in resultat) Console.WriteLine(mots);
```

Vous avez à votre disposition de nombreuses clauses : where, select, group, orderby, into, etc.

Il faut bien saisir que la fonction lambda suivante :

```
Func<int> lambda = (int x) => x + x;
```

Est l'équivalent du délégué :

```
delegate T Func<T>(T t);
Func<int> funclambda = delegate(int x)
{
    return x + x;
};
```

En C# v3 nous pouvons d'ailleurs aussi écrire :

```
Func<int> lambda = x => x + x;
```

6 Les méthodes d'extension

Avec les méthodes d'extension (est-ce la bonne traduction, ou pouvons-nous dire méthodes étendues, mystère ?), vous pouvez attacher des fonctions additionnelles à un type. Elles permettent ainsi de transformer un appel de type `MaClasse.MaMethode(monObjet)` en `monObjet.MaMethode()`. Voici un exemple très simple :

```
using System;
using MonExtension;

namespace MonExtension
{
    public static class Extensions
    {
        public static string Bidon(this string s)
        {
```

```
        return (s + (" World\n"));
    }
}

class MaClasse
{
    public static void Main()
    {
        var s1 = "Hello";
        var s2 = s1.Bidon();
        Console.WriteLine(s2);
        Console.ReadLine();
    }
}
```

Remarquez au passage le mot clé « this » devant le premier paramètre de la méthode statique Bidon. Ce qualificateur this est suivi d'un paramètre qui indique le type de la méthode d'extension. Voici un autre exemple :

```
using System;
using System.Collections.Generic;

public static class Extensions
{
    public static void Print<T>(this ICollection<T> col)
    {
        foreach(T t in col) Console.WriteLine(t);
    }
}

class MaClasse
{
    public static void Main()
    {
        List<string> Mots = new List<string>();
        Mots.AddRange (new string[] {"Hello", "World !"});
        Mots.Print();
        Console.ReadLine();
    }
}
```

`Print()` n'est pas un membre de `List` mais est quand même appelé comme s'il s'agissait d'une de ses méthodes. Contrainte : la méthode `Print` est une méthode statique définie à l'intérieur d'une classe statique. Si vous avez bien suivi vous savez que `Mots.Print()` est l'équivalent de `Extensions.Print(Mots); !`

Conclusion

C# v3 emprunte des notions aux langages fonctionnels mais aussi à C Omega (alias X# ou Xen). Ce dernier langage issu des laboratoires de Microsoft n'est pas destiné à devenir un langage à part, mais sert de plate-forme de test pour les futures moutures de C#.

■ **Xavier Leclercq** Xavier.Leclercq@programmez.com

LINQ : le futur framework .NET de requêtage des données



Lors de la dernière PDC, Microsoft a présenté son projet LINQ (Language Integrated Query). LINQ définit un ensemble d'opérateurs permettant d'effectuer directement avec son langage .net préféré, comme C# v3, des requêtes sur n'importe quel type de données, que celles-ci soient issues d'une base de données relationnelles (DLINQ), ou du XML (XLINQ). LINQ se propose de manipuler les données comme des objets.

Pour l'instant, LINQ n'est ouvert qu'au C# v3 (et à la version 9 de VB.NET). Il ne fonctionne qu'avec le futur Visual Studio, Orcas, prévu pour 2007 ou 2008. Nous avons pu réaliser quelques essais en installant la preview de LINQ, fonctionnant avec Visual C# 2005 bêta 2 (Express comprise). LINQ utilise certaines fonctionnalités avancées du langage ainsi que des assemblages supplémentaires, comme System.Data.Linq.dll, System.Query.dll et System.Xml.Linq.dll. Autrement dit, la preview de LINQ fonctionne avec Visual C# 2005, le framework de base 2.0.50215, mais aussi avec le compilateur C# 3, et les assemblages cités précédemment (qu'il est nécessaire d'ajouter en références au projet). Pour installer, vous devez exécuter le fichier de commandes «Install Visual Studio IDE Support.cmd» situé sous le répertoire bin de la preview de LINQ.

1 Présentation de LINQ au travers d'un exemple simple

L'API « Standard Query Operators » autorise C# à construire une requête autour d'un tableau ou d'une collection. C'est-à-dire de méthodes déclarées dans l'espace de nom System.Query de l'assemblage System.Query.dll. Cette API est compatible avec le CLS (Common Language Specification) de .NET 2.0. Le langage utilisant l'API doit supporter les génériques et les méthodes anonymes (C# v2), mais aussi les variables typées implicitement, les expressions lambda, les méthodes d'extension, les initialiseurs d'objets et de collections et enfin les types anonymes (C# v3).

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Query;
```

```
class app {

    class ATP {
        public string Nom;
        public string Nationalité;
        public int Points;
        public int classement;
    }
}
```

Nous avons créé une classe ATP. Utilisons maintenant le mécanisme d'inférence de type, ainsi que celui des initialiseurs d'objets pour garnir notre tableau de classement :

```
static void Main() {

    var ClassementATP = new ATP[] {
        {Nom = "FEDERER Roger", Nationalité= "SUI", Points=1210,classement
        =1},
        {Nom = "NADAL Rafael", Nationalité= "ESP", Points=853,classement =2},
        {Nom = "RODDICK Andy", Nationalité= "USA", Points=548,classement
        =3},
        {Nom = "HEWITT Lleyton", Nationalité= "AUS", Points=490,classement =4},
        {Nom = "AGASSI Andre", Nationalité= "USA", Points=455,classement =5},
        {Nom = "CORIA Guillermo", Nationalité= "ARG", Points=422, classement
        =6},
        {Nom = "DAVYDENKO Nikolai", Nationalité= "RUS", Points=378,
        classement =7},
        {Nom = "SAFIN Marat", Nationalité= "RUS", Points=346, classement =8},
        {Nom = "GAUDIO Gaston", Nationalité= "ARG", Points=344, classement
        =9},
        {Nom = "PUERTA Mariano", Nationalité= "ARG", Points=308, classement
        =10}};
```

Maintenant, nous pouvons faire appel aux notions, de typage implicite, de méthode d'extension et d'expressions lambda pour écrire :

```
var requête1 = ClassementATP.Where(c => c.Points >= 455).Select(c =>
c.Nom);
```

Effectivement, cette requête retourne le nom de tous les joueurs dont le score est supérieur à 455. Il nous est aussi loisible avec LINQ de transformer cette syntaxe en quelque chose de plus proche du langage SQL :

```
var requête2 = from c in ClassementATP
                where c.Points >= 455
                select new {c.Nom};
```

Nous pouvons également demander de classer la liste à l'aide de la clause orderby et de grouper les données de classement et de nom, grâce

aux types anonymes :

```
var requête3 = from c in ClassementATP
               where c.Points >= 455
               orderby c.classement
               select new {c.classement, c.Nom};
```

Pour afficher les noms (via le mécanisme de méthodes d'extension), il suffit de coder :

```
foreach (var ligne in requête3) Console.WriteLine(ligne.Nom);
Console.ReadLine();
}
}
```

2 Comment LINQ fonctionne-t-il ?

Lorsque le compilateur rencontre :

```
var requête2 = from c in ClassementATP
               where c.Points >= 455
               select new {c.Nom};
```

Il le transforme automatiquement en :

```
var requête2 = ClassementATP.Where(c => c.Points >= 455).Select(c
=> c.Nom);
```

Ce qui, en fait, cache un système de délégués comme ceci :

```
delegate R Func<A,R>(A arg);

class C<T>
{
public C<T> Where(Func<T,bool> predicate);
....
}
```

L'assemblage System.Query.dll n'est rien d'autre qu'une colle logicielle permettant de mettre tout ceci en place.

La syntaxe obtenue avec LINQ est assez proche de ce qui se fait avec Visual Foxpro avec un SELECT SQL. Sauf que les opérateurs comme where ou orderby sont bel et bien des méthodes d'extension qui « étendent » C<T>. Une méthode étendue est une méthode statique capable de manipuler l'instance, et qui sera ajoutée aux méthodes du type. C'est assez compliqué à comprendre en théorie, mais facile à mettre en pratique : les types délégués sont construits à la volée (type créé à la compilation). Ainsi le compilateur vérifie que la méthode n'existe pas, soit d'abord dans la classe source de l'objet, soit ensuite dans les éventuelles classes d'extension.

3 LINQ possède un grand nombre d'opérateurs

La variable locale requête2 est initialisée avec une expression d'interrogation composée de plusieurs opérateurs, pour effectuer par exemple

une restriction (un filtre avec where), une projection (selectmany), ou une agrégation (count, min ,max, average, fold). Nous aurions pu écrire orderby c.classement descending pour classer par ordre inverse. Vous pouvez aussi classer en prenant plusieurs critères en considération. Par exemple, pour classer la collection d'éléments selon la longueur des chaînes des noms, puis selon le classement, vous pouvez écrire orderby c.Nom.length, c.classement.

Le filtre de restriction appliqué est ici fort simple (where c.Points >= 455) mais il aurait pu être plus complexe. Par exemple nous pourrions appliquer une opération de partitionnement avec l'opérateur TakeWhile :

```
int[] nombres = { 3,14,15,92,65,35,89,79};

var ListeNbrPlusGrandQueVingt = nombres.TakeWhile(n => n > 20);

Console.WriteLine("Liste nbr > 20 :");
foreach (var n in ListeNbrPlusGrandQueVingt) {Console.WriteLine(n);}
```

Nous avons affaire à une drôle d'expression (n => n > 20); qui, pour rappel, est baptisée « expression lambda ». Remarquez au passage que nous avons encore affaire à un type implicite.

Pour s'amuser voici un petit hello world avec LINQ :

```
using System;
using System.Query;

class Program
{
static void Main(string[] args)

{
string[] ListeMot = {"programmation", "pratique", "Hello", "C_
SHARP", "World", "LINQ"};
var resultat = from s in ListeMot where s.Length == 5 select s;
foreach (var Mots in resultat) Console.WriteLine(Mots);
}
}
```

4 DLINQ et XLINQ

LINQ peut être utilisé pour interroger une base de données relationnelle, en l'occurrence, pour l'instant SQL SERVER, ou encore une source de données hiérarchique XML. Voici comment programmer un select avec DLINQ :

```
var q =
from c in db.Customers
where c.City == "London"
select c.ContactName;

ObjectDumper.Write(q);
```

et maintenant une insertion :

```
Console.WriteLine();
Console.WriteLine("*** INSERT ***");
var newCustomer = new Customer { CustomerID = "MCSFT",
```

```

        CompanyName = "Microsoft",
        ContactName = "John Doe",
        ...
    };
    db.Customers.Add(newCustomer);
    db.SubmitChanges();

```

Et voici comment charger et interroger une source XML :

```

XDocument doc = XDocument.Load(dataPath + "nw_customers.xml");
foreach (XElement result in doc.Elements("Root").Elements
("Customers"))
    Console.WriteLine(result);

```

Pour utiliser LINQ avec SQLSERVER, vous devez commencer par importer l'espace de nom Dlinq, puis importer l'espace de nom nwind, qui a été généré à partir du schéma de la base de données SQL Northwind. La preview inclut un outil baptisé SQLMETAL qui se charge de ce travail.

```

usage: sqlmetal [options] [<input file>]
options:
  /server:<name>   database server name
  /database:<name> database catalog on server
  /user:<name>     login user id
  /password:<name> login password
  /views          extract database views
  /xml[:file]     output as xml
  /code[:file]    output as c# source
  /namespace:<name> namespace used for source code
  /pluralize      auto-pluralize table names

```

De cette manière, pour générer le code source C#, vous taperez : sqlmetal /server:myserver /database:northwind /namespace:nwind /code:nwind.cs

```

using System;
using System.Query;
using System.Data.Dlinq;
using nwind;

class Program
{
    static void Main(string[] args)
    {
        Northwind db = new Northwind("Data Source=(local);Initial Catalog
=Northwind;Integrated Security=True");
        Table<Customers> allCustomers = db.GetTable<Customers>();
        var resultat =
            from c in allCustomers
            where c.ContactTitle.Length == 6
            select c.ContactName;

        foreach (var ligne in resultat) Console.WriteLine(ligne);
    }
}

```

La classe Customers est autogénérée. La première ligne établit une connexion à la base, et la deuxième extrait les données de la table

Customer. Enfin, la troisième ligne initie une requête avec comme restriction que chaque longueur de chaîne ContactTitle soit égale à 6. Vous pouvez aussi partir du data context qui fait penser à la chaîne de connexion ADO. Ce contexte peut aussi être généré avec SQLMETAL : D'abord en créant un fichier xml :

```
sqlmetal /server:myserver /database:northwind /xml:mymeta.xml
```

Puis, en générant un fichier mdf (le contexte) :

```
sqlmetal /xml:mymeta.xml mydbfile.mdf
```

Pour ouvrir la base de données vous pouvez maintenant écrire :

```
Northwind db = new Northwind("c:\northwind\northwnd.mdf");
```

Une requête pourra s'écrire :

```

string id = "ALFKI";
var cust = db.Customers.First(c => c.CustomerID == id);

```

Changeons le nom du contact :

```
cust.ContactName = "Nouveau Contact";
```

Et effaçons un enregistrement :

```

Order ord0 = cust.Orders[0];
db.Orders.Remove(ord0);

```

DLINQ supporte évidemment les procédures stockées :

```

if (original.UnitsInStock != current.UnitsInStock) {
    int rowCount = this.ExecuteCommand(
        "exec UpdateProductStock " +
        "@id={0}, @originalUnits={1}, @decrement={2}",
        original.ProductID,
        original.UnitsInStock,
        (original.UnitsInStock - current.UnitsInStock)
    );
    if (rowCount < 1)
        throw new OptimisticConcurrencyException();
}

```

Conclusion

Nous n'avons abordé que quelques-unes des possibilités de LINQ. Microsoft doit maintenant travailler à améliorer cette technologie, et surtout répondre aux questions légitimes que tout le monde se pose. DLINQ est-il extensible à d'autres bases de données que SQL SERVER ? Le framework pourra-t-il gérer plusieurs bases de données comportant des dizaines de tables ? Quid de l'héritage, du mapping de BLOBs, de la sécurité ? Comment une requête LINQ du style `Select(p => p.Name).Fold((s, t) => t.Length > s.Length ? t : s)` sera-t-elle implémentée sous dlink ? (absence du mot clé FOLD en langage SQL).

Liste des liens utiles :

COMEGA : <http://research.microsoft.com/Comega/>

LINQ : <http://msdn.microsoft.com/netframework/future/linq/>

■ Xavier Leclercq

Xavier.Leclercq@programmez.com



Manipuler vos images avec Java

Toujours présent lorsqu'il s'agit de multimédia, Sun Microsystems fournit une API pour travailler avec les images. Cette API, la JAI pour Java Advanced Imaging est complexe et très puissante. Nous vous proposons de découvrir son maniement de base.

Java a considérablement augmenté ses ambitions depuis son apparition. La manipulation d'images en est un exemple. Au départ, une API relativement simple, prenant peu de formats d'images en charge a d'abord été considérablement enrichie avec Java 2. Aujourd'hui, Sun propose au téléchargement (<http://java.sun.com/products/java-media/jai/>) une API séparée, qui se veut exhaustive, et même extensible. Puissante, cette API est très intéressante et très complète, à tel point qu'il faudrait un livre pour en faire le tour. Ce livre existe. Il s'agit du 'JAI Guide' téléchargeable sur le site de Sun. Ce document, rédigé en anglais, compte près de 500 pages. Notre article a pour objectif de vous familiariser avec la philosophie de l'API et l'organisation de ses classes. Pour cela, nous allons réaliser quelques opérations de base à partir desquelles vous pourrez écrire vos propres applications.

1 Charger et sauvegarder une image

L'opération peut s'effectuer soit directement depuis un fichier, soit par l'intermédiaire d'un flux (Stream). Les Streams de l'API sont dit "seekables", ce qui signifie que les données de l'image y sont mises en cache et qu'il est possible de parcourir ces données dans un sens comme dans l'autre. Quelle que soit la méthode, de nombreux formats sont supportés : BMP, FPX, GIF, JPEG, PNG, PNM, TIFF. Au centre de l'API se trouve le package `javax.media.jai`, et au centre ce package de se trouve la classe `JAI`. Cette classe a pour principale méthode la méthode `create`, multiples fois surchargée. Au plus simple, cette méthode retourne un objet de type `PlanarImage` ou dérivé, moyennant un nom d'opération à effectuer et un objet paramètre en argument. Comme on s'en doute, un objet `PlanarImage` gère une collection de pixels, donc une image. L'invocation ressemble à ceci :

```
PlanarImage image = JAI.create("nom_operation", parametres);
```

La première question à résoudre est de trouver le nom de l'opération. Pour cela, il faut visiter la documentation du package `javax.media.jai.operator`. A quelques exceptions près, toutes les classes ont leurs noms suffixés par 'Descriptor'. Ces classes encapsulent l'opération que l'on souhaite effectuer. Quand la méthode `JAI.create` doit effectuer une opération, elle délègue en fait cette opération à la méthode `create` de la classe concernée. Le tout est donc de fournir un nom d'opération correct et ce nom est donné dans la documentation. Ainsi, voulons-nous charger une image ? L'opération est prise en charge par la classe `FileLoadDescriptor` et nous voyons dans la documentation que le 'Global Name' de cette classe est "fileload". Nous passerons donc la chaîne "fileload" en premier argument de `JAI.create`.

Nous comprenons que `JAI.create` est une méthode très générique. Il faut donc un protocole pour passer les paramètres adaptés à chaque opéra-

tion. Ces paramètres sont gérés par un objet `ParameterBlock` (une classe de `java.awt.image.renderable`). L'objet contient deux vecteurs. Un pour les sources, l'autre pour les paramètres proprement dit. On utilise les sources lorsqu'il s'agit de faire une opération à partir d'une ou plusieurs images. Nous n'en sommes pas encore là, car nous devons charger un fichier pour obtenir une image. Donc nous ne spécifions pas de sources. Nous devons seulement passer des paramètres via la méthode `add` du `ParameterBlock`. Les paramètres requis sont énumérés dans le tableau 'Parameter List' de la documentation de `FileLoadDescriptor`. Si l'on omet de donner un paramètre, une valeur par défaut (également documentée) est utilisée. Il est bien évident que l'ordre d'ajout des paramètres est déterminant. Pour charger une image, nous n'avons besoin que du nom de fichier. Le code est donc finalement très simple :

```
ParameterBlock pb = new ParameterBlock();
pb.add("moulin.jpg");
PlanarImage image = JAI.create("fileload", pb);
```

Etudions maintenant le cas de la sauvegarde. Le principe est le même et le restera pour toutes les autres opérations. C'est un objet `FileStoreDescriptor` qui sera impliqué cette fois. La documentation nous donne le nom global de l'opération, ainsi que les paramètres qui sont au minimum un nom de fichier cible, éventuellement suivi par le type de format d'image, un codec et un booléen demandant de vérifier l'opération. Il y a en outre une autre différence. Nous avons cette fois une source. Il s'agit de notre image en mémoire que nous voulons sauvegarder. Nous aboutissons donc à ce code, pour sauvegarder notre image au format PNG :

```
private void saveImage() {
    ParameterBlock pb = new ParameterBlock();
    pb.addSource(image);
    pb.add("moulin.png");
    pb.add("PNG");
    pb.add(null);
    pb.add(true);
    JAI.create("filestore", pb, null);
}
```

Nous donnons ci-dessous le code de notre premier exemple, `LoadSaveImage.java`.

```
package programmez.fred.imaging;

import javax.swing.*;
import javax.media.jai.*;
import java.awt.*;
import java.awt.image.*;
```

```

import java.awt.image.renderable.*;

public class LoadSavelmage extends JPanel{

    private PlanarImage image;
    private BufferedImage buflmage;

    public LoadSavelmage() {
        loadImage();
        createBufferedImage();
        createFrame();
        savelmage();
    }

    private void createFrame() {
        JFrame frame = new JFrame("LoadSavelmage");
        frame.setSize(buflmage.getWidth(),
            buflmage.getHeight()+33);
        frame.add(this);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }

    private void createBufferedImage() {
        buflmage = image.getAsBufferedImage();
    }

    private void loadImage() {
        ParameterBlock pb = new ParameterBlock();
        pb.add("moulin.jpg");
        image = JAI.create("fileload", pb);
    }

    private void savelmage() {
        ParameterBlock pb = new ParameterBlock();
        pb.addSource(image);
        pb.add("moulin.png");
        pb.add("PNG");
        pb.add(null);
        pb.add(true);
        JAI.create("filestore", pb, null);
    }

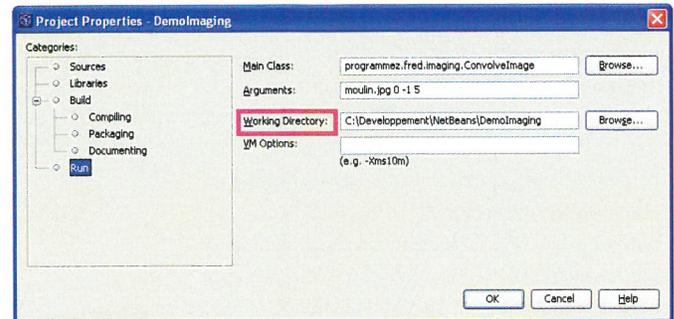
    public void paintComponent(Graphics g) {
        g.drawImage(buflmage, 0, 0, this);
    }

    public static void main(String[] args) {
        new LoadSavelmage();
    }
}

```

Ce code illustre en outre comment afficher une image dans un composant Swing après

Une image chargée et visualisée par notre premier exemple.



N'omettez pas de définir le répertoire de travail pour que l'image puisse être trouvée.

l'avoir transformée en BufferedImage. Le code a été écrit avec NetBeans 4.1 et pour simplifier, on considère que l'image à charger est placée à la racine du projet NetBeans. Pour que cela fonctionne, vous devez en outre définir ce répertoire comme répertoire de travail dans les propriétés du projet. Comme d'habitude, vous trouverez tous les sources sur le Cd-Rom accompagnant le magazine.

2 Transformer une image

Rien de plus facile, car le principe vu plus haut s'applique. Bien sûr, des paramètres supplémentaires interviennent, notamment un objet gérant l'interpolation. Voici un extrait de l'exemple ZoomImage.java, qui réduit l'image de moitié (opération Scale).

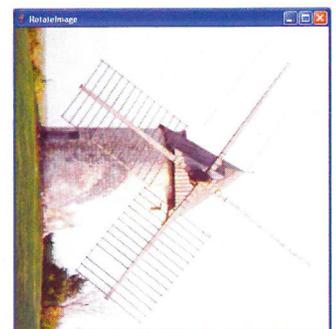
```

private void scaleImage() {
    // Choix de la méthode d'interpolation
    // et obtention d'une instance
    // de la classe correspondant
    Interpolation interpolation =
        Interpolation.getInstance(
            Interpolation.INTERP_BILINEAR);

    ParameterBlock params = new ParameterBlock();
    params.addSource(image);
    params.add(0.5F); // coefficient d'échelle en x
    params.add(0.5F); // coefficient d'échelle en y
    params.add(0.0F); // translation en x
    params.add(0.0F); // translation en y
    params.add(interpolation);
    image = JAI.create("Scale", params);
}

```

De la même manière, voici un code qui effectue une rotation de 90° à l'image (exemple Rotatelmage.java) :



Notre image après rotation de 90°.

```
private void rotateImage() {
    Interpolation interpolation =
        Interpolation.getInstance(
            Interpolation.INTERP_BILINEAR);

    ParameterBlock params = new ParameterBlock();
    params.addSource(image);
    params.add(0.0F); // localisation de l'origine en x
    params.add(0.0F); // localisation de l'origine en y
    params.add((float)(java.lang.Math.PI/2.0)); // Angle de rotation
    params.add(interpolation);
    image = JAI.create("Rotate", params);
}
```

3 Faire une convolution

Une convolution est une opération qui remplace chaque pixel d'une image par une combinaison du pixel et de ses voisins immédiats. La combinaison est décrite par une matrice carrée dite kernel. Soit une matrice :

```
a b c
d e f
g h i
```

La convolution remplace le pixel du centre (e) par la somme de

- a fois la valeur du pixel en haut à gauche
- b fois la valeur du pixel au-dessus
- c fois la valeur du pixel en haut à droite

et ainsi de suite. La matrice n'est pas forcément de 3x3 mais peut être plus grande. Pour préserver la luminosité de l'image, on s'arrange pour que la somme de tous les coefficients vaille 1. On divise donc chacun des coefficients par cette somme lors de la construction du kernel. Cette opération est dite normalisation (voir ci-dessous).

Extrait de ConvolveImage.java

```
class KernelBuilder {

    private static final float[] IDENTITY =
        {0, 0, 0,
         0, 1, 0,
         0, 0, 0};

    private static final float[] EDGE =
        {0, 1, 0,
         1, 0, 1,
         0, 1, 0};

    private static final float[] CORNER =
        {1, 0, 1,
         0, 0, 0,
         1, 0, 1};

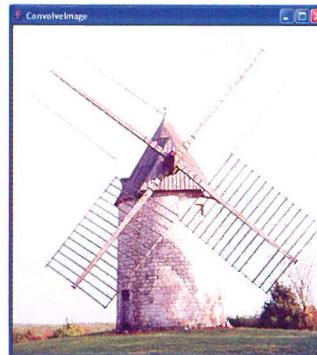
    public static KernelJAI buildKernel(int corner,
```

```
int edge,
int identity) {
    float[] kernel = new float[9];
    int sum = corner * 4 + edge * 4 + identity;
    if (sum == 0)
        sum = 1; // Pour éviter une division par zéro
    for (int i = 0; i < 9; i++) {
        kernel[i] = (corner * CORNER[i]
            + edge * EDGE[i]
            + identity * IDENTITY[i]) / sum;
    }
    return new KernelJAI(3, 3, kernel);
}
```

Si cette normalisation n'est pas possible (lorsque la somme des coefficients vaut zéro avant la normalisation) les résultats sont assez psychédéliques. Le dernier programme d'exemple (ConvolveImage.java) démontre cela avec un kernel 3x3. Il attend 4 paramètres sur la ligne de commande. Le nom du fichier d'images à charger, un coefficient commun aux angles, un coefficient commun aux autres pixels de bord et un coefficient pour le centre. Essayez le programme avec différentes valeurs :

Coefficients	Effet
1 0 0	Effet de flou
0 -1 5	Accentuation des détails
-4 -4 32	Gravure à l'eau forte

L'accentuation des détails et la "gravure à l'eau forte" sont illustrés ci-dessous.



Une convolution peut faire ressortir les détails d'une image.



Ou, au contraire, une convolution peut dénaturer une image.

Dans le dernier cas, la somme des coefficients vaut zéro. Le kernel n'est pas normalisé et la luminosité de l'image est complètement dénaturée. Enfin, rien n'oblige à donner les coefficients de façon symétrique dans le kernel. A vous de jouer !

■ Frédéric Mazué
fmazue@programmez.com



Mettre en œuvre l'interopérabilité COM/.Net

COM est une technologie avec des atouts indéniables. Il en est de même pour la technologie .Net. La coopération entre les deux est prévue par .Net sous le terme d'interopérabilité. Cet article vous propose quelques manipulations pratiques, afin de faire connaissance avec les possibilités qui nous sont offertes.

Microsoft met aujourd'hui sa technologie .Net en avant, à juste titre. Il n'y a pas si longtemps, une technologie phare de Microsoft était COM. Windows est d'ailleurs basé dessus en grande partie et des montagnes de code ont été écrites en COM. Il est donc légitime de pouvoir travailler avec COM depuis .Net, que ce soit pour de la programmation système, de la réutilisation de code existant, une migration en douceur vers .Net, de la programmation Office, ou même pour fournir une interface à un langage de script. Les cas de figure ne manquent pas. Pour examiner par la pratique quelques possibilités, nous allons utiliser Visual Basic 2005 bêta (fourni sur le Cd-Rom de *Programmez!* 75). Nous travaillons donc avec le Framework 2.0. Si vous n'en disposez pas, l'article reste valable, même si les nouveaux EDIs sont plus conviviaux. Quelle que soit votre configuration, veillez bien à faire pointer votre PATH sur le répertoire bin de votre framework (sous c:\Windows\Microsoft.Net...) et sur le répertoire bin de la SDK sous l'EDI. En effet, nous allons utiliser quelques outils en lignes de commandes. A ce moment, un PATH pointant sur une mauvaise version du framework .Net pourrait vous apporter bien des désagréments. Comme de coutume, vous trouverez les sources de cet article sur le Cd-Rom accompagnant le magazine.

1 COM, .Net et leurs différences

Sans entrer dans les fastidieux détails, regardons brièvement ces technologies et ce en quoi elles diffèrent. COM est conçu pour les langages à pointeurs. Un composant logiciel, un "serveur" COM expose à ses clients des interfaces qui sont un tableau de pointeurs de fonctions. Une plomberie (des bibliothèques système Windows) fait que les dits pointeurs de fonctions sont vus par le client comme faisant partie de son espace d'adressage, ce qui rend l'appel des fonctions possible. Les composants COM sont répertoriés dans la base de registre et c'est sur ces informations que la plomberie s'appuie pour initialiser la communication. Un serveur COM est en général un agrégat, un regroupement de diverses interfaces dont les fonctionnalités œuvrent dans une même direction. Puisque COM n'est disponible que pour les langages à pointeurs, Microsoft a étendu cette technologie avec Automation. Un composant Automation est un composant présentant systématiquement l'interface COM conventionnelle Dispatch, qui permet aux runtimes de langages interprétés, d'obtenir des informations sur la totalité du composant et d'accéder à ses méthodes ou propriétés. Notons que du point de vue de la programmation objet, les hiérarchies d'objet COM obéissent à une relation "A-un".

Enfin, un client COM doit s'occuper de l'acquisition et de la libération des interfaces. Cette tâche revient donc, soit au programmeur, soit dans le meilleur des cas (Automation), au runtime du langage interprété.

Sous .Net tout est différent. Les hiérarchies de classes obéissent à une relation "Est-un". Le code est géré, ce qui signifie que l'acquisition et la libération des ressources échappent au programmeur. Enfin, pas de pointeurs de fonctions avec les langages de la plate-forme .Net. La collaboration entre les deux technologies s'annonce difficile. Microsoft a résolu le problème avec la génération d'objets intermédiaires, des "wrappers", qui font croire à .Net qu'il travaille avec du code géré lorsqu'il communique avec un serveur COM, ou à COM qu'il travaille avec du code non géré lorsqu'il travaille avec un serveur écrit en .Net. Ce sont ces wrappers qui permettent l'interopérabilité, de façon quasi transparente finalement.

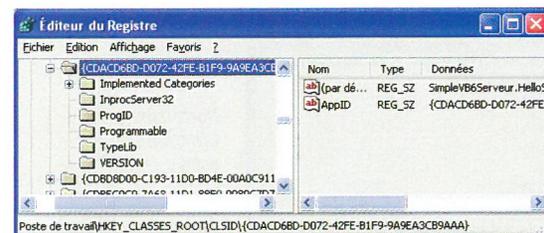
2 Un client .Net à un serveur COM

Pour que nous parlions de la même chose, voici un composant COM/Automation écrit en Visual Basic 6.0 :

Option Explicit

```
Public Function Hello(nom As String) As String
    Hello = "Bonjour, " + nom + ", depuis Visual Basic 6"
End Function
```

Le serveur a pour nom SimpleVB6Serveur et la fonction ci-dessus est contenue dans la classe HelloServeur. Vous trouverez la dll de ce composant sur le Cd-Rom sous le nom de SimpleVB6Serveur.dll. Vous devez déposer cette librairie quelque part et l'enregistrer depuis l'endroit où vous l'avez déposée.



Tout composant COM/Automation doit être inscrit dans la base de registre.

```
regsvr32 SimpleVB6Serveur.dll
```

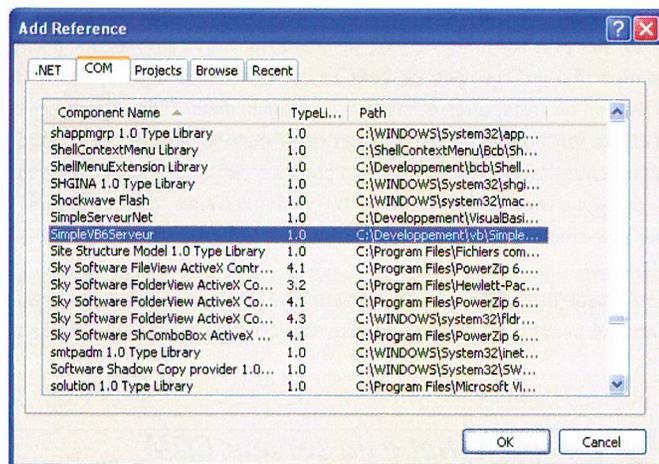
A l'issue de cette opération, des informations sont écrites dans la base de registre et le composant COM est opérationnel. La désinscription est similaire:

```
regsvr32 /u SimpleVB6Serveur.dll
```

Nous pouvons maintenant écrire notre client en Visual basic .Net :

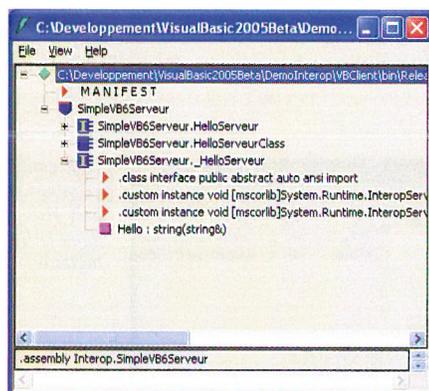
```
Imports SimpleVB6Serveur

Module Module1
    Sub Main()
        Dim com_obj As New SimpleVB6Serveur.HelloServeur()
        Console.WriteLine(com_obj.Hello("Fred"))
    End Sub
End Module
```



Tout composant COM/Automation peut être référencé comme un vulgaire assembly .Net.

Par ce code, nous voyons que notre serveur est "instancié" comme on le ferait avec une classe .Net ordinaire. A ce stade toutefois, la directive Imports ne fonctionne pas. Nous devons ajouter une référence sur le composant COM à notre projet Visual Basic. En fait, comme vous pouvez le constater en explorant les répertoires bin/Debug ou bin/Release de votre projet, cette opération a pour effet de voir un assembly automatiquement généré. Cet assembly porte le nom de



Notre composant COM/Automation est vu comme une classe .net grâce à un wrapper généré automatiquement.

Interop.SimpleVB6Serveur.dll et c'est une référence sur cet assembly qui est en fait finalement ajoutée à votre projet. Si vous ouvrez cet assembly avec ildasm.exe, vous pourrez voir que notre composant COM est devenu une vraie classe .Net, au moins du point de vue de ce dernier. Les fonctionnalités d'achèvement de code de l'EDI sont mêmes opérationnelles. Si vous ne disposez d'aucun IDE vous pouvez quand même travailler. Ici Visual Basic n'a fait que lancer un utilitaire comme ceci :

```
tlbimp SimpleVB6Serveur.dll /out:Interop.SimpleVB6Serveur.dll
```

En outre, le préfixe Interop n'est qu'une simple convention que vous n'êtes pas obligé de suivre.

3 Un client .Net à liaison retardée

L'utilitaire tlbimp génère le wrapper à partir de la librairie de types, autrement dit, des informations sur les interfaces COM qui accompagnent le composant. Il arrive parfois que l'on rencontre des composants sans librairies de type. Nous devons alors procéder différemment. Nous ne procédons plus à une liaison dite immédiate comme précédemment, mais à une liaison dite retardée ou différée. Bien sûr nous devons connaître par un autre moyen (une documentation) les fonctionnalités du composant qui n'est pas autodescriptif (sans librairie de types). On utilise alors le mécanisme de réflexion de .Net :

```
Imports System.Reflection

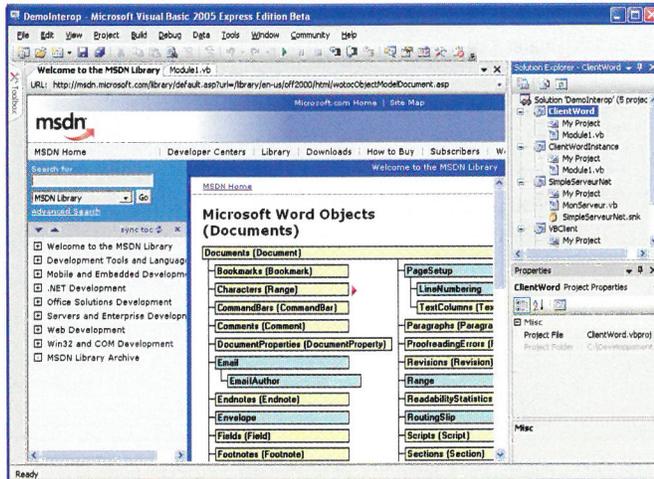
Module Module1
    Sub Main()
        Dim serveurObject As Type = _
            Type.GetTypeFromProgID("SimpleVB6Serveur.HelloServeur")
        Dim serveurDispatch As Object = _
            Activator.CreateInstance(serveurObject)
        Dim args() As Object = {"Fred"}
        Dim resultat As Object
        resultat = serveurObject.InvokeMember("Hello", _
            BindingFlags.InvokeMethod, _
            Nothing, serveurDispatch, _
            args)

        Console.WriteLine(resultat)
    End Sub
End Module
```

Globalement, ce code est lourd, lent et laid. On utilisera donc cette approche que contraints et forcés.

4 Programmer Word depuis .Net

Word, ainsi que les autres composants de la suite Office, sont des serveurs COM/Automation en plus d'être des applications. Toutes les fonctionnalités de ces applications sont automatiquement exposées à l'installation de la suite sous forme d'interfaces COM. Ainsi, il est possible par exemple de piloter totalement Word depuis l'extérieur, ce qui constitue un bon exercice de manipulation d'agrégat d'interfaces. Le code d'exemple que nous fournissons a été testé avec Word 2000 mais est



La documentation de Word 2000 vue depuis Visual Basic 2005.

directement transposable Word XP. Mieux que cela, si vous ne disposez pas de la suite Office, vous pouvez appliquer le principe à Open Office qui lui aussi expose des interfaces COM/Automation. Dans un cas comme dans l'autre, la difficulté consiste à explorer la documentation des applications afin d'avoir une idée claire de l'organisation des interfaces et du rôle de chacune. En ce qui concerne Word, celui-ci est documenté sur le site de Microsoft à <http://msdn.microsoft.com>. Il vous suffit d'y faire, par exemple, une recherche sur les mots "Microsoft Word Objects Microsoft Word 2000 Documents" pour obtenir un lien sur une page donnant un diagramme de l'interface "Documents". Vous pouvez même naviguer dans cette documentation en ligne, directement depuis Visual Basic 2005. Voici un exemple de code qui lance une instance de Word, y crée un document, écrit du texte dedans, modifie le texte, puis enregistre le document.

```
Imports Word
Imports System.Threading

Module Module1

    Sub Main()
        Dim texte As String = "Programmez!"
        Dim nouveau_texte As String = "Programmez! Word avec VB .Net"
        Dim wordapp As New Word.Application()
        wordapp.Visible = True
        Dim mondoc As Word.Document
        mondoc = wordapp.Documents.Add()
        Dim monrange As Range
        monrange = mondoc.Range(0, 0)
        monrange.Text = texte
        Thread.Sleep(1000)
        Dim selection As Selection
        selection = wordapp.Selection
        Dim find As Find
        find = wordapp.Selection.Find
        find.Text = texte
        find.Execute()
    End Sub
End Module
```

```
Thread.Sleep(1000)
selection.TypeText(nouveau_texte)
Thread.Sleep(1000)
mondoc.SaveAs("prog_net.doc")
wordapp.Quit()
End Sub
End Module
```

Bien entendu, on prendra soin d'importer Word exactement de la même manière que nous avons importé notre petit serveur écrit en VB6. Ce code parle de lui-même. On voit que l'on accède aussi facilement à une interface COM que si celle-ci était un membre public d'un objet .Net. Notre exemple démarre une instance de Word, mais on peut vouloir souhaiter travailler avec une instance déjà existante plutôt que d'en démarquer une nouvelle. L'espace de nom System.Runtime.InteropServices contient une multitude de classes pour une programmation plus avancée avec le code non managé. Parmi ces classes, la classe Marshal offre un jeu de méthodes qui sont la réplique d'API COM pour la programmation au plus bas niveau. Parmi ces méthodes nous trouvons GetActiveObject qui est la solution à notre problème.

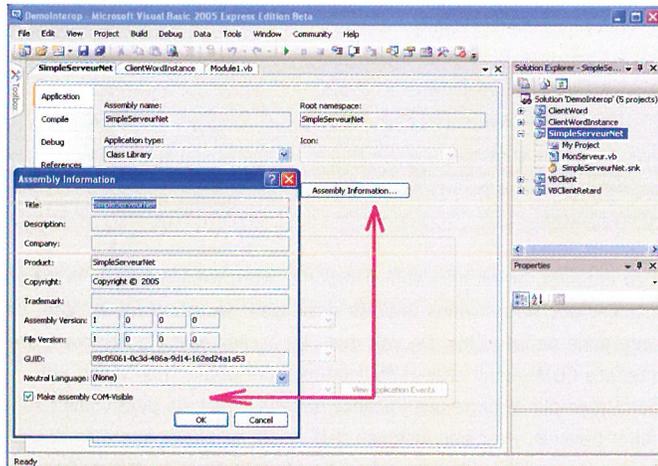
```
Module Module1
    Sub Main()
        Try
            Dim wordapp As Word.Application = _
                Marshal.GetActiveObject("Word.Application")
            Console.WriteLine("Une instance de Word a été trouvée")
        Catch ex As Exception
            Console.WriteLine("Erreur: MS-Word ne tourne pas actuellement")
        End Try
    End Sub
End Module
```

Travailler à bas niveau est rarement utile grâce aux wrappers. Toutefois il est intéressant de jeter un coup d'œil aux classes de System.Runtime.InteropServices.

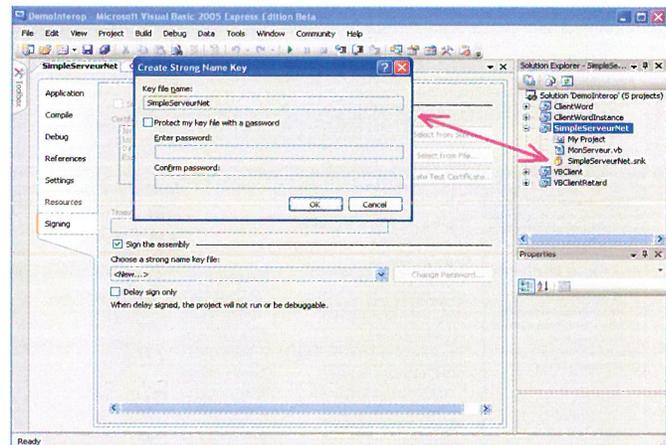
5 Écrire un serveur COM en .Net

Nous abordons maintenant la démarche inverse qui consiste à écrire avec .Net un serveur vu par le système comme un composant COM. Est-ce une démarche valable ? Oui, dans de nombreux cas de figure. Par exemple, on peut souhaiter écrire une application qui offre une interface de programmation avec des langages de scripts. Automation est la solution toute trouvée. Un autre cas est la migration vers .Net. COM est parfois capricieux. Réécrire les serveurs avec .Net apporte un gain non négligeable en fiabilité. Mais cela n'implique pas que l'on veuille réécrire des clients a priori codés en divers langages et éparpillés ici et là. Que notre serveur .Net se comporte comme un composant COM est une solution bien pratique.

Les règles à suivre sont simples et documentées dans le framework .Net. Nous résumons ici l'essentiel et d'un point de vue pratique uniquement. On crée un projet de type 'Class Library' sous Visual Basic 2005. Depuis les propriétés du projet on coche l'option 'COM-Visible' afin que classe puisse exposer des types à COM. Ceci n'est pas suffisant en soi. Il



Avant toute chose, nous devons rendre notre composant .Net visible à COM.



Vous devez signer votre assembly pour pouvoir l'installer dans le GAC et l'exposer à COM.

convient en outre que tous les types, méthodes et propriétés soient déclarés 'Public'. Enfin, on donnera une 'allure COM' à notre application en déclarant des interfaces et en implémentant celles-ci dans une ou plusieurs classes. Autrement dit, nous constituons une classe autour d'un agrégat d'interfaces. Voici l'équivalent .Net du serveur VB6 vu plus haut.

```
Imports System.Runtime.InteropServices

Public Interface IHelloServeur
    Function Hello(ByVal nom As String) As String
End Interface

<ClassInterface(ClassInterfaceType.AutoDual)> _
Public Class MonServeur
    Implements IHelloServeur
    Public Sub New()
    End Sub
    Public Function Hello(ByVal nom As String) As String _
        Implements IHelloServeur.Hello
        Return "Hello, " & nom & ", depuis Visual Basic .Net"
    End Function
End Class
```

On remarquera l'attribut <ClassInterface (ClassInterfaceType.AutoDual)>. Il a pour rôle d'assurer que notre composant expose à la fois des interfaces COM et une interface Automation, cette dernière étant impérativement requise pour communiquer avec les langages de script.

6 Déployer un serveur COM/.Net

Si nous compilons immédiatement, nous obtenons un assembly privé. Ce qu'il nous faut est un assembly partagé, c'est-à-dire un assembly installé dans le GAC ou Global Assembly Cache. Et pour que cette installation soit possible, notre assembly doit être doté d'un nom fort, ce qui signifie que son identité est renforcée par une clé publique et une signature digitale. Tout cela s'obtient depuis l'EDI de Visual Basic. Dans les propriétés du projet choisissez "Signing", puis "Sign the assembly". Donnez

ensuite un nom de fichier. Inutile de choisir un fichier crypté. A l'issue de l'opération, un élément nom_du_fichier.snk apparaît dans le volet d'exploration du projet. Compilez à présent votre projet pour que la signature soit incorporée à l'assembly. Pour installer ce dernier dans le GAC, ouvrez une console et positionnez-vous dans le répertoire qui contient l'assembly. A priori ce répertoire est sous \bin\Release dans l'arborescence de votre projet. Saisissez la commande :

```
gacutil \i SimpleServeurNet.dll
```

Si vous souhaitez ultérieurement désinstaller l'assembly, la commande sera :

```
gacutil /u SimpleServeurNet (sans .dll)
```

Enfin, il faut générer et enregistrer un wrapper COM accompagné d'une librairie de types :

```
regasm /tlb SimpleServeurNet.dll
```

Pour désinstaller le wrapper la commande est

```
regasm /u SimpleServeurNet.dll (avec .dll)
```

Et voici un client en Python qui accède à notre tout nouveau serveur.

```
from win32com.client import *
from win32com.server.exception import *
from pythoncom import *

serveur = Dispatch("SimpleServeurNet.MonServeur")
try:
    print serveur.Hello('Fred')
except com_error, (hr, msg, exc, arg):
    print msg
    print exc[1]
    print exc[2]
```

Mais vous pouvez essayer avec n'importe quel autre langage de script de votre choix, c'est tout l'intérêt de COM.

■ Frédéric Mazué
fmazue@programmez.com

Développement d'applications avec Visual Studio 2005 Tools for Office

2^e partie

Nous avons vu dans la première partie deux exemples simples pour apprendre à utiliser les VSTO 2005. Le premier nous a permis de découvrir la manière de bâtir des applications Office 2003 à partir de Visual Studio 2005 et des VSTO, puis de voir ensuite l'accès aux données. Nous allons continuer notre dernier exemple et l'améliorer en ajoutant des fonctionnalités, grâce à la programmation du volet Office.

Démonstration 3 : utilisation du contrôle Action Panel

Dans les applications Office, la visualisation du volet Office s'effectue en sélectionnant l'option correspondante depuis le menu « Affichage » ou en cliquant sur CTRL+F1. Il est utilisé pour effectuer par exemple des recherches dans l'aide en ligne, et contient également des actions sur les documents.

L'objectif, en utilisant l'action panel, est de rendre la cinématique et la navigation dans votre application Office plus aisée. Il sera en effet plus simple d'avoir les boutons d'actions ou les zones modifiables dans le volet Office à droite, plutôt que de devoir atteindre les zones dans le document, et devoir par exemple utiliser l'ascenseur pour y accéder dans le cas d'un document déjà complexe et surchargé.

La première opération à effectuer est d'ajouter l'action panel à notre solution. Pour cela, on sélectionne le projet dans l'explorateur de solution, et avec le bouton droit, on ajoute un nouvel élément de type « Actions Pane Control » (figure 17).

L'action panel s'affiche en mode design, ce qui va nous permettre de définir l'interface utilisateur. Nous allons pour le moment nous contenter de définir deux boutons, qui vont nous servir à naviguer dans notre application pour passer d'un collaborateur à l'autre.

Comme on va le voir encore une fois, peu de chose à faire pour arriver au résultat demandé.

On glisse nos deux boutons sur l'action panel, précédent et suivant (figure 18).

Une fois nos deux boutons définis, il suffit d'ajouter le code correspondant. On double-clique pour cela sur le premier bouton et on ajoute le code suivant :

```
private void button1_Click(object sender, EventArgs e)
{
    Globals.Feuil2.employeesBindingSource.MovePrevious();
}
```

C'est en effet à partir de la classe Globals, créée automatiquement par les VSTO, que l'on va pouvoir accéder à partir de n'importe quel endroit dans le code de l'application au document Word ou Excel.

La classe Globals nous permet donc d'accéder au contrôle lié aux données employeesBindingSource défini dans la feuille Excel et d'appeler ses méthodes, ici MovePrevious pour passer à l'enregistrement précédent. On définit alors le code du deuxième bouton de la même manière, en appelant cette fois-ci la méthode MoveNext du contrôle employeesBindingSource :

```
private void button2_Click(object sender, EventArgs e)
{
    Globals.Feuil2.employeesBindingSource.MoveNext();
}
```

Et enfin, dernière opération pour que l'action panel s'exécute, il faut le charger au lancement de la feuille Excel. Pour cela, nous retournons dans le code de Feuill2.cs et nous allons ajouter le chargement de l'action panel. Une seule ligne de code est nécessaire pour réaliser l'opération, à ajouter à la fin de l'événement Feuill2_Startup :

```
//Chargement de l'ActionPane
Globals.ThisWorkbook.ActionsPane.Controls.Add(new ActionsPane
Control1());
```

Encore une fois, la classe Globals est utilisée pour atteindre le document et ajouter l'action panel ActionsPaneControl1 à la collection de contrôles de l'ActionPanel.

Il ne reste plus alors qu'à compiler et exécuter notre projet, qui doit maintenant afficher le volet Office à droite, avec nos deux boutons suivant et précédent. Ces deux boutons, si tout fonctionne correctement, nous permettent de naviguer d'un collaborateur à l'autre.

A noter, évidemment, que tous les contrôles du document sont mis à jour dynamiquement (figure 19).

Afin de compléter notre application, nous allons voir qu'il est possible de synchroniser facilement les données entre l'Action panel et notre feuille Excel.

Retournons sur l'Action panel en mode design, et nous allons améliorer l'interface pour y afficher le détail des informations concernant le collaborateur (figure 20).

Outre les zones de texte classiques, nous allons définir deux contrôles

plus spécifiques, à savoir un contrôle DateTimePicker, pour afficher la date de naissance, ainsi qu'un contrôle PictureBox pour afficher la photo du collaborateur.
Une fois notre interface terminée, la partie code consiste à effectuer la correspondance entre les différents contrôles et la source de données. Cette correspondance doit évidemment se faire lors du chargement de

l'action panel. Il suffit donc de double cliquer sur l'action panel pour y coder l'évènement ActionsPaneControl1_Load

```
private void ActionsPaneControl1_Load(object sender, EventArgs e)
{
    //Bind des controles avec la source de données définie sur la
    Feuil1

    txtNom.DataBindings.Add("Text",
    Globals.Feuil2.employeesBindingSource, "LastName", true,
    txtNom.DataBindings.DefaultDataSourceUpdateMode);

    txtPrenom.DataBindings.Add("Text",
    Globals.Feuil2.employeesBindingSource, "FirstName", true,
    txtPrenom.DataBindings.DefaultDataSourceUpdateMode);

    txtVille.DataBindings.Add("Text",
    Globals.Feuil2.employeesBindingSource, "City", true,
    txtVille.DataBindings.DefaultDataSourceUpdateMode);

    txtPays.DataBindings.Add("Text",
    Globals.Feuil2.employeesBindingSource, "Country", true,
    txtPays.DataBindings.DefaultDataSourceUpdateMode);

    txtTel.DataBindings.Add("Text",
    Globals.Feuil2.employeesBindingSource, "HomePhone", true,
    txtTel.DataBindings.DefaultDataSourceUpdateMode);

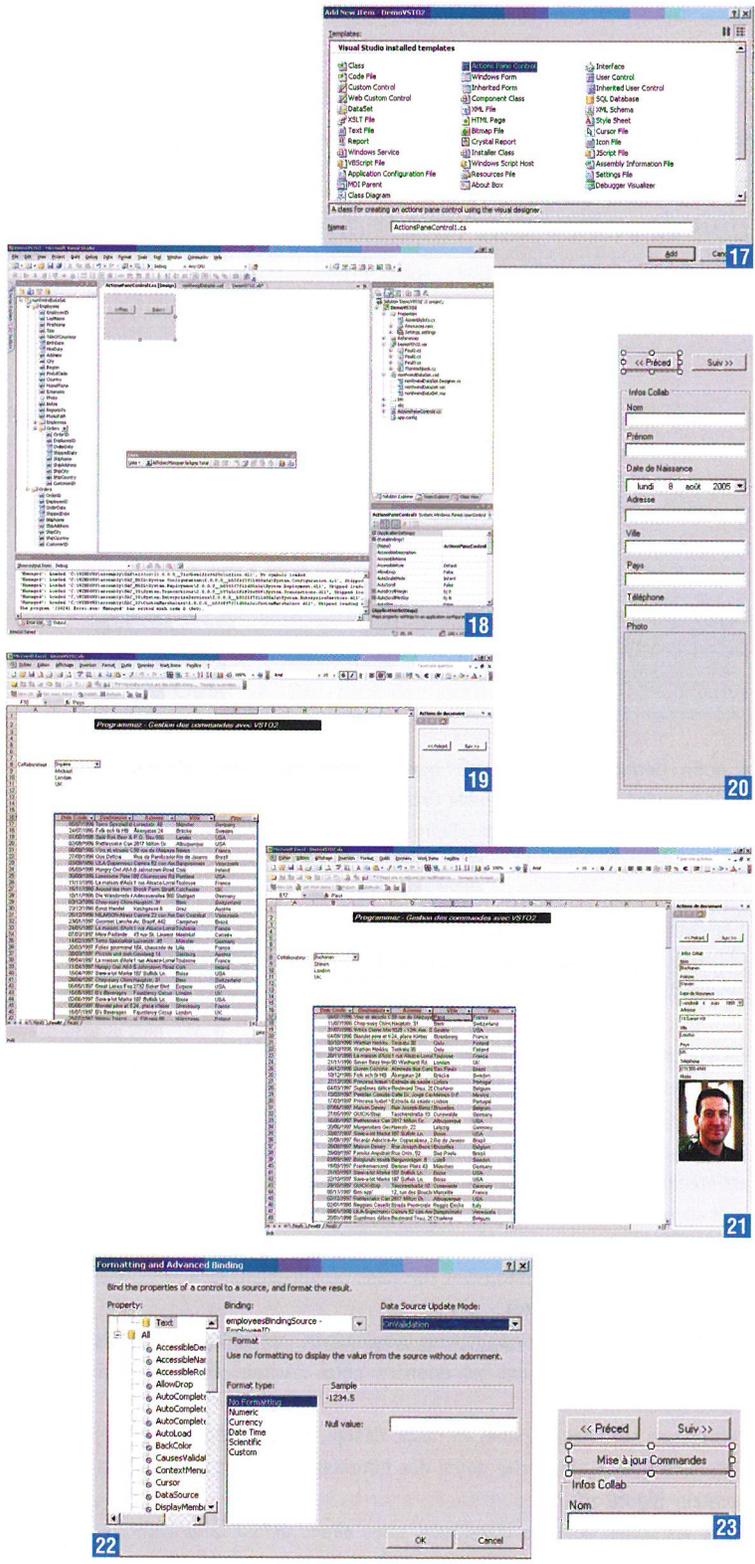
    txtAdresse.DataBindings.Add("Text",
    Globals.Feuil2.employeesBindingSource, "Address", true,
    txtAdresse.DataBindings.DefaultDataSourceUpdateMode);

    dtdateNaiss.DataBindings.Add("Value",
    Globals.Feuil2.employeesBindingSource, "BirthDate", true,
    dtdateNaiss.DataBindings.DefaultDataSourceUpdateMode);

    // bind de l'image avec la colonne Photo de Employees
    pictureBox1.DataBindings.Add("Image",
    Globals.Feuil2.employeesBindingSource, "Photo", true);
}
```

Pour chaque champ, la même opération est à effectuer. Une correspondance est définie à la volée entre la propriété du contrôle affichant la valeur du champ et le champ de la source de données employeesBindingSource correspondant. Le dernier paramètre est utilisé pour la mise à jour des champs, nous verrons son intérêt à la fin de cette démonstration. A noter que le champ Image est récupéré et directement mappé sur le contrôle pictureBox, sans besoin de transformation supplémentaire.

On peut alors compiler et exécuter notre application. Comme on peut le voir dans l'écran suivant, notre Action Panel s'est agrémenté d'informations sur le collaborateur. Evidemment, le passage d'un collaborateur à un autre met à jour l'ensemble des contrôles (figure 21). Lorsque l'application est lancée, essayons par exemple de modifier le



champ « ville » pour la première commande, de supprimer deux commandes dans le tableau, et de passer au collaborateur suivant.

Si on revient sur le collaborateur précédent sur lequel nous avons effectué les modifications, on constate que les modifications ont bien été prises en compte au niveau de l'affichage.

La prise en compte de ces modifications est définie à partir de l'option « Data Source Update Mode » que l'on trouve dans les options avancées des propriétés DataBindings de chaque contrôle lié aux données.

Lorsque cette option est positionnée sur « OnValidation », les mises à jour sont effectuées dans le DataSet sur validation (figure 22).

Ainsi, si les contrôles sont correctement paramétrés, le DataSet est mis à jour automatiquement sans code supplémentaire. Il suffit pour que les mises à jour soient effectives au niveau de la base de données, de demander la mise à jour depuis les informations de ce DataSet.

Nous allons donc créer un nouveau bouton sur notre ActionPanel, qui va nous servir à valider les mises à jour et effectuer les modifications vers la base de données (figure 23).

Nous définissons alors le code suivant sur le click du bouton :

```
private void button3_Click(object sender, EventArgs e)
{
    try
    {
        // Call EndEdit if you are using a BindingSource
        // to commit any changes that are in process.
        Globals.Feuil2.ordersBindingSource.EndEdit();
        Globals.Feuil2.ordersTableAdapter.Update(
        Globals.Feuil2.northwindDataSet.Orders);
    }
}
```

```
catch (System.Exception ex)
{
    MessageBox.Show("Mise à jour échouée");
}
}
```

L'appel à la méthode Update sur l'objet TableAdapter permet de mettre à jour la base de données avec toutes les modifications effectuées au niveau du dataset concernant uniquement la table Orders.

Si on compile le programme, que l'on effectue des modifications sur la liste des commandes et que l'on clique à la fin sur le bouton de mise à jour, celles-ci seront effectuées vers la base SQL Server.

A travers ce dernier exemple, nous avons une application complète qui nous montre les bases des possibilités offertes par les VSTO pour le développement d'application Office. Nous nous sommes axés dans ces exemples sur les nouveautés liées aux VSTO, l'utilisation du modèle Objet Excel, sujet à part entière, n'était pas l'objectif des démonstrations. Un point important à comprendre est que l'objectif n'est pas de créer des applications Office codées de bout en bout en .Net. Il faut combiner les avantages entre ce que savent faire les applications Office et compléter par le code .Net pour tout le reste.

Pour terminer, voici un tableau qui récapitule les différents avantages liés à cette nouvelle version des Visual Studio Tools for Office, par rapport au VBA et à la version précédente des VSTO.

■ Gilles Pommier gillespommier@tiscali.fr
Consultant/formateur MCT Microsoft. Intervenant à la TechEd 2005 sur Visual Studio 2005 et VSTO

Tableau comparatif des solutions pour le développement des applications Office

	VBA	VSTO 2003	VSTO 2005
IDE	Application Office – enregistrement de macros	Application Office (design) et Visual Studio (code)	Visual Studio 2005
Langage de développement	Visual Basic for Applications	Visual Basic .Net (recommandé), C#	Visual Basic .Net, C#
Accès au Framework .Net, aux Services Web	Non	Oui	Oui
Définition de l'interface utilisateur	Contrôles Office	Contrôles Office	Contrôles managés - Visual Studio 2005 Contrôles spécifiques pour Excel et Word
Sécurité d'exécution du code	Aucune - Code embarqué dans le document	Code séparé du document et géré par la CLR	Code séparé du document et géré par la CLR
Stratégie de déploiement	Local	Local/Local, Local/Réseau, Réseau/réseau	Local/Local, Local/Réseau, Réseau/réseau
Déploiement click-Once	Non	Non	Oui
Accès aux données	ADO	ADO.Net	ADO.Net (Développement RAD)
Programmation Volet Office/Smart tag	Non	Complexe	Simple
Modèles de développement disponibles	N/A	Word, Excel	Word, Excel, Outlook, Infopath
Modèle de programmation	Modèle objet Office et VBA	Modèle objet Office et framework .Net	Modèle objet Office et framework .Net
Mode Offline	Non	Non	Oui
Gestion du cache des données	Non	Non	Oui, via les XML Data-Island
Programmation serveur	Non recommandée, installation Office sur le serveur, ne monte pas en charge	Non recommandée, installation Office sur le serveur, ne monte pas en charge	Oui, pas d'installation côté serveur pour Word et Excel

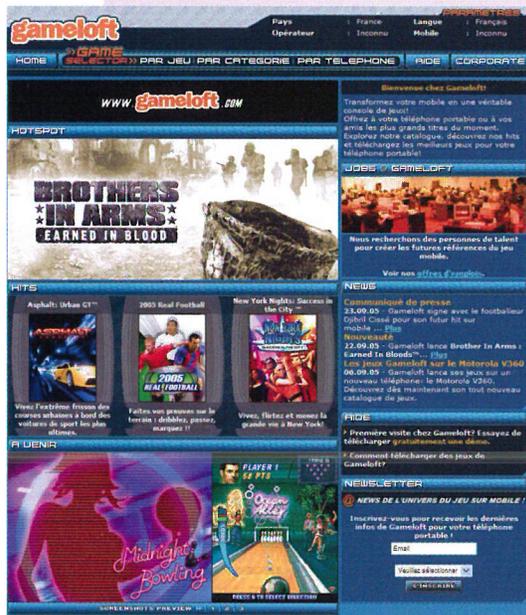
Bizness

Jeuxvideo.fr passe à Clubic

La société lyonnaise Cyréalisis, déjà editrice de l'excellent Clubic.com, vient de racheter le fameux jeuxvideo.fr, le 3^e site français dans le domaine, fort d'un trafic de 30.000 visites par jour.

Gameloft : petits jeux, mais gros sous

Quand on vous dit que le développement de jeux pour téléphones mobiles, c'est l'avenir (ou au moins le présent) ! Gameloft, éditeur international dans ce domaine, vient d'annoncer une augmentation de son chiffre d'affaires de 134% pour le premier semestre 2005. Sa politique d'embauche est aussi significative, de quoi faire rêver : en un an, de juin 2004 à juin 2005, ses effectifs sont passés de 432 à 1375 employés, dont 1200 développeurs. Alors, convaincu ? <http://www.gameloft.com>



Des hits adaptés et des licences à gogo pour jouer en cellulaire.

DEVELOPPEMENT

Power Render 6.4 : 3D spécialisée jeux

Power Render est un SDK d'origine canadienne, de rendu 3D pour les jeux. La version 6.4, pleinement compatible DirectX 9, vient de sortir : en version Pro mais aussi en version Personnel, téléchargeable...

Power Render demande Microsoft DirectX 9 et Visual C++.NET 2003. Plus qu'un moteur, c'est un framework qui entend couvrir l'ensemble des besoins de développement ludique en 3D. Les modules de développement qui le composent forment un ensemble cohérent : un moteur API pour la programmation des applications annexes ou centrales, des bibliothèques, des DLL. Une bibliothèque d'algorithmes et de rendus est d'ailleurs fournie, mais trop limitée à notre goût. En revanche, les fonctions d'import et d'export sont remarquablement complètes et permettent une pleine interaction avec toutes les applications 3D purement graphiques, aussi bien pour l'échange de textures, que d'ombres, de mesh, ou de sprites.

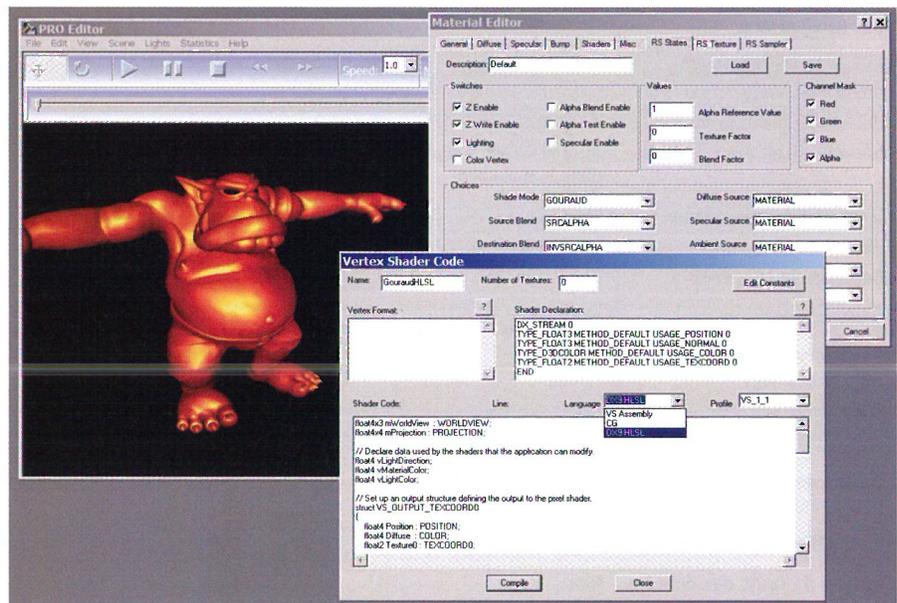
est d'ailleurs très claire et réellement pensée selon les besoins spécifiques des contextes jeux.

On appréciera aussi le module de création de personnages et d'animations depuis des séquences de motion capture, ou encore l'Archive Manager, qui regroupe et compacte aisément toutes vos données 3D dans un format propriétaire à Power Render, mais non bloquant.

La nouvelle version 6.4 axe ses modifications sur des améliorations de vitesse de traitement. Mais surtout, elle se décline maintenant en une édition Personnel, à notre avis bien mieux adaptée à la cible principale de Power Render, c'est-à-dire les développeurs de jeux indépendants. Cette édition est téléchargeable en version d'essai toutes fonctions, seulement limitée en durée (<http://www.powerrender.com>).

OFG : Only For Games

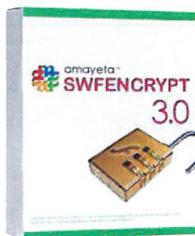
L'interface d'intégration des différents éléments



Les instructions d'ombres et d'éclairages suivent pleinement la logique Objet.

PROTECTION

SWF Encrypt : protection de Flash



Pas de meilleur langage que le Flash pour de beaux petits jeux sur le web. Mais pas de meilleur code non plus pour se faire voler ses créations... Alors il faut mettre un cadenas à la porte : SWF Encrypt.

Les décompilateurs du format Flash SWF de Macromédia ne manquent pas. Quel intérêt ? demanderez-vous. Pour récupérer son code depuis l'application ou le site quand on l'a perdu ? Hum, ce n'est pas si fréquent...

En fait, statistiquement, le problème concerne tout spécialement les jeux en Flash. C'est ainsi que ces décompilateurs sont fréquemment utilisés par des webmasters peu scrupuleux pour aller récupérer le code de jeux Flash d'autres sites. Reste alors à effectuer quelques modifications simples, changer le format de la fenêtre, les couleurs, remplacer des aliens par des vaches et des planètes par des prés, altérer un ou deux scripts et variables, pour obtenir, quasiment sans une goutte de code à soi, un jeu équivalent, sans risque de poursuite.

Code secret pour code secret

SWF Encrypt pose donc un cadenas de cryptage sur votre code ActionScript. La manipulation est enfantine, soit automatique, soit manuelle, en personnalisant les différents niveaux de protection. Selon les tests pratiqués, le mode automatique suffit à décourager la plupart des voleurs : selon les décompilateurs, soit une erreur est directement renvoyée, soit une partie du code ActionScript est seulement extraite (moins de 25%), soit la plupart des variables manquent ou sont illisibles – bon courage au hacker !

On notera hélas que les éléments graphiques et sonores ne sont pas protégés, et c'est dom-

```

1 // Action script...
2
3 // [Initial MovieClip Action of sprite 1]
4 #initclip 1
5 function FUIComponentClass()
6 {
7     this.init();
8 } // End of the function
9 FUIComponentClass.prototype = new MovieClip();
10 FUIComponentClass.prototype.init = function
11 {
12     this.enable = true;
13     this.focused = false;
14     this.useHandCursor = false;
15     this._accImpl = new Object();
16     this._accImpl.stub = true;
17     this.styleTable = new Array();

```

```

1 // Action script...
2
3 // [Initial MovieClip Action of sprite 1]
4 #initclip 1
5
6 #endinitclip
7
8 // [Action in Frame 1]
9
10

```

Décompilation comparative : avant et après protection...

mage. Mais la préservation intégrale du code d'origine est déjà une excellente garantie, une preuve efficace en cas de litige. C'est exactement un cadenas : de quoi décourager les voleurs de passage, mais peut-être pas les grands professionnels...

Editeur : Amayeta, 80 € environ.

Version d'essai téléchargeable.

<http://www.amayeta.com>

Software

Modélisation au pas à pas

Z Brush 2, un des logiciels de modélisation 3D les plus performants en puissance (traitement temps réel en millions de polygones) et finesse (grâce entre autres à son système de calques) a maintenant son Learning Kit, très accessible quel que soit votre niveau. Son Quick Start Guide intégré est un modèle du genre.

<http://pixologic.com>

Microsoft oublie WGF...

La prochaine API de Windows ne s'appellera finalement pas WGF (Windows Graphic Foundation) mais tout simplement... DirectX 10 ! Microsoft a donc décidé de ne pas briser la série dans son appellation, quoique DirectX 10 vienne rompre la traditionnelle compatibilité récurrente avec les précédentes versions : la compatibilité avec DirectX 9,8, et même 7 sera assurée par une surcouche logicielle indépendante, qui devrait donc entraîner quelques ralentissements...

En tout cas, voilà de quoi ajouter quelques cauchemars et nuits blanches aux programmeurs de jeux ! Mais, pour Microsoft, la rupture est nécessaire « pour un double gain, de vitesse et de charge CPU ».

Agenda

Serious Game Summit

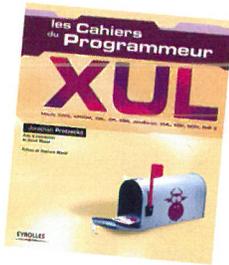
Washington, USA, 31 octobre-1 novembre

Le seul salon international consacré aux développeurs et programmeurs de jeux informatiques « sérieux », c'est-à-dire dans les secteurs de la santé, de l'éducation, de l'armée, de la science ou encore de la communication. Idéal pour s'ouvrir de nouveaux horizons.

<http://www.seriousgamesummit.com>

XUL

- Auteurs : Jonathan Protzenko, Benoît Picaud
- Editeur : Eyrolles
- Prix : 32 €



On parle régulièrement de XUL. Pour mieux appréhender son fonctionnement et la manière de concevoir une interface XUL, cet ouvrage fournit une approche progressive de ce langage. Les différents aspects sont abordés : la structure des fichiers, la décomposition de l'interface, les commandes, l'intégration avec Mozilla et consorts. Si XUL paraît amusant, réaliser une application n'est pas aussi évidente que cela. Il faut maîtriser des notions comme le XPFE, XML, XPCOM ou encore RDF.

Spywares et virus

Protégez vous des logiciels escrocs !

- Auteur : Eric Charton
- Editeur : CampusPress
- Prix : 20 €



Pour bien se défendre, il faut connaître l'ennemi. Ainsi débute la lecture avec la définition de l'infection et ses diverses formes. Les malwares sont désormais omniprésents, et les dérivés ne manquent pas, avec une importance toute particulière sur le spyware. Puis on arrive aux outils pour contrer l'ennemi : anti-virus, anti-malware, firewall. Au-delà des outils, il faut aussi savoir comment nettoyer son PC. Pour se détendre un peu, l'auteur aborde les parasites de la musique. Le réseau P2P et les fichiers téléchargés deviennent une menace difficile à contrôler. Et si après tout ça vous n'avez pas d'idée de mise en place d'une stratégie de sécurité, vous aurez à la fin du livre quelques pistes... Malgré l'approche pédagogique, deux lacunes sont à noter : l'ouvrage concerne uniquement Windows et n'évoque pas les nouvelles attaques de type rootkit. Un plus avec la présence d'un Cd Rom.

SQL en concentré 2^e édition

- Auteur : collectif
- Editeur : O'Reilly
- Prix : 48 €



SQL demeure souvent une torture pour le développeur. La donnée et sa manipulation n'ont jamais été une grande spécialité du codeur. Mais SQL demeure incontournable dans bien des cas. Donc, autant le connaître. L'ouvrage couvre les principales bases (SQL Server, DB2, MySQL, Oracle et PostgreSQL). Très complet, les auteurs ont segmenté SQL par chapitres. On débute par un court historique, puis on plonge dans les fondamentaux du langage avant d'aborder les mots-clés. En annexe, on trouvera ceux qui sont spécifiques à chaque SGBD. Tout au long des mots-clés, les différences entre les SGBD sont indiquées et expliquées et on dispose systématiquement de conseils de programmation pour utiliser au mieux le mot et éviter les pièges, exemples à l'appui. Un ouvrage de qualité.

Design Patterns

- Auteur : collectif
- Editeur : O'Reilly
- Prix : 54 €

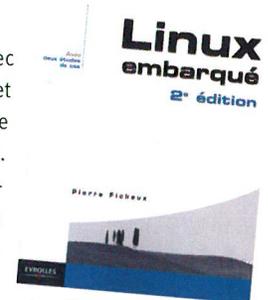
Comment bien coder son application ? Comment bien définir l'architecture de son application ? Tout cela peut se résumer à deux mots : Design Pattern. Les design patterns permettent de définir les bonnes choses à utiliser dans une architecture, un code. Mais cette technique exige pratique et savoir-faire pour déterminer le design pattern adéquat à tel moment. Un art pas toujours simple. Au-delà, les auteurs mettent aussi en avant le principe du MVC (Model-View-Controller). Au lieu d'avoir une approche austère, l'ouvrage propose un apprentissage par le fun en évitant de se prendre trop au



sérieux, même si le contenu l'est. Surtout, l'ensemble des patterns sont décrits, expliqués, mis en œuvre. Au final, un ouvrage visuel et amusant à lire. Toujours de qualité, il n'est cependant pas pour le développeur débutant. Le prix est amplement justifié.

Linux embarqué 2^e édition

- Auteur : Pierre Ficheux
- Editeur : Eyrolles
- Prix : 39 €



L'embarqué avec Linux existe bel et bien et fonctionne même très bien. Cette seconde édition réactualise son contenu technique en tenant compte des évolutions depuis fin 2002. On débute par la définition du système embarqué : le système, l'architecture, le logiciel. Puis l'auteur explique de bout en bout de quelle manière créer un Linux embarqué et les techniques d'optimisation. Pour être plus concret, plusieurs systèmes et outils sont longuement exposés : uClinux, RTLinux, Crosstool, etc. La dernière partie propose deux études de cas. Si un tour d'horizon des autres systèmes embarqués est proposé, il aurait été bienvenu de l'approfondir. Au demeurant, l'ouvrage constitue une bonne introduction, claire et assez complète.

Et aussi :

- **Linux Fedora Core 4** (Eni, 27,14 €) : la fondation servant dans la distribution Red Hat arrive en version 4. Ce livre intéressera les administrateurs et surtout l'utilisateur avancé de Fedora. On trouve les classiques du genre : gestion de disques, des paquetages, des utilisateurs, des ressources système. Rapide à lire, il apporte le minimum vital. Il est regrettable que la distribution ne soit pas incluse...

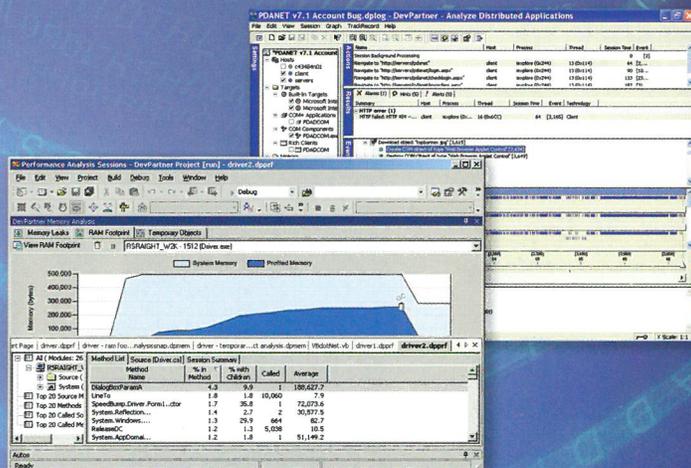
- **XML par la pratique** (Eni, 27,14 €) : rien ne vaut la pratique pour connaître XML et ses dérivés. Au fur et à mesure des pages, l'auteur décrit les différents composants de XML, des fichiers XML et des dérives à connaître tels que XHTML ou XSLT. Un bon livre de chevet.

Développez des applications fiables, sécurisées et robustes avec Compuware DevPartner

Gagnez jusqu'à 30% sur vos délais de développement en accélérant la mise au point du code et en y apportant sécurité, performance et fiabilité avec Compuware DevPartner. Les solutions Compuware DevPartner couvrent tous les environnements Microsoft : DNA, Visual Studio 2003 et Visual Studio 2005

DevPartner Studio Professional Edition : pour des développements de qualité

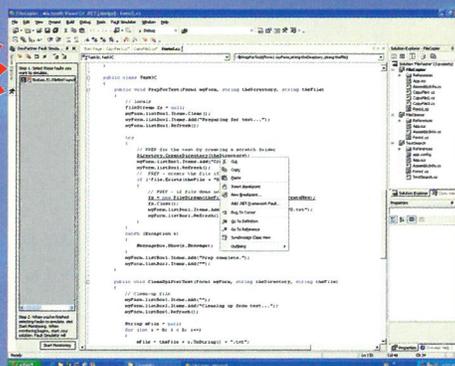
- Analysez et optimisez la mémoire sous .NET
- Détectez automatiquement les erreurs en DNA, .NET, Visual Studio 2003 et 2005
- Diagnostiquez en architecture distribuée multi-serveurs
- Détectez et analysez les deadlocks



Nouveau

DevPartner Fault Simulator : le « crash test » de vos applications !

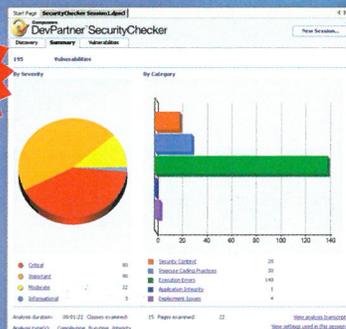
- Simulez les erreurs applicatives et systèmes sur votre code
- Testez et mettez au point vos codes de gestion d'erreurs
- Obtenez le détail des erreurs grâce à une analyse précise
- Testez la robustesse de votre code en simulant des erreurs



Nouveau

DevPartner SecurityChecker : « L'anti-hacker » !

- Détectez et corrigez les failles de sécurité lors du développement
- Identifiez les failles de sécurité à l'exécution
- Vérifiez l'intégrité de votre application web



Pour plus d'information, contactez-nous au **0800-912141** ou visitez notre site www.compuware.fr

COMPUWARE
www.compuware.com



ATELIER DE GÉNIE LOGICIEL PROFESSIONNEL

UN CODE UNIQUE :

Windows, .Net, Java, PHP, J2EE,
XML, Internet, Intranet,
Pocket PC, SmartPhone, Client léger ...

151
Nouveautés

à découvrir sur www.pcsoft.fr
ou lors du Tour de France



**VENEZ DÉCOUVRIR
WINDEV 10 &
WEBDEV 10
PRÈS DE CHEZ VOUS**

Montpellier: **8 Nov**

Genève: **15 Nov**

Lyon: **16 Nov**

Strasbourg: **17 Nov**

Nantes: **22 Nov**

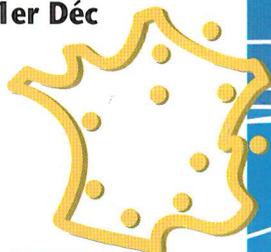
Bordeaux: **23 Nov**

Toulouse: **24 Nov**

Paris: **29 Nov**

Bruxelles: **30 Nov**

Lille: **1er Déc**



GRATUIT
inscrivez-vous
sur www.pcsoft.fr

WINDEV [®] **N°1 EN FRANCE**

DÉVELOPPEZ 10 FOIS PLUS VITE

**OPÉRATION
1 PC + 1 ÉCRAN PLAT 17" + 1 IMPRIMANTE
POUR «1 EURO DE PLUS»**

Achetez WINDEV ou WEBDEV ou WINDEV Mobile entre le 17 octobre et le 16 décembre 2005 et **recevez un PC + un écran plat + une imprimante multifonction de marque DELL pour 1 euro de plus.** WINDEV ou WEBDEV ou WINDEV Mobile doit être acheté chez PC SOFT, référence WD91EE ou WM91EE ou WB91EE.

Détails sur www.pcsoft.fr ou appelez-nous. Offre réservée aux sociétés et administrations.

Le Logiciel et le PC peuvent être acquis séparément: WINDEV ou WEBDEV ou WINDEV Mobile au tarif de 1650€ hors taxes, le PC seul au tarif de 1002 €TTC, l'écran seul au prix de 279 €TTC, l'imprimante au prix de 124 €TTC ou la configuration matérielle complète 1.223 €TTC

(tous ces tarifs sont modifiables sans préavis, la livraison est incluse)

PC de marque DELL, gamme Optiplex, Mini Tour. Configuration minimum : Processeur: 2,53 Ghz Intel Celeron bus 533; Ecran 17 pouces XGA; Mémoire centrale 512 Mégas; 6 ports USB; Carte Ethernet 10/100; Graveur DVD+/-RW 16x; Disque dur 80 Go SATA 2; Windows XP Home; Imprimante multifonction 924,

avec photocopieur couleur & noir et blanc, scanner 48 bits, impression 4.800 x 1.200DPI, jusqu'à 16 pages par mn en couleurs, mode photo. Garantie DELL; PC et écran: 3 ans sur site; imprimante: 1 an.



PC SOFT

Document non contractuel. Logiciel professionnel. Versions 10 annoncées. Tout est en français. Séminaire professionnel, nombre de places limité.