

Choc fratricide dans l'open source

Les forks vont-ils tuer les projets open source ?

- MySQL, menacé par MariaDB et SkySQL ?
- Les challenges de Mandriva
- LibreOffice veut-il la mort d'OpenOffice ?

EXCLUSIF :
MANDRIVA, nouvel O.S. DE LA RUSSIE ?

**PDC, TechEd,
 Forum PHP,
 Adobe Max 2010**

Toutes les dernières
 annonces !

Webmaster

Adapter son
 contenu web aux
 smartphones

**Microsoft
 réinvente l'IDE**

Visual Studio
 Lightswitch
 et Webmatrix :
 tout simplifier !

Carrière

**Profils précieux
 pour recruteurs :**
 Ingénieur sécurité,
 Ingénieur développement,
 Chef de Projet

Êtes-vous un DÉVELOPPEUR DURABLE ?

Créez des *logiciels "verts"*: écologiques,
 économes en énergie, respectueux de la nature...

Mobile

Découvrez Samsung Bada

Google

Développer un module Chrome

Pylons

Un framework surpuissant en Python

Tablette

iPad : ce qui change par rapport à l'iPhone

M 04319 - 136 - F: 5,95 €



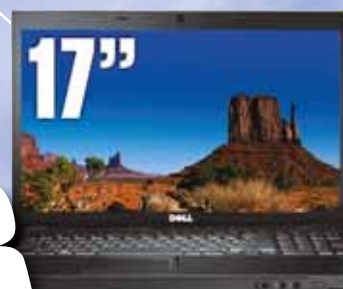
Printed in France - Imprimé en France - BELGIQUE 6,45 €
 SUISSE 12 FS - LUXEMBOURG 6,45 € - DOM Surf 6,90 €
 Canada 8,95 \$ CAN - TOM 940 XPF - MAROC 50 DH

JUSQU'AU 19 DÉCEMBRE

WINDEV®

NOUVELLE
VERSION

16



Portable DELL
Vostro 3700 BTS
puissant Processeur **Intel**
Core i5-560
Mémoire **4Go**
Disque **500 Go**
Wifi, Bluetooth, graveur DVD,
Webcam 2M,
empreinte digitale,...
Windows 7 pro



**Station de
travail DELL**
Precision T1500;
puissant Processeur
Intel Core i5-750
Mémoire **4Go**
Disque **1.000 Go**
Ecran 20p
Graveur DVD/Bluray
Windows 7 pro
64bits

OPÉRATION

1 PC POUR 1 EURO DE PLUS

OU

**ACHETEZ WINDEV 16 ET
RECEVEZ UN PC DELL
POUR 1 EURO DE PLUS
RENDEZ-VOUS SUR
WWW.PCSOFT.FR**

Offre réservée aux entreprises, administrations, collectivité, indépendants, GIE, associations,... en France métropolitaine. Chaque élément de l'offre peut être acquis séparément. Tous les détails sont sur www.pcsoft.fr.



Elu «Langage le
plus productif du
marché»

Environnement de développement professionnel totalement intégré, qui couvre l'intégralité du cycle de développement.

Windows, .Net, Linux, Mac, Internet, Intranet, Java, PHP, Android, Windows Phone 7,...

Fournisseur Officiel de la Préparation Olympique



► **Dossier gratuit 200 pages** sur simple demande. Tél: **04.67.032.032** info@pcsoft.fr

www.pcsoft.fr

\\ actus

En bref	6
Mandriva : l'O.S. de la Russie en 2011 ?	13

\\ événement

Résumé du Forum PHP 2010	10
PDC : Azure, asynchrone et Silverlight	11
Adobe Max 2010: Adobe présente à tout va !	12

\\ webmaster

Optimisez vos applications Flash pour les mobiles	14
---	----

\\ sgbd

SQL Server 2011 : automne 2011 !	18
MySQL- Proxy : le proxy MySQL qui vous simplifie la vie	20

\\ gros plan

Modélisation : Simplifiez-vous le code (2^e partie)

Une semaine d'immersion dans une équipe d'ingénierie par les modèles	23
Modélisation : simplifier le problème du développeur	26
Créer son outil de modélisation de processus métier en utilisant Workflow Foundation 4	28

\\ enquête

Les forks nuisent-ils à l'Open Source ?

Fork ou pas fork, le point de vue de l'AFUL	33
M(ariaDB) ou M(ySQL) : le choix entre l'original et le fork ?	36
Mandriva vs Mageia : un nouveau challenge pour Mandriva	38

\\ dossier

Le développeur se met au vert

Le GreenIT moteur de l'entreprise	41
Le développeur est la clé du greenIT	42
Le Lean et le logiciel vert	43
Programmation Java : des Green Patterns	45
Comment créer des applications plus performantes et moins gourmandes	49

\\ carrière

Certains profils sont précieux pour les recruteurs	52
3 fiches métiers : ingénieur sécurité	52
ingénieur développement et chef de projet	54

\\ architecture

FraSCAti, prenez le contrôle de vos applications	56
--	----

\\ code

WebMatrix : simplifiez-vous le développement web	60
Visual Studio Lightswitch : IDE nouvelle génération ?	63
Silverlight Pivot : une autre manière de naviguer sur le web !	66
iPad : qu'est ce qui change pour le développeur	68
Découverte de Samsung Bada (1 ^{re} partie)	70
Pylons, un nouvel outil de développement Web agile	73
Ecrire des extensions pour le navigateur Google Chrome	78

\\ temps libre

Les livres du mois	82
--------------------------	----



6



12



23



32



40



70



68

L'info continue sur www.programmez.com

CODE

Les sources
des articles

NOUVEAU

Livres blancs :
langages, outils...

TÉLÉCHARGEMENT

Les dernières versions de vos
outils préférés + les mises à jour

QUOTIDIEN

Actualité, Forum
Tutoriels, etc.

Advanced database technology for breakthrough applications



Laissez vos applications s'envoler...

Avec InterSystems Caché, faites décoller vos applications. Elles bénéficieront immédiatement de performances hors du commun, deviendront massivement scalable et ne nécessiteront plus d'administration fastidieuse.

InterSystems **Caché**® base de données post-relationnelle, mais aussi serveur d'application, framework Ajax, ... se fonde sur une technologie Objet avancée qui permet de construire beaucoup plus facilement des applications XML, Web Services, AJAX, Java et .NET.

InterSystems Caché est aussi une base SQL jusqu'à 5 fois plus rapide que les bases de données relationnelles classiques en accès SQL et bien plus en accès Objet !

Grâce à son Architecture de Données Unifiée unique, Caché élimine le

besoin de mapping objet-relationnel, réduit les temps de développement, et facilite l'évolution et la maintenance de votre application.

Déployé sur plus de 100.000 systèmes de par le monde pour des applications de 2 à 50.000 utilisateurs, Caché est disponible sur toutes les plates-formes majeures du marché.

Nouveau: L'intégration de Caché avec JAVA - JNI permet littéralement d'exploser les performances et d'offrir enfin aux programmeurs JAVA un moteur sans compromis et digne de leurs réalisations.

Depuis plus de 30 ans InterSystems vous apporte des technologies avancées qui vous permettent de construire des applications qui font la différence.

INTERSYSTEMS

Téléchargez votre version gratuite complète InterSystems Caché - sans limite de temps: InterSystems.fr/avancee

Exclusif :

Mandriva donnera t-il naissance à l'O.S. de la Russie en 2011 ?

Programmez dévoile qu'un OS issu du français Mandriva sera peut-être bientôt installé sur chaque PC scolaire russe.

➔ Lire l'article page 13



Le langage de tous les vices

Depuis une quinzaine d'années les langages de développement suivent une courbe double et inquiétante.

Qui n'a jamais entendu : « avec la version 3 de notre langage, le développeur réduit de 70 % ses lignes de code » ou encore, « on introduit plusieurs centaines de nouveautés, des nouvelles API par dizaines mais on en retire aussi certaines », etc. Tout en martelant : « on simplifie, on vous aide à coder, on se rapproche du langage naturel ». En bon observateur, nous sommes de plus en plus sceptique ... L'embonpoint touche tous les langages. Java a certes fait le ménage depuis la JDK 5 mais chaque version rajoute, décale des fonctions, au point de ne plus trop savoir ce que contiendra la v7. PHP met à mort la v6 mais annonce une v5.4 qui devrait finalement récupérer la plupart des nouveautés attendues. Sans oublier C#, VB, Ruby... qui évoluent régulièrement. A côté, l'évolution de C++ paraît sage, voire préhistorique. Mais l'avantage de cette lenteur est que le langage est normalisé et n'introduit que des nouveautés jugées utiles. A chaque nouvelle version, le développeur doit se mettre à niveau, désapprendre des réflexes, changer d'outils, vérifier que son code passe bien, etc. Que de temps perdu !

“ Stop ! Arrêtons de faire ressembler les langages à des plomberies tellement compliquées... ”

Nous disons : Stop ! Arrêtons de faire ressembler les langages à des plomberies tellement compliquées que l'on perd en compréhension, en performances et finalement en productivité. Si en plus on y rajoute de la programmation parallèle, de l'agilité, du fonctionnel, du multi terminal, il y a de quoi se mettre au jardinage. Il devient urgent que les éditeurs, les communautés en charge des langages finissent par comprendre qu'il faut simplifier et surtout améliorer la productivité du développeur !

Il existe bien quelques projets de langages simplifiés mais ils restent désespérément marginaux. Mais il ne faut pas s'arrêter aux langages, que dire des IDE ? Quand on voit les lourdeurs, la complexité de plus en plus alarmante des Eclipse, NetBeans, Visual Studio, on se met à regretter HyperCard, VB 3. Là aussi, le développeur peut perdre ou gagner plusieurs heures de travail... et quelques cheveux...

Dans ce numéro, nous avons voulu mettre en avant deux projets intéressants et utiles de Microsoft : Visual Studio Lightswitch et Webmatrix. Webmatrix est particulièrement excitant pour le développeur débutant, ou tout un chacun à la maison ayant quelques connaissances en programmation. Cet environnement permet de créer, de coder et de déployer en quelques clics une application, un site, avec des templates projets. Bien sûr nous sommes loin de la programmation par métaphore mais nous trouvons que Webmatrix a le mérite de cacher la complexité. Qui n'a jamais voulu créer un site web avec Dreamweaver ou Visual Studio ? Même le site le plus basique nécessite la maîtrise de nombreuses fonctions de l'IDE et de langage. Cette démarche est la bienvenue et nous ne pouvons que l'encourager.

A l'époque de VB, les développeurs puristes (ou se considérant comme tels) parlaient de VB comme d'un langage Lego, avec lequel on ne « développait » pas réellement. Bref on disposait d'un sous-langage... Mais avec du recul, 15 ans plus tard, la question est : n'avons-nous pas intérêt à opter pour la programmation Lego ?

■ François Tonic

Rédacteur en chef - ftonic@programmez.com

Editeur : Go-02 sarl, 21 rue de Fécamp 75012 Paris - diff@programmez.com.

Rédaction : redaction@programmez.com

Directeur de la Rédaction : Jean Kaminsky.

Rédacteur en Chef : François Tonic - ftonic@programmez.com.

Ont collaboré à ce numéro : F. Mazue, F. Dewasmes.

Experts : C. Pierre de Geyer, G. Madison, M. Chaize, T. Imbert, N. Balaci, A. Henry, C. Anderson, J. Balouka, C. Villeneuve, D. Seguy, J-Ph. Gouigoux, L. Laske, G. Plouin, L. Bar, C. Demarey, D. Fournier, E. Vernié, A. Detrie, P. Couzy, F. Bordage, F. Bellahcène, D. Martin.

Illustrations couverture : © iStockphoto/Inok, © iStockphoto/Tomml.

Publicité : Régie publicitaire, K-Now sarl. Pour la publicité uniquement : Tél. : 01 41 77 16 03 - diff@programmez.com.

Dépôt légal : à parution - Commission paritaire : 0712K78366 ISSN : 1627-0908. Imprimeur : S.A. Corelio Nevada Printing, 30 allée de la recherche, 1070 Bruxelles Belgique. Directeur de la publication : J-C Vaudecrane

Abonnement : Programmez 22, rue René Boulanger, 75472 Paris Cedex 10

Tél. : 01 55 56 70 55

abonnements.programmez@groupe-gli.com

Fax : 01 40 03 97 79 - du lundi au jeudi de 9h30 à 12h30 et de 13h30 à 17h00, le vendredi de 9h00 à 12h00 et de 14h00 à 16h30. Tarifs

abonnement (magazine seul) : 1 an - 11 numéros France métropolitaine : 49 € - Etudiant : 39 € - CEE et Suisse : 55,82 € - Algérie, Maroc, Tunisie : 59,89 € - Canada : 68,36 € - Tom : 83,65 € Dom : 66,82 € - Autres pays : nous consulter. PDF : 30 € (Monde Entier) souscription exclusivement sur www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

www.programmez.com

L'INFO PERMANENTE
WWW.PROGRAMMEZ.COM



PROCHAIN NUMÉRO
N°137 janvier 2011
parution 30 décembre

✓ **Java 7**
Enfin de bonnes nouvelles et un avenir !

✓ **Webmaster**
Avec Twitter, RSS, RestFull... gardez le fil du Web

■ **Softfluent** a dévoilé mi-novembre CodeFluent Entities Modeler RC1, nouvelle version de l'usine à logiciels de l'éditeur. Elle dispose d'une interface reprenant le ruban introduit avec MS Office et présent dans plusieurs logiciels Microsoft. CodeFluent Entities propose une méthode structurée et les outils correspondants pour développer des applications avec un niveau de productivité inégalé. Le produit permet la création de composants par simple description, utilisant les patterns de programmation basés sur les meilleures pratiques et proposant de nombreuses architectures prêtes à l'emploi. Site : www.softfluent.com

■ **Amazon** présente Cluster **GPU Instances**, pour tirer profit de la GPU depuis Amazon EC2. Il s'agit d'une nouvelle instance cloud disponible pour EC2, qui permet d'utiliser la GPU dans une utilisation HPC ou de faire du calcul non intensif pour décharger la CPU.

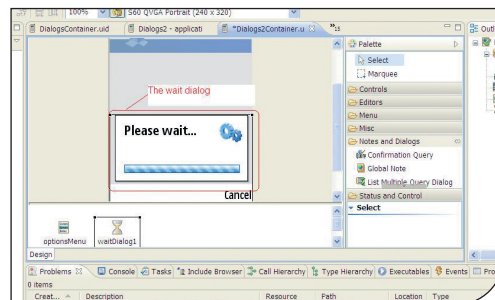
■ **ASP.Net MVC 3.0** va très rapidement être disponible en version finale. Une mouture qui se distingue notamment par Razor Intellisense, c'est-à-dire l'intégration dans Visual Studio et Visual Web Developer Express de Razor, le nouveau moteur de templates, associé à la coloration syntaxique et l'achèvement de code. C'est aussi l'intégration du packager open source NuGet qui permet de facilement délivrer des applications faisant appel à des bibliothèques tierces

Smartphone

Nokia veut sa part du gâteau

Le premier constructeur de téléphone portable, Nokia, est à la peine sur le marché des Smartphones. Bien que ce marché soit marginal par rapport au marché global du mobile, Nokia ne pouvait se permettre de laisser Blackberry, iPhone, Android, Bada et autres Windows Phone 7 seuls. Jusqu'à présent, le constructeur avait du mal à être lisible auprès des développeurs : plusieurs modèles de développement, des gammes pléthoriques. Aujourd'hui, une des priorités de Nokia est de renforcer son kit de développement autour de Nokia Qt SDK, simplifier l'expérience utilisateur et le concept Ovi Store, la boutique en ligne des applications. Enfin, faciliter aux développeurs l'accès aux informations, aux outils.

Si Nokia peut se prévaloir de 175 millions de téléphones Symbian, le développement n'était pas tou-



jours très simple, ni homogène. Aujourd'hui, la base technique commune sur les nouveaux modèles est la librairie Qt. « *Les développeurs découvriront en Nokia Qt SDK l'outil le plus complet et moderne du marché offrant un flux de travail optimisé, un développement simplifié et des bibliothèques d'interfaces utilisateurs intuitives ; autant d'atouts qui permettent de réduire de 70 % le nombre de lignes de codes et d'accélérer très sensiblement la création* », commente le constructeur. Les nouvelles opportunités offertes aux développeurs permettront la fonctionnalité « In App purchase » (l'achat de contenu depuis une applica-



tion), assortie d'un large éventail d'options de tarification des applications sur Ovi Store, comme les modèles d'abonnement, les micro-transactions ou l'option « essayer avant d'acheter ». Si l'utilisateur est séduit par la version gratuite de l'application, il pourra facilement demander une mise à niveau vers la version payante.

Cela permet également aux développeurs d'apporter une valeur ajoutée à leurs applications, comme la vente de produits virtuels, des niveaux supplémentaires pour leurs jeux ou des fonctionnalités améliorées ou localisées « In App ».

Le Qt SDK combine Qt et les outils de développements attachés à Qt. Il s'utilise pour les systèmes Symbian, Maemo et le futur MeeGo. Le développeur peut l'utiliser sur Windows, Linux et MacOS X. Il possède un émulateur de mobile, une fonction de compilation distante, et deux IDE complets (Qt Designer et Qt Creator). Le langage de référence du kit demeure le C++. Ce SDK vise les gammes suivantes : N8, E7, C7 et N900. Par contre pour les systèmes Symbian, le développeur devra toujours s'accommoder du Symbian SDK. Nokia met aussi à la disposition des développeurs un environnement de programmation complet, pour Symbian et Qt : Carbide.c++. Il fonctionne uniquement sur Windows.

Outre les améliorations apportées au Nokia Qt SDK pour les smartphones et ordinateurs portables, Nokia annonce également un kit de développement logiciel pour les Series 40 Touch and Type. Il s'agit du premier SDK tactile pour téléphones mobiles.

Pour en savoir plus :

<http://www.forum.nokia.com/Develop/Qt/>

agenda \

DECEMBRE

• La Nuit de l'info 2010 aura lieu les 2 et 3 décembre 2010, de 16h40 à 08h00. A l'image des éditions précédentes, elle réunira des étudiants de toute la France, pour une grande aventure collective, pour un temps de fête, pour un mélange d'informatique, de

communication, de marketing, d'extrême programming, de modélisation, de pizzas, de café, de glaces, de musique, de films... <http://www.nuitdelinfo.com>

• Le 08 décembre, Microsoft France - Centre de Conférences 41 Quai du Président Roosevelt - 92130 Issy Les Moulineaux. Séminaires Business Intelligence en Libre Service avec Microsoft SQL Server 2008 R2 et PowerPivot. Comment concilier

autonomie des utilisateurs et pérennité de vos applications décisionnelles pour la Direction des Systèmes d'Information avec la plateforme SQL Server 2008 R2 (SSIS / SSAS / SSRS) et PowerPivot. <http://www.finelog.fr>

• Tour de France Windev : Genève le 1er décembre, Lyon le 2 décembre, Marseille le 07 décembre www.pcsoft.fr

DESIGN

Concevez des applications qui vous aident à diriger votre business



Notre contrôle™ xamMap dans Silverlight et WPF vous permet de tracer toutes les données géospatiales dont vous avez besoin, comme ce plan de cabine d'avion vous permettant de gérer votre business ! Venez l'essayer dès aujourd'hui sur infragistics.com



NetAdvantage® ULTIMATE

for ASP.NET, Windows Forms, WPF, Silverlight, WPF Data Visualization, Silverlight Data Visualization

Infragistics®

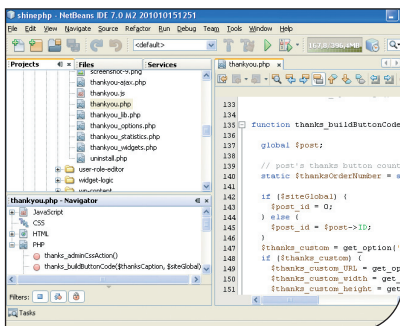
Infragistics Ventes France **N°1 Vert**
0800 667 307 • Infragistics Europe Ventes
+44 (0) 800 298 9055 • [@infragistics](http://t.infragistics)

■ **Infragistics** lance un nouveau package de composants .Net et d'interface : NetAdvantage Ultimate. Il regroupe tous les composants d'interface de l'éditeur dans une seule unique solution. Si le prix peut paraître cher : 1895 \$, on économise plus de 4000 \$ par rapport à l'achat des packages séparément ! Site : <http://www.infragistics.com/dotnet/netadvantage/new.aspx>

■ **Oracle** lance Solaris 11 Express 2010.11. Cette version, uniquement dédiée au développement est gratuite. Elle fonctionne sur Sparc et x86. Les grandes améliorations sont : temps de démarrage réduit, ZFS avec cryptage, nouvelles briques de virtualisation, administration serveur et déploiement améliorés. Le fork issu d'OpenSolaris, Illumos, intégrera sans doute ces nouveautés.

■ L'outil **Mercurial** arrive en version 1.7. Cette version introduit un grand nombre de changements : les noms des projets peuvent contenir désormais des espaces, commande shell dans les alias, fonction merge améliorée avec l'ajout de -tool pour indiquer l'outil que l'on souhaite utiliser, etc.

Site : <http://mercurial.selenic.com/>



■ **Netbeans 7** avance. Le projet a dévoilé les premières pré-version de la v7. Cette version intégrera Java 7 (en pré-version) et surtout le développeur web bénéficiera d'un meilleur support de PHP ainsi que les premiers supports de HTML5.

Site : <http://netbeans.org/>

ALM

ALM France User Group : pour mieux appréhender les enjeux du cycle de vie

Le 4 novembre dernier, le groupe utilisateur ALM Group a ouvert ses portes durant une soirée inaugurale. Le but du groupe est de promouvoir, expliquer, discuter autour des environnements et de la stratégie ALM (Application Lifecycle Management). Si le marché commence à mûrir, la France attendait, encore en retard sur la gestion applicative et sur les stratégies autour des usines logicielles.

Durant la soirée, l'auditoire a pu en savoir sur plus les outils ALM dans le monde open source, l'approche modèle de Softfluent ou encore les tendances et perspectives expliquées par Fran-

çois Tonic (Programmez !). Une question s'est posée à la suite des conférences : est-ce que agilité et ALM sont liées, complémentaires ou opposées ? Pour François Tonic, un ALM « nouvelle génération » doit être agile, mais cette vision ne fut pas partagée par tout le monde. C'est aussi ça l'objectif de l'ALM France : lancer des débats !

Dès janvier 2011, le groupe organisera une session par trimestre autour d'un retour terrain, d'une présentation technique ou plus générale. Venez nombreux !

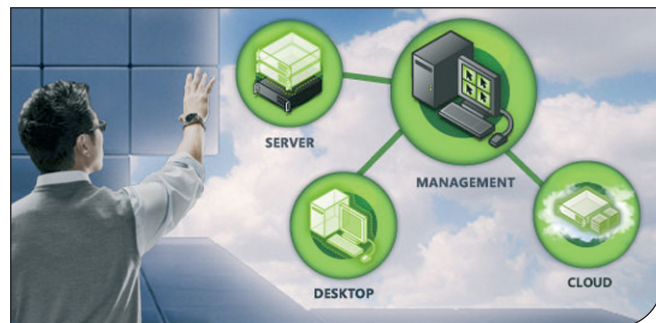
Site : <http://blog.alm-france.org/>

TechEd Europe 2010

Microsoft devient un éditeur 3 couches du cloud

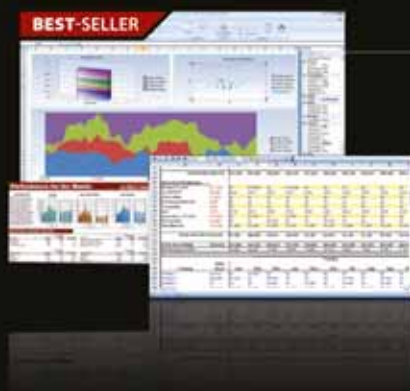
Après avoir dévoilé une nouvelle approche autour du cloud public avec Windows Azure, Microsoft complète son offre en rajoutant une couche IaaS avec Hyper-V Cloud. Cette offre se base sur Hyper-V, System Center par l'administration, Windows Server 2008 R2. L'approche cloud privé se compose de "Fast Track" qui permet aux utilisateurs, entreprises, de construire leur cloud privé.

Cette offre est prise en compte par plusieurs constructeurs tels que Hitachi, Dell, IBM. Provider Program vise, comme son nom l'indique, les hébergeurs et infogéneurs. 70 providers participent déjà au programme. A cela se rajoutent des guides, des bonnes pratiques. Il s'agit là d'une stratégie importante pour Microsoft car elle complète son offre cloud en en faisant un acteur 3 couches : SaaS (Online services), PaaS (Azure) et maintenant IaaS, comme peut l'être un VMware.



Cependant, on peut s'étonner du silence autour de Windows Azure Appliance qui doit aussi être une autre mi-PaaS, mi-IaaS autour de Windows Azure. Hyper-V Cloud et Appliance répondent, selon Microsoft, à deux stratégies différentes. Appliance vise plutôt les grands comptes, les entreprises ayant des besoins importants en montée en charge alors que Hyper-V Cloud convient aussi bien à de petites entreprises qu'aux grosses et c'est un pur IaaS. Mais ce point méritera des éclaircissements surtout au niveau API. D'autre part, Microsoft a profité de TechEd pour dévoiler les packs d'administration pour Windows

Azure : Windows Azure Application Monitoring Management Pack, Opalis 6.3 (racheté en 2009). Opalis est intégré à System Center et propose d'orchestrer et de gérer les process de virtualisation. Notons aussi une session entière consacrée au futur de C# avec Visual Studio Asynchrone. Cette approche est, selon Microsoft, la prochaine grande évolution de la programmation afin de répondre aux problèmes de montée en charge, de réactivité de l'interface et de latence dans les applications. F# a eu droit à sa session, surtout depuis son passage en licence Apache qui devrait aider à son utilisation.


Spread for Windows Forms à partir de € 702

Feuille de calcul pour les applications Windows Forms, compatible avec Microsoft Excel.

- Accélérez le développement avec les concepteurs de feuilles de calcul, l'Assistant de prise en main et les concepteurs de graphiques
- Renseignement automatique : anticipation de la frappe dans la cellule
- Nouveau - outil intégré de création de diagrammes avec 85 styles
- Nouveau - préserve les .XLS et restaure les fonctions non supportées
- Inclut des apparences prédéfinies ainsi que la possibilité de créer des apparences personnalisées


FusionCharts à partir de € 140

Graphiques Flash & JavaScript (HTML5) interactifs pour les applications Web.

- Animez vos applications Web avec des graphiques interactifs et pilotés par les données
- Créez des graphiques AJAX avec des possibilités d'exploration en quelques minutes
- Exportez les graphiques au PDF et les données en CSV directement depuis les graphiques
- Créez des jauges, des tableaux de bord, des graphiques financiers et plus de 550 types de carte
- Adopté par plus de 17 000 clients et 330 000 utilisateurs dans 110 pays


IBM ILOG Elixir Standard à partir de € 798

Étendez les plateformes Adobe Flex et AIR avec des contrôles de visualisation interactifs.

- 10 contrôles d'UI : graphique 3D, jauge, carte, carte de clics, calendrier, OLAP et tableau croisé dynamique, organigramme hiérarchique, arborescence, graphique en radar et chronologie
- Intégration transparente avec l'environnement de développement intégré d'Adobe Flash Builder 4
- Controls can be used individually or in combination, alongside other Flex components
- Utilisation individuelle ou combinée des contrôles avec les autres composants Flex


Nevron Chart for .NET Enterprise à partir de € 624

Fonctionnalités riches de création de tableaux à vos applications Windows Forms et ASP .NET.

- Graphiques en 2 et 3D : histogrammes, lignes, escaliers, aires, secteurs, points, bulles, bourse, flottants, radar, polaires, max./min., mailles, grilles, formes, courbes lisses, barres flottantes, Venn et erreur. Nombreuses variations : nuages de points XY et XYZ, barres empilées, etc.
- Axes avec barre de défilement, légende intégrée, annotations et filigranes sur graphique
- Support de la conception Visual Studio, nombreuses fonctionnalités avancées (éditeurs de style)

© 1999-2010 ComponentSource. Tous droits réservés. Tous les prix sont corrects au moment de la presse. Prix en ligne mai différentes de celles décrites en raison de fluctuations quotidiennes et remises en ligne.

 Siège social en Europe
 ComponentSource
 30 Greyfriars Road
 Reading
 Berkshire
 RG1 1PE
 Royaume-Uni

 Siège social aux États-Unis
 ComponentSource
 650 Claremore Prof Way
 Suite 100
 Woodstock
 GA 30188-5188
 États-Unis

 Siège social au Japon
 ComponentSource
 3F Kojimachi Square Bldg
 3-3 Kojimachi Chiyoda-ku
 Tokyo
 Japon
 102-0083

 Numéro vert:
0800 90 92 62
www.componentsource.com

Nous acceptons les bons de commande. Contactez-nous pour demander un compte de crédit.





© Julien Pauli

Résumé du Forum PHP 2010

Les 9 et 10 Novembre derniers l'association française des utilisateurs de PHP (www.afup.org) organisait l'évènement francophone phare PHP : le Forum PHP. Cette année ce n'était pas une édition ordinaire puisque PHP fêtait ses 15 ans tandis que l'AFUP soufflait sa 10^e bougie.

Près de 550 personnes se sont inscrites pour participer au Forum PHP. L'AFUP n'en attendait pas tant puisqu'un mois avant l'évènement, le nombre maximum de visiteurs était atteint. Suite à des discussions internes, le président de l'association, Nicolas Silberman, a décidé de pousser les murs : ouverture de nouvelles salles, ajout de conférences.... Un très beau succès pour cette édition !

Après l'introduction par les organisateurs, c'est Rasmus Lerdorf, créateur de la toute première version de PHP, qui a ouvert le cycle de conférences en présentant l'histoire de PHP et son évolution possible. Le message que l'on retiendra de son intervention est que PHP est désormais un langage mature, celui-ci étant à l'origine d'un écosystème lui aussi parvenu à maturité, comme en témoignent les nombreux frameworks, CMS, librairies, extensions etc. qui composent cette plateforme, qui ne s'arrête donc pas au seul langage PHP, et que l'ensemble des conférences aura mis en valeur.

Conférences de qualité et diversifiées

Dans le cadre de la thématique de l'industrialisation de PHP, chère à l'AFUP, Canal+ a présenté son industrialisation de PHP. Avec plus de 700 000 visites par jour et un grand nombre d'applications, les mots d'ordre sont disponibilité, performances et industrialisation.

Lors de cette conférence on entend parler de Capistrano, d'environnements de production virtualisés, de

Magento, de Zend Framework, d'eZ Publish et de Symfony !

Adobe, sponsor Platinum, a dévoilé pendant sa conférence les prochaines possibilités offertes par le framework Flex (actuellement en bêta). Ces nouvelles possibilités vont permettre de réaliser de nouvelles applications portables, touchant les secteurs de la simulation et du déplacement d'objets par l'intermédiaire de votre téléphone mobile. Les échanges de données pourront naturellement être traités par PHP.

Une autre thématique forte de ce forum concernait les performances de PHP. La première session a été menée par Guillaume Plessis (responsable de dotdeb) et Cyril Pierre de Geyer (co-auteur de PHP 5 avancé) sur leur thématique favorite : « Optimisation des performances de sites Web ». Pour optimiser une application il faut savoir ce qu'il se passe (via les logs et les outils de monitoring), simuler (siege, Jmeter), analyser (Xdebug, XhProf) et améliorer !

Après cette introduction aux performances PHP, c'est la société Weka qui est venue présenter son utilisation de PHP pour créer des jeux Facebook. Avec parfois près de 50 000 utilisateurs simultanés les performances et la qualité de services sont indispensables. Pour y parvenir, Laurent Letourmy, directeur technique de Weka, a mis en place avec ses équipes différents outils : GIT pour la gestion des versions, QoS pour les métriques qualité, Pinba pour le monitoring, Redis pour sa puissance fonctionnelle et ses performances, devOP pour la méthodologie de travail.

Attention, la spécificité de Weka est d'utiliser massivement le Cloud pour son architecture. Le conseil à retenir de leur part est donc : pensez votre code en mode « scale » pour profiter du Cloud le jour où votre application explosera.

Enfin, peu avant la clôture, Zeev Suraski, co-fondateur de Zend Technologies (l'autre sponsor Platinum de cette édition du forum), s'est essayé avec succès à démythifier le paradoxe entre l'enrichissement fonctionnel permanent de la plateforme PHP et les performances qui sont cependant sans cesse améliorées. Pour mémoire, Zeev est co-auteur des versions modernes (3, 4 et 5 notamment) de PHP. Autant dire qu'il maîtrise son sujet ! L'intervention de SkySQL était très attendue. Occasion d'en savoir plus sur cette société de services pour MySQL. On y apprend que SkySQL s'est associé avec Monty Program pour fournir des offres de support jusqu'au niveau 3 pour MySQL.

Les retours

Le forum PHP 2010 ce n'est pas que des conférences, c'est également des exposants et des communautés qui présentent leurs projets. Cette année, l'AFUP avait invité plus d'une dizaine de ces communautés : Hoa, Drupal, l'ERP Dolibarr, RBS, CakePHP-fr, Jelix, Symfony, le CMS Magix, Crisis camp, etc. A chaque pause c'est un maelström d'échanges et de conseils qui fusaient.

Toutes les présentations sur <http://www.afup.org/forumphp/>.

■ Cyril Pierre de Geyer

PDC : Azure, asynchrone et Silverlight

Nouveau lieu, nouveau format, la conférence développeur de Microsoft ou PDC, mise sur la diffusion en direct pour faire profiter les milliers de développeurs qui ne pouvaient se déplacer ou trouver une place à la conférence. Windows Phone 7, IE9 et Azure ont tenu la vedette.



Windows Azure : virtualisation en prime

Commençons tout d'abord par l'arrivée d'un nouveau rôle : VM Role. Sous ce terme un peu obscur se cache la possibilité de déployer sur Azure, une machine virtuelle Windows Server 2008 R2 !

C'est une évolution très importante car jusqu'à présent, ce type de déploiement était surtout dévolu aux offres IaaS (partie infrastructure) et non à la plate-forme (PaaS). Le VM Role doit permettre de simplifier la migration vers le cloud des entreprises. Il s'agit aussi de pouvoir, en cas de pic de charge ou pour faire des tests, de rapidement mettre en place une plate-forme.

Cette annonce s'accompagne aussi de plusieurs autres fonctions : l'apparition de Virtual Network et de sa première fonction, Connect, qui est en réalité le projet Sydney. Le but est de créer très rapidement un tunnel de communication, un réseau pour déployer, migrer des applications, données, machines virtuelles (le tout en IPSec). Il s'agit de reprendre la fonction DirectAccess mais sur le cloud ! La machine virtuelle s'exécute telle quelle avec la même configuration, le même mapping.

Autre nouveauté importante, la possibilité de déployer une image applicative virtuelle directement sur Windows Azure. Il s'agit en fait d'avoir sur

Azure, l'approche bulle applicative de Microsoft App-V. Précision importante : une machine virtuelle sous Azure pourra être maintenant multi-site. A noter que l'on pourrait aussi créer plusieurs comptes administrateurs, chose impossible jusqu'à présent. Et côté administrateur, on pourra faire du Remote Desktop sur une instance pour voir ce qui s'y passe, monitorer les applications déployées dessus. Dans le même esprit, le support de IntelliTrace (pour le debug) va subir une sérieuse mise à niveau pour aider le développeur à mieux déboguer une application Azure.

Autre annonce : l'ouverture de Azure Marketplace pour les applications et les données (projet Dallas). Il s'agira d'y trouver des offres de services, des applications Azure que les développeurs, entreprises proposeront. Sur la partie données, SQL Azure se complètera de deux nouvelles fonctions : reporting et synchronisation des données.

D'autre part, le portail Azure va être entièrement refondu pour être plus productif, plus pratique et perdre son apparence développeur.

Il utilise massivement la technologie Silverlight. Sur ce point, nous attendons des précisions sur la compatibilité avec les navigateurs sous MacOS X et Linux.

Toutes ces nouveautés et mises à jour (nous n'avons listé que les princi-

pales) seront disponibles en préversion (CTP) d'ici fin 2010 ou courant 2011, avec une disponibilité finale courant 2011.

Les autres nouveautés

Microsoft a aussi levé le voile sur une nouvelle édition de Visual Studio : Visual Studio Async CTP. Cette version repose sur les briques TPL. Le but est d'offrir une programmation asynchrone et vise à réduire les temps de latence et les problèmes d'attente aussi niveau interface ou entre les différentes couches d'une application.

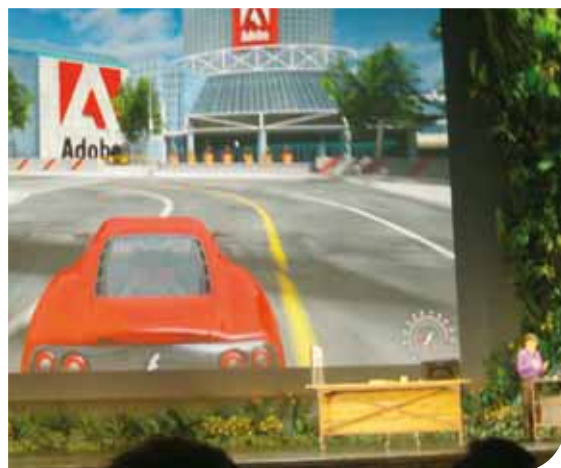
C'est une évolution très importante dans le modèle de programmation. Cela sera donc possible en VB et C#. L'approche asynchrone fait suite à l'implémentation du déclaratif, de la concurrence et du dynamique. L'asynchrone répond aux problèmes de latence, de réactivité de l'interface et de la montée en charge (notamment en contexte multicore).

Microsoft a aussi répondu aux interrogations autour de Silverlight, à cause des annonces de l'éditeur envers HTML5.

Pour Microsoft, Silverlight ne remplacera pas HTML et l'inverse est vrai aussi. Et surtout, Silverlight est au cœur de la stratégie Windows Phone 7. Mais quelle sera la place exacte de HTML 5 et de Silverlight par rapport à HTML 5 ?

Adobe Max 2010 : Adobe présente à tout va !

C'est à Los Angeles, que s'est déroulée l'Adobe MAX, en octobre dernier, la grand-messe des designers et développeurs. Au programme : un nombre vertigineux de conférences et d'ateliers autour des technologies phare d'Adobe. Plusieurs milliers de personnes se sont pressées pour faire le plein d'inspiration, de connaissances, d'infos croustillantes et de surprises.



C'est Kevin Lynch, le directeur technique d'Adobe, qui lance véritablement la conférence.

Il explique à quel point les smart-phones et les tablettes suivent une courbe similaire à la montée en puissance des PC personnels, et que ces derniers seront dépassés en 2013. Le réel défi à venir, c'est de pouvoir proposer une expérience similaire partout : c'est la vision multi-écran.

Le Web

Afin de nous démontrer la pertinence de son propos, Kevin Lynch rétrécit la fenêtre d'un site internet lambda à plusieurs reprises ... rien ne bouge, tout s'occulte. Il nous propose alors de voir le même site réalisé avec la dernière version de Dreamweaver. Grâce à un système multi CSS, le contenu s'adapte automatiquement aux différentes résolutions des périphériques. Adobe ne veut pas mettre tous ses œufs dans le même panier et tient à nous le faire savoir. Avec Edge, un nouveau logiciel encore en version prototype, il est possible de réaliser des animations poussées en HTML5 sans faire une ligne de code. Autrement dit, du Flash sans Flash en wysiwyg. A mi-chemin entre Powerpoint et Photoshop, la prise en main a l'air d'une efficacité redoutable. Le chapitre consacré au WEB se referme sur SiteCatalyst, un outil en ligne qui traque les moindres faits et gestes de vos visiteurs en temps réel. Arborant une interface claire et irréprochable, il est facile de déterminer les points d'abandon, comment différents segments de visiteurs interagissent avec votre site ou tout

simplement le cheminement vers les pages les plus consultées.

La vidéo

La vidéo est tellement importante sur le web, qu'Adobe se devait d'avoir un positionnement on ne peut plus clair sur ce secteur. En misant sur le monde des média télévisuels, Kevin Lynch veut imposer l'utilisation de AIR pour apporter une dimension interactive. D'ailleurs Google et Adobe ont travaillé main dans la main sur l'interface de la Box Google TV, sur laquelle il sera possible de faire tourner des applications AIR. Des constructeurs, tels que Samsung, prévoient d'incorporer cette technologie dans leur futur téléviseur. On nous présente alors la nouvelle version d'Adobe media encoder, qui est désormais livrée avec toute une batterie de preset pour convertir une vidéo avec une multitude de dispositifs en une seule opération.

Flash

Adobe dévoile la 3D dans Flash ! L'API 3D permettra aux développeurs d'intégrer de la vraie 3D au travers de Flash, elle porte l'étrange nom de "Molehill". Et la consommation énergétique semble très bonne : entre 0 et 1%, ce prodige s'explique par le fait que Flash utilise la puissance du GPU.

Le Sneak Peek : très geek !

Le Sneak Peek est le 2e moment fort de toute Adobe Max. Cette année n'a pas déçu. Video Tapestries nous a laissé bouche bée. Les concepteurs sont partis du postulat qu'il était très difficile de se retrouver dans un film, car les vignettes d'aperçus n'étaient

pas toujours visuellement explicites. Ils ont donc développé un système, avec entre autres, une reconnaissance de visage et un algorithme qui détoure les éléments animés, en permettant de découper des scènes clés et de les agencer sur une time line qui se lirait comme une tapisserie médiévale. Il est même possible de manipuler et de zoomer dans cette fresque qui se recompose alors en temps réel.

Photoshop Whirlwind a valu à son démonstrateur d'être traité de "sorcier" par les spectateurs. Imaginez, d'un côté une simple photo que vous venez de prendre avec votre mobile préféré et de l'autre une photo d'un photographe célèbre avec un grain, un flou, une atmosphère colorimétrique bien particulière... glissez/déposez cette dernière sur votre cliché, après 3 ou 4 secondes de calcul, votre photo a littéralement aspiré le style du photographe professionnel ! Côté développeur, des choses tout aussi étonnantes, comme de pouvoir faire fonctionner les applications en live dans la vue design de Flash builder et d'apporter des modifications en temps réel. Ajoutez des boutons, des listes etc., ils sont utilisables dans l'instant. A noter aussi la possibilité d'exporter/convertir les animations de Flash Professionnel en HTML 5. Cette possibilité est plutôt bienvenue en termes de productivité, si l'on considère que cela offrira, sans labeur supplémentaire, une version alternative de vos animations sur des plateformes « fruitées » dépourvue de Flash.

■ Greg Madison - Wygwam

Exclusif :

Mandriva donnera t-il naissance à l'OS de la Russie en 2011 ?



*Non, le logo officiel de Mandriva n'a pas (encore) changé ! Cette illustration « libre » n'est que notre interprétation...

Les « logiciels libres » ne s'arrêtent plus aux frontières du « monde libre »...

Mandriva était soutenu depuis 2006 par OCAM, un Fonds Commun de Placement à Risques, qui a investi au total 3 millions d'euros dans le groupe.

Mais l'exploitation restait déficitaire et l'entreprise frôlait la liquidation. Mandriva a trouvé son sauveur. Il est russe.

Artur Akopyan dirige le fonds NGI (New Generation Initiative) et apporte 2 millions d'euros à la société.

Mais Akopyan, ancien directeur financier de l'opérateur russe Synterra Telecommunications serait en fait soutenu, selon certains médias russes, par Leonid Reiman, ancien Ministre des

communications et désormais conseiller du Président Medvedev. Il est surtout celui qui annonçait, alors ministre, en octobre 2007 : « *Russia OS sera installé sur tous les ordinateurs des écoles en Russie pour 2009.*

Bien plus chaque élève pourra ainsi utiliser des applications produites en Russie. » Les objectifs : ne pas dépendre de l'informatique américaine, réduire le coût des licences et du piratage.

Russia OS, distribution GNU/Linux à créer, devait être testée dans 3 régions russes avant d'être installée sur l'ensemble des machines scolaires, avec un budget de 2,7 milliards de roubles (88 millions de dollars).

Medvedev, alors premier Ministre, s'était également enga-

gé sur le développement de logiciels libres pour l'administration.

Trois compétiteurs sont en présence

D'abord, ALT Linux, distribution GNU/Linux russe déjà connue, participe depuis l'origine au développement de Russian OS. Elle a travaillé avec les 3 régions test, représentant un millier d'écoles. Or Alt Linux n'est rien moins qu'un fork russe de Mandrake créé en 2001 ! Une société d'état, Russian Technology, a pris des parts en juillet de cette année dans Alt Linux, afin de la promouvoir comme l'opérateur national. De son côté, NGI, après son investis-

sement dans Mandriva, a acquis une société de services open source russe, PingWin Software. Cette société effectue le support dans les écoles russes. Un troisième compétiteur existe, avec une version issue de Fedora, la distribution libre de RedHat, développée pour...l'ex Armée Rouge ! Mandriva en tout cas a des chances sérieuses de participer ou en direct, ou au travers de son fork, au russe OS et de développer un marché de logiciels et de services importants. L'opérateur de l'OS national sera retenu après un appel d'offre début 2011.

■ Jean Kaminsky

Sources : <http://en.rnan.ru/papers/20100907/160501001.html> (en anglais)
<http://www.cnews.ru/news/top/index.shtml?2010/11/03/414798> (en russe)

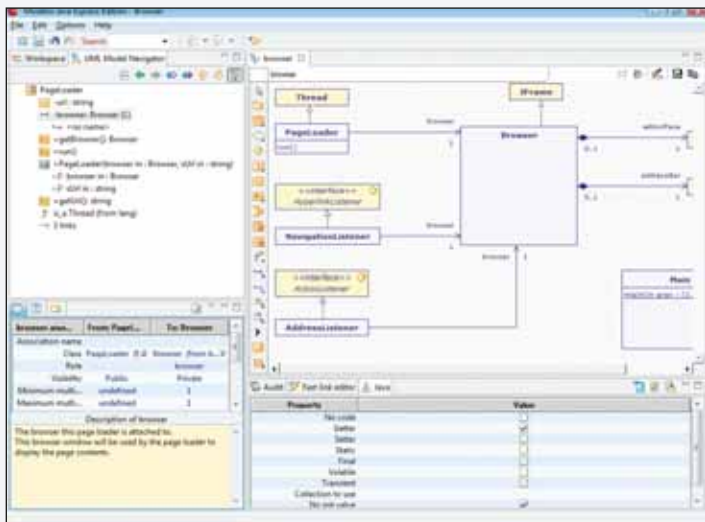


Modelio : une offre de modélisation unique

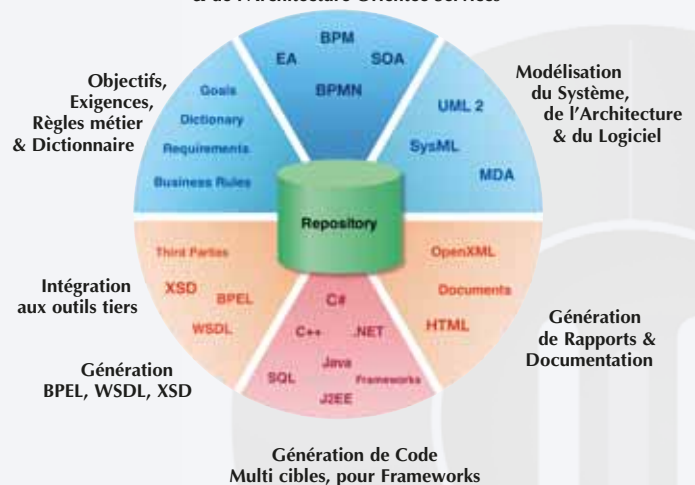
Libérez la vraie puissance de vos modèles !

UML, BPMN, Exigences ..., MDA, génération de code ...

- Modélisation intégrée de UML2, BPMN, SysML, l'Architecture d'Entreprise, les exigences, le dictionnaire, ... dans un seul référentiel
- Génération Java, C#, C++, SQL, XML, XSD, BPEL, WSDL, Hibernate...
- MDA simple et puissant - transformation, extensibilité et adaptabilité
- Travail de groupe distribué, intégré à SVN/Subversion
- Ergonomie simple, productive et familière aux développeurs (RCP/Eclipse)



Modélisation de l'Architecture d'Entreprise, des Processus Métier & de l'Architecture Orientée Services



Modelio est disponible en trois éditions

- **Free** : Un outil de modélisation UML2, BPMN, et de développement MDA, complet et gratuit !
- **Express Java** : Un outil de développement UML2/Java performant pour seulement 100 € !
- **Enterprise** : La solution de modélisation complète, supportant le travail de groupe, extensible avec une riche palette de modules de modélisation et de génération disponibles sur étagère

Téléchargez la nouvelle version de Modelio !
www.modeliosoft.com

sales@modeliosoft.com
 Tél. : 01 30 12 18 40



Optimisez vos applications Flash pour les mobiles

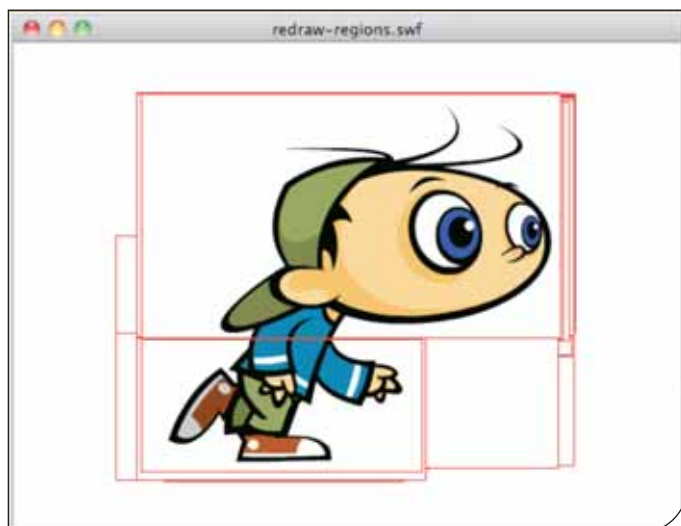
Adobe a annoncé la disponibilité du Flash Player 10.1 sur les mobiles Android en Juin dernier. Les chanceux munis d'un smartphone Android ont juste à se rendre sur l'Android Marketplace pour installer Flash et accéder à tout le web par le navigateur.

Dernièrement, le runtime Adobe AIR 2.5 est aussi disponible sur Android. Il permet de transformer des applications Flash en applications natives Android. Cette technique de packaging est aussi possible sur l'iPhone ou l'iPad, grâce à l'iPhone packager disponible sur labs.adobe.com, et ré-autorisé dernièrement par Apple. Enfin, BlackBerry, HP (Palm) et Microsoft ont aussi annoncé l'arrivée de Flash sur leurs mobiles et tablettes. C'est une grande nouvelle pour les 3 millions de développeurs de la Flash Platform, qu'ils soient développeurs Flash ActionScript 3 ou Flex. Mais je dois les mettre en garde. Les développeurs de la Flash Platform sont habitués à exécuter leurs réalisations sur Windows, MAC et Linux, sur des environnements avec des processeurs rapides et une quantité de mémoire importante. Pour que leurs applications Flash s'exécutent correctement sur mobiles, il faut tenir compte de quelques règles de base d'optimisation. Cet article présente comment optimiser son code ActionScript 3 pour environnements mobiles, mais c'est aussi valable pour les desktop bien sûr. Vous pouvez retrouver tous les exemples utilisés dans le guide en ligne « Optimizing performance for the Flash Platform ».

Pour comprendre comment améliorer les performances d'une application, il est essentiel de comprendre comment le moteur d'exécution de la plate-forme Flash exécute le code. Le moteur d'exécution fonctionne en boucle, certaines actions se produisant sur chaque « image ». On entend ici par image un simple bloc de temps déterminé par la cadence définie pour l'application. Le temps alloué à chaque image correspond directement à la cadence. Si vous spécifiez une cadence de 30 images par seconde, par exemple, le moteur d'exécution s'efforce de faire durer chaque image un trentième de seconde. Chaque boucle d'image comprend deux phases, divisées en trois parties : les événements, l'événement enterFrame et le rendu. Si les opérations combinées de la boucle d'image (exécution du code et rendu) durent trop longtemps, le moteur d'exécution ne peut pas assurer une cadence uniforme. L'image s'étend au-delà du temps qui lui est alloué, retardant ainsi le déclenchement de l'image suivante. Et c'est souvent le rendu, plus que le calcul pur, qui dans 90% des cas ralentit une application Flash. Conséquence, on perçoit des animations saccadées, des vidéos qui ne sont pas fluides, une fenêtre qui scintille, un décalage entre le moment où l'on saisit le texte et quand il s'affiche, etc. Dès que l'on rencontre ces écueils, il faut cibler la bonne optimisation. Une première piste, optimisez la mémoire de l'application Flash.

Optimisation de la mémoire

ActionScript 3.0 propose un grand nombre d'objets d'affichage. Une des plus simples techniques d'optimisation visant à limiter la consommation de mémoire consiste à choisir le type approprié d'objet d'affichage. Pour créer des formes simples qui ne sont pas interactives,



utilisez les objets **Shape**. Pour créer des objets interactifs ne nécessitant pas de scénario, faites appel aux objets **Sprite**. Pour une animation s'appuyant sur un scénario, recourez aux objets **MovieClip**. Choisissez toujours le type d'objet le plus performant pour l'application. Utilisez la méthode `getSize()` pour connaître l'occupation mémoire d'un objet. Cela peut aller du simple au double pour rien.

```
trace(getSize(new Shape()));
// output: 236
trace(getSize(new Sprite()));
// output: 412
trace(getSize(new MovieClip()));
// output: 440
```

Réutilisez les objets

Une autre technique simple d'optimisation de la mémoire consiste à réutiliser les objets pour éviter, dans la mesure du possible, de les recréer. Une erreur typique mais très gourmande en ressource consiste à recréer des objets dans une boucle, comme dans l'exemple qui suit :

```
const MAX_NUM:int = 18;
const COLOR:uint = 0xCCCCCC;
var area:Rectangle;
for (var:int = 0; i < MAX_NUM; i++)
{
    area = new Rectangle(i,0,1,10);
    myBitmapData.fillRect(area,COLOR);
}
```


La création d'objet comme ici l'objet Rectangle à chaque itération e la boucle ralentit la machine virtuelle. Voici comment il faut procéder :

```
const MAX_NUM:int = 18;
const COLOR:uint = 0xCCCCCC;

var area:Rectangle = new Rectangle(0,0,1,10);
for (var:int = 0; i < MAX_NUM; i++)
{
    area.x = i;
    myBitmapData.fillRect(area,COLOR);
}
```

L'objet Rectangle reste un objet léger, mais appliquer un objet Bitmap (pour un jeu par exemple), l'effet sur la mémoire peut être colossal.

La technique du pooling

La technique du pooling consiste à stocker temporairement des objets non visibles pour les réutiliser ultérieurement. Vous créez un nombre défini d'objets lors de l'initialisation de l'application et les enregistrez dans un pool, tel qu'un objet Array ou Vector. Lorsque vous en avez terminé avec un objet, vous le désactivez pour éviter qu'il ne consomme des ressources de l'unité centrale et vous supprimez toutes les références mutuelles. Vous ne définissez toutefois pas les références sur null, ce qui rendrait l'objet éligible pour le processus de nettoyage de la mémoire. Vous vous contentez de lui faire réintégrer le pool et de l'en extraire lorsque vous avez besoin d'un nouvel objet. Voici une mise en place simple d'un pool d'objets :

```
package
{
    import flash.display.Sprite;
    public final class SpritePool
    {
        private static var MAX_VALUE:uint;
        private static var GROWTH_VALUE:uint;
        private static var counter:uint;
        private static var pool:Vector.<Sprite>;
        private static var currentSprite:Sprite;
        public static function initialize( maxPoolSize:uint, growthValue:uint ):void
        {
            MAX_VALUE = maxPoolSize;
            GROWTH_VALUE = growthValue;
            counter = maxPoolSize;
            var i:uint = maxPoolSize;
            pool = new Vector.<Sprite>(MAX_VALUE);
            while( --i > -1 )
            pool[i] = new Sprite();
        }

        public static function getSprite():Sprite
        {
            if ( counter > 0 )
            return currentSprite = pool[--counter];
            var i:uint = GROWTH_VALUE;
            while( --i > -1 )
```

```
pool.unshift ( new Sprite() );
counter = GROWTH_VALUE;
return getSprite();
}
public static function disposeSprite(disposedSprite:Sprite):void
{
    pool[counter++] = disposedSprite;
}
}
```

La classe SpritePool crée un pool de nouveaux objets lors de l'initialisation de l'application. La méthode getSprite() renvoie des occurrences de ces objets tandis que la méthode disposeSprite() les libère. Le code autorise l'expansion du pool une fois celui-ci entièrement consommé. Il est également possible de créer un pool de taille fixe, qui une fois épuisé, interdit l'allocation de nouveaux objets.

Eviter si possible l'utilisation de filtres

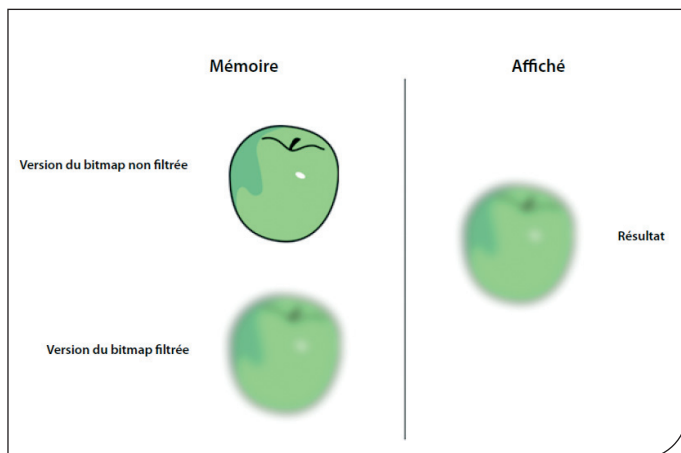
Utilisez un minimum d'effets tels que les filtres, y compris ceux traités sur les périphériques mobiles par le biais de Pixel Bender. Lors de l'application d'un filtre à un objet d'affichage, Flash Player crée deux bitmaps en mémoire, chacun de la taille de l'objet d'affichage. Le premier est une version pixellisée de l'objet d'affichage et sert à créer le second, auquel le filtre est appliqué. Lors de la modification de l'une des propriétés d'un filtre, les deux bitmaps sont mis à jour en mémoire pour créer le bitmap résultant. Ce processus sollicite l'unité centrale et les deux bitmaps peuvent nécessiter beaucoup de mémoire. Il est recommandé de créer des bitmaps dans un outil de création, tel qu'Adobe Photoshop, pour émuler les filtres, si possible. Evitez l'utilisation de bitmaps dynamiques créés à l'exécution dans ActionScript. L'utilisation de bitmaps créés en externe permet à Flash Player de réduire la charge de l'unité centrale ou du processeur graphique, surtout si les propriétés du filtre ne changent pas à terme. Si possible, créez les effets requis sur un bitmap dans un outil de création. Vous pouvez ensuite afficher ce bitmap dans Flash Player sans le traiter de quelque façon que ce soit, ce qui est beaucoup plus rapide.

Libérer proprement les objets

Pour optimiser le code, figez et libérez systématiquement les objets. Il est important d'effectuer ces opérations pour tous les objets, et plus particulièrement les objets d'affichage. Même s'ils ne figurent plus sur la liste d'affichage et sont en attente de nettoyage, les objets d'affichage sont toujours susceptibles de solliciter fortement l'unité centrale. Il se peut, par exemple, qu'ils continuent d'utiliser Event.ENTER_FRAME. Il est donc impératif de figer et de libérer correctement les objets à l'aide des événements Event.REMOVED_FROM_STAGE et Event.ADDED_TO_STAGE. L'exemple suivant illustre un clip qui s'exécute sur la scène et communique avec le clavier :

```
stage.addEventListener(KeyboardEvent.KEY_DOWN, keyIsDown);
stage.addEventListener(KeyboardEvent.KEY_UP, keyIsUp);

var keys:Dictionary = new Dictionary(true);
function keyIsDown(e:KeyboardEvent):void
{
```



```

keys[e.keyCode] = true;
if (e.keyCode==Keyboard.LEFT || e.keyCode==Keyboard.RIGHT)
{
    runningBoy.play();
}
}

function keyIsUp(e:KeyboardEvent):void
{
    keys[e.keyCode] = false;
    for each (var value:Boolean in keys)
    if ( value ) return;
    runningBoy.stop();
}

runningBoy.addEventListener(Event.ENTER_FRAME, handleMovement);
runningBoy.stop();
var currentState:Number = runningBoy.scaleX;
var speed:Number = 15;

function handleMovement(e:Event):void
{
    if (keys[Keyboard.RIGHT])
    {
        e.currentTarget.x += speed;
        e.currentTarget.scaleX = currentState;
    } else if (keys[Keyboard.LEFT])
    {
        e.currentTarget.x -= speed;
        e.currentTarget.scaleX = -currentState;
    }
}

```

Imaginons un utilisateur qui cliquerait sur un bouton « Supprimer » pour détruire ce clip de la scène.

```

showBtn.addEventListener (MouseEvent.CLICK,showIt);
removeBtn.addEventListener (MouseEvent.CLICK,removeIt);
function showIt (e:MouseEvent):void
{
    addChild (runningBoy);
}
function removeIt(e:MouseEvent):void
{

```

```

if (contains(runningBoy)) removeChild(runningBoy);
}

```

Bien que supprimé de la liste d'affichage, le clip distribue toujours l'événement `Event.ENTER_FRAME`. Son exécution se poursuit même si son rendu est interrompu. C'est une erreur classique que j'ai souvent rencontré, et qui peut anormalement utiliser du CPU alors que rien ne bouge à l'écran !!! Un cas typique est l'usage d'un « ticker » pour un preloader. Une fois le contenu chargé, on supprime le ticker qui s'il est mal codé, consomme du CPU en tâche de fond pour rien. Pour gérer correctement cette situation, écoutez les événements appropriés et supprimez des écouteurs d'événements, afin d'éviter l'exécution de code sollicitant fortement l'unité centrale :

```

Event.REMOVED_FROM_STAGE    runningBoy.addEventListener(Event.
    ADDED_TO_STAGE,activate);
runningBoy.addEventListener(Event.REMOVED_FROM_STAGE,deactivate);

function activate(e:Event):void
{
    e.currentTarget.addEventListener(Event.ENTER_FRAME,handleMovement);
}

function deactivate(e:Event):void
{
    e.currentTarget.removeEventListener(Event.ENTER_FRAME,handle
    Movement);
    e.currentTarget.stop();
}

```

Lorsque l'utilisateur appuie sur le bouton Show (Afficher), le clip redémarre, écoute à nouveau les événements `Event.ENTER_FRAME` et le clavier le contrôle correctement.

Timer ou ENTER_FRAME ?

Lorsque le contenu n'est pas animé et s'exécute pendant une longue période, les minuteurs sont préférables aux événements `Event.ENTER_FRAME`.

Dans ActionScript 3.0, il est possible d'appeler une fonction à intervalles réguliers de deux manières. La première repose sur l'utilisation de l'événement `Event.ENTER_FRAME` distribué par les objets interactifs (`InteractiveObject`). La seconde est fondée sur l'utilisation d'un minuteur. Les développeurs ActionScript privilégient fréquemment l'utilisation de l'événement `ENTER_FRAME`. L'événement `ENTER_FRAME` est distribué sur chaque image. La fréquence d'appel de la fonction est donc relative à la cadence en cours. La cadence est accessible par le biais de la propriété `Stage.frameRate`. Il est cependant préférable, dans certains cas, de faire appel à un minuteur plutôt qu'à l'événement `ENTER_FRAME`. C'est le cas, par exemple, si vous ne définissez pas d'animation mais souhaitez appeler le code à fréquence régulière.

Un minuteur peut se comporter de manière similaire à un événement `ENTER_FRAME`, mais il permet de distribuer un événement sans être tributaire de la cadence. Ce comportement peut permettre une optimisation sensible. Prenons l'exemple d'un lecteur vidéo. Dans ce cas, une cadence élevée n'est pas indispensable car seuls les contrôles de l'application se déplacent.


```
stage.frameRate = 10;

// Une mise à jour par seconde
var updateInterval:int = 1000;
var myTimer:Timer = new Timer(updateInterval,0);
myTimer.start();
myTimer.addEventListener( TimerEvent.TIMER, updateControls );

function updateControls( e:TimerEvent ):void
{
    e.updateAfterEvent();
}
```

Sur chaque image, le moteur d'exécution distribue un événement `enterFrame` à chaque objet figurant sur sa liste d'affichage. Vous pouvez enregistrer des écouteurs pour l'événement `enterFrame` sur plusieurs objets d'affichage mais cela signifie que le volume de code exécuté sur chaque image est plus élevé. Envisagez plutôt de définir un gestionnaire `enterFrame` centralisé unique qui exécute tout le code destiné à chaque image.

En centralisant ce code, il est plus facile de gérer tout le code qui s'exécute fréquemment. De même, si vous utilisez plusieurs objets `Timer`, la création et la distribution des événements correspondants se traduisent par une augmentation du temps système nécessaire. Utilisez un minimum d'objets `Timer` et regroupez les opérations en fonction de leur fréquence d'exécution. Utilisez un seul objet `Timer` et déclenchez les opérations à des fréquences consistant en des multiples de la propriété `delay` de l'objet. Supposons, par exemple, que vous souhaitiez exécuter certaines opérations toutes les 100 millisecondes et d'autres toutes les 200 millisecondes. A cet effet, créez un seul objet `Timer` et définissez sa propriété `delay` sur 100 millisecondes. Dans le gestionnaire d'événement `timer`, ajoutez une instruction conditionnelle qui exécute une fois sur deux seulement les opérations à répéter toutes les 200 millisecondes. L'exemple suivant illustre cette technique :

```
var timer:Timer = new Timer(100);
timer.addEventListener(TimerEvent.Timer, timerHandler);
timer.start();
var offCycle:Boolean = true;

function timerHandler(event:TimerEvent):void
{
    // Lancer des actions toutes les 100 ms
    if (!offCycle)
    {
        // Lancer des actions toutes les 200 ms
    }
}
```

```
}
offCycle = !offCycle;
}
```

Utilisation du GPU

Une nouvelle fonction importante de Flash Player 10.1 lui permet d'utiliser le processeur graphique pour effectuer le rendu du contenu graphique sur les périphériques mobiles. Auparavant, l'unité centrale effectuait seule le rendu des graphiques. L'utilisation du processeur graphique optimise le rendu des filtres, des bitmaps, de la vidéo et du texte.

Gardez à l'esprit que le rendu par le processeur graphique n'est pas toujours aussi précis que le rendu logiciel. Lorsque vous utilisez le rendu matériel, la définition du contenu à l'écran est moins nette.

Flash Player 10 possédait une fonction d'accélération du processeur graphique. Il n'utilisait néanmoins pas celui-ci pour calculer les graphiques mais simplement pour les envoyer à l'écran. Flash Player 10.1 utilise le processeur graphique pour le calcul des graphiques, d'où une amélioration sensible de la vitesse de rendu. La charge de l'unité centrale est également réduite, ce qui est utile sur les périphériques aux ressources limitées, tels que les périphériques mobiles.

Lorsque vous utilisez l'outil `Packager for iPhone`, le rendu du processeur graphique fonctionne différemment. Le processeur graphique est réservé aux bitmaps, aux formes pleines et aux objets d'affichage dont la propriété `cacheAsBitmap` est définie. Par ailleurs, il peut effectuer le rendu des objets qui pivotent ou sont mis à l'échelle, et dont les propriétés `cacheAsBitmap` et `cacheAsBitmapMatrix` sont activées.

A vous de jouer

Toutes ces astuces sont documentées dans le guide « Optimizing Performance for the Flash Platform », disponible gratuitement sur Internet. Vos contenus Flash pourront ainsi s'exécuter correctement, et sans ralentissement, sur mobiles et Tablettes Android, iOS et bientôt WebOS et BlackBerry. Ces techniques d'optimisation ne sont pas contraignantes à mettre en place si elles le sont dès le début du développement. Donc désormais, à vous de jouer et de nous surprendre sur mobiles. Vous pourrez aussi retrouver de nombreux exemples optimisés et des mini jeux open source sur mon blog RIAgora.com.



■ Michaël Chaize



Thibault Imbert ■

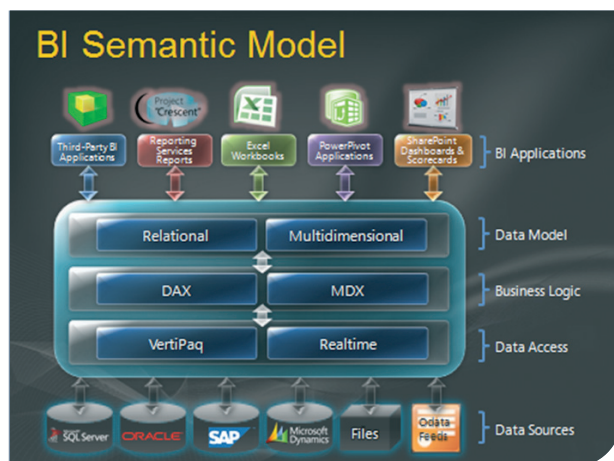
L'information permanente

- L'actu de Programmez.com : le fil d'info quotidien
- La newsletter hebdo : la synthèse des informations indispensables.
Abonnez-vous, c'est gratuit !

www.programmez.com

SQL Server Denali : automne 2011 !

Début novembre, Microsoft dévoilait la première pré-version technique du futur SQL Server : SQL Server 2011, alias Denali dont on avait eu quelques échos depuis 6 mois sans réellement en savoir plus. Et Denali ne vient pas seul !



Cette version de SQL Server devrait sortir à l'automne 2011. Plusieurs grosses nouveautés sont prévues : renforcement de la gestion du failover et du multi-clustering, support de PowerShell 2 (attention : il doit être installé car non livré avec Denali), .Net 4, nouveautés dans la haute disponibilité, la montée en charge, la BI, les analyses. Trois piliers peuvent être définis : SQL Server AlwaysOn (solution pour la haute disponibilité), le projet Crescent (reporting et visualisation des données basé sur la technologie PowerPivot) et enfin une nouvelle partie pour l'intégration et la

gestion des données. Côté administration et développement, il faut noter trois éléments :

- une amélioration des fonctions d'administration : audit renforcé pour la sécurité ainsi que sur les tests de données
- de meilleures performances sur le FileStream, la recherche ou encore le spatial 2D
- apparition du projet « Juneau ». Outil dédié aux développeurs pour faciliter les développements des applications données.

Un des gros morceaux de Denali est l'intégration services. Tout d'abord, Denali apporte un nouveau modèle de

déploiement pour l'intégration des données dans la pile de l'entreprise. Un projet peut être déployé directement sur le moteur SQL Server mais surtout, on bénéficie de nouveaux outils, de nouvelles vues pour T-SQL, les données, les packages à monitorer. On aura aussi un designer spécifique pour les dépendances. Les équipes de SQL Server ont également beaucoup travaillé sur la fonction « merge » afin d'en réduire l'occupation mémoire. Notons par ailleurs la présence d'un SQL Server Management Studio intégrant un éditeur de T-SQL amélioré avec un debug pouvant déboguer un script s'exécutant sur une instance SQL Server 2005 SP2 (ou supérieur), le point d'arrêt bénéficie de nouveaux supports, idem sur l'IntelliSense plus complet.

Sur le moteur en lui-même, on pourra configurer l'index plein texte et la recherche de proximité pourra elle aussi être personnalisée. Sur les données spatiales, on bénéficie de trois nouveaux types de données : circularString, CompoundCurve et CurvePolygon. Quant à la sécurité proprement dite, de nouvelles permissions seront accessibles ainsi qu'un nouveau rôle de gestion et on pourra faire du hash sur SHA2_256 et 512. Cette version continue la lignée de la 2008 et 2008 r2. On peut retenir que Denali pose les bases de la BI sémantique autour du modèle de données, de la logique business et de l'accès aux données.

Nous aurons plus de détails durant les prochains mois et surtout, il faudra attendre avant que l'ensemble des fonctions soit figé.

Un SQL Server taillé pour l'énormité !

SQL Server 2008 R2 Parallel Data Warehouse est conçu pour répondre de façon simple aux besoins des entreprises dans le domaine des entrepôts de données stratégiques. Cette architecture de traitement massivement parallèle est déjà

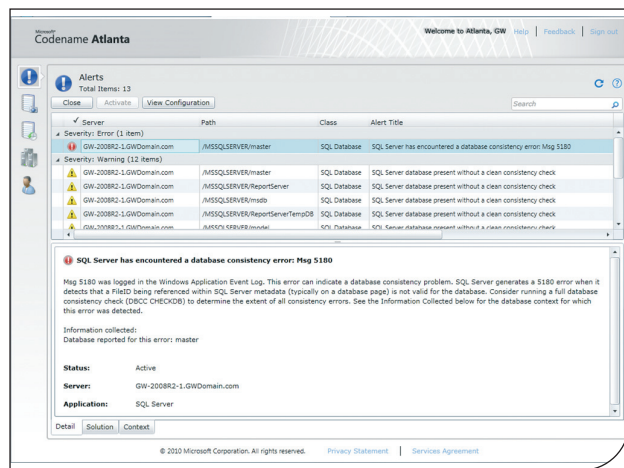
configurée et optimisée pour permettre un traitement distribué des requêtes avec les plus hauts niveaux de performance.

Microsoft rend cette offre d'appliance Parallel Data Warehouse disponible pour les clients tout en leur laissant

le choix du matériel. Dans la foulée de l'annonce, HP a indiqué la disponibilité d'une solution basée sur cette édition en décembre 2010.

Du nouveau aussi sur le cloud

Dès aujourd'hui, vous pouvez vous inscrire au projet Microsoft nommé « Atlanta », un nouveau service dans le cloud pour surveiller la configuration de SQL Server afin d'assurer les meilleures pratiques d'exploitation. Ce service sécurisé peut aider les administrateurs à anticiper d'éventuels problèmes de configuration, et à résoudre rapidement des problèmes identifiés en suivant des procédures par étapes. Il s'agit d'une édition spécifique de System Center.



1&1 HÉBERGEMENT

TOUS LES PACKS AU PRIX DU MOINS CHER !

**TOUS LES PACKS
HÉBERGEMENT
À SEULEMENT :**

1,99€
HT/mois
(2,38 € TTC/mois)
pendant les 3 premiers mois*

**Dernière offre de l'année.
Décidez-vous avant le 31 décembre !**



Vous voudriez choisir votre solution d'hébergement sans vous soucier du prix ? Chez 1&1, tous les packs d'hébergement vous sont actuellement proposés au prix du moins cher : seulement 1,99 € HT/mois* durant 3 mois quelque soit le pack !

Faites votre choix sur 1and1.fr, où vous découvrirez également nos serveurs, e-boutiques et noms de domaines à petits prix !

DOMAINES À PRIX CASSÉS :

le .fr à 3,99€ HT/an (4,77€ TTC/an),

le .info à 0,99€ HT/an (1,18€ TTC/an)* !

* Tous les Packs Hébergement sont au prix du Pack Initial (1,99 € HT/mois, soit 2,38 € TTC/mois). À l'issue des 3 premiers mois, les produits concernés sont aux prix habituels (Pack Confort à 5,97 € TTC/mois, Pack Pro à 11,95 € TTC/mois, Pack Premium à 23,91 € TTC/mois). Frais de mise en service : 5,97 € TTC (Pack Confort) ou 11,95 € TTC (Pack Pro, Pack Premium). Offre soumise à un engagement de 12 mois. Offre domaines applicable la première année au lieu du prix habituel de 6,99 € HT/an (8,36 € TTC). Conditions détaillées sur www.1and1.fr. Offres sans engagement également disponibles.



Appel non surtaxé

0970 808 911

www.1and1.fr

1&1

MySQL-Proxy : Le proxy MySQL qui vous simplifie la vie

MySQL proxy est un petit utilitaire qui s'introduit dans une architecture client/serveur et permet de faire de la réécriture de requêtes, de monitorer ses serveurs ou encore de bloquer des requêtes non voulues. Autant le dire de suite, un outil parfait pour les DBA MySQL.

Encore en pré-version et ce depuis un petit bout de temps, MySQL proxy vous permettra de faire pas mal de choses en matière d'administration supplémentaire ou d'apporter certaines améliorations pour la sécurisation de votre architecture. Qui dit alpha dit précautions en matière de déploiement dans un environnement de production. En effet la version actuelle n'est pas assez stable pour garantir une intégrité totale des données transitant par le proxy. Cependant, hormis certains aspects pour le moment déroutants il peut être assez intéressant dans le cas de l'utilisation d'une architecture dite de load balancing ou encore de failover, pouvant en même temps analyser, modifier ou filtrer les requêtes entrantes ou sortantes.

INSTALLATION, FONCTIONNEMENT

Dans un premier temps, il est nécessaire d'avoir tous les outils à notre disposition. Les machines de tests sont dans cet exemple, sous Fedora 12. Notons toutefois qu'il est disponible pour Windows. La version utilisée depuis le gestionnaire de paquet est la version 0.7.2, dite stable.

Bien évidemment nous préférons pour cet article, une version plus actuelle (0.8.1) et faire une installation manuelle téléchargeable depuis le site de MySQL, <http://dev.mysql.com/downloads/mysql-proxy/#downloads>.

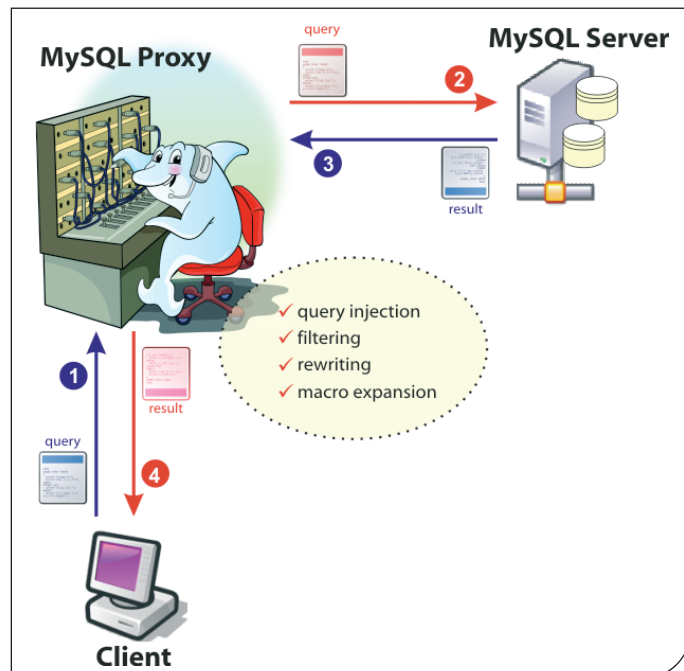
Voyons son déploiement dans une architecture load balancing. Une fois après avoir mis en place une architecture répliquée simple, nous vous proposons de séparer les opérations d'écriture et de lecture ainsi, créons notre compte proxy qui lui permettra d'assurer le relais et dispatchera les R/W selon les serveurs.

—celui-ci sur le master en mode distant

```
GRANT ALL PRIVILEGES ON *.* TO 'monproxy'@'%' IDENTIFIED BY 'proxpwd';
FLUSH PRIVILEGES;

--celui-là sur l'esclave en mode localhost
GRANT ALL PRIVILEGES ON *.* TO 'monproxy'@'localhost' IDENTIFIED BY 'proxpwd';
FLUSH PRIVILEGES;
```

Une fois les comptes créés, assurez-vous que tous les paramètres entre vos serveurs soient corrects notamment au niveau des bin-



logs. Notons que le déploiement du logiciel proxy est soit sur une machine différente ou soit sur le master.

Ouvrez un terminal et lançons mysql proxy avec le script de R/W splitting fourni et accompagné d'options. *proxy-backend-addresses* correspond à notre master et est en écriture, *proxy-read-only-backend-addresses* est pour nos esclaves et en lecture. Le script admin permet d'avoir le détail du statut des serveurs soit actif ou non en fonction de la charge dédiée au pool de connexion du script rwsplitting. Un mode debug s'en suit pour mieux analyser les requêtes et keepalive permet de garder le processus actif et de le relancer en cas de crash, cependant les données seront perdues.

```
mysql-proxy --proxy-lua-script=/home/ccif/mysql-proxy/lib/rw-splitting.lua --proxy-backend-addresses=192.168.110.51:3306 --proxy-read-only-backend-addresses=192.168.110.223:3306 --admin-username=root --admin-password=secret --admin-lua-script=/home/ccif/mysql-proxy/lib/admin.lua --log-level=debug
```

Ajoutons le blocage de certaines requêtes, exemple Temporary table qui n'est pas supportée mais qui peut être réécrite. Plaçons celui-ci tout en haut de notre script.

Script avec block queries uniquement

```
local commands = require("proxy.commands")
local tokenizer = require("proxy.tokenizer") --variable jeton qui correspond au parseur
local lb = require("proxy.balance")
local auto_config = require("proxy.auto-config")
--fonction d'expression régulière gérant les minuscules ou majuscules
function make_regexp_from_command(cmd)
```



```

local regexp= '^%s*';
for ch in cmd:gmatch('(.)') do
    regexp = regexp .. '[' .. ch:upper() .. ch:lower() .. ']'
end
return regexp end
--nom de notre commande qui est basée sur les expressions régulières
local CREATE_REGEXP = make_regexp_from_command('create')
--requêtes à filtrer
queries_to_filter = {{ prefix = CREATE_REGEXP, keywords = {
'CREATE', 'TEMPORARY', 'TABLE'} }}
--fonction d'erreur avec une réponse personnalisée du proxy
function error_result (msg)
    proxy.response = { type = proxy.MYSQLD_PACKET_ERR,
errmsg = msg, errcode = 7777, sqlstate = 'X7777', }
return proxy.PROXY_SEND_RESULT end --fin du bloc de configuration

```

```

local tokens = tokenizer.bare_tokens(full_tokens,
true)

local found = 0
local requested = #query_to_filter.keywords
for j, keyword in pairs(query_to_filter.keywords)
do --on va toutes les parser
    for k, token in pairs(tokens) do
        if token:upper() == keyword then found
= found + 1 break end
    end -- fin for k
end --fin for j
if found == requested then return error_result('la
commande est <' .. table.concat(query_to_filter.keywords, ' ')
.. '> non autorisé !') end --fin if
end
end --fin do
--fin de la configuration

```

Puis effectuons la modification suivante dans le corps du script.

```

--if cmd.type == proxy.COM_QUERY then --print(" query
" .. cmd.query) -- end
end --à ne pas commenter ici
--suite, nous modifions le retour de paquet selon les méthodes
vues plus haut.
local query = packet:sub(2) --requêtes renvoyées
for i, query_to_filter in pairs(queries_to_filter) --les
requêtes arrivent
do
    if query:match(query_to_filter.prefix) then -- si
correspondance
        local full_tokens = tokenizer.tokenize(query)

```

En conclusion, cet outil est pour le moment très efficace et a de beaux jours devant lui, même s'il ne couvre pas toutes les commandes de MySQL telles que les procédures stockées ou commençant par SET. Si vous voulez une solution de load balancing intégrée, le moteur NDB CLUSTER le gère par défaut et est assez robuste pour un environnement de production.

■ Nabil Balaci

suit actuellement un cycle d'ingénierie logiciel au sein de l'Exia.Cesi de Reims. Passionné de sécurisation informatique où il a collaboré à l'écriture de plusieurs articles et de nouvelles technologies il est aussi membre de la communauté Microsoft en tant que MSP.

nabil.babaci@yahoo.fr

L'INFO permanente

- L'actu : le fil d'info quotidien de la rédaction
- La newsletter hebdo : abonnez-vous, comme 46 000 professionnels déjà. C'est **gratuit** !

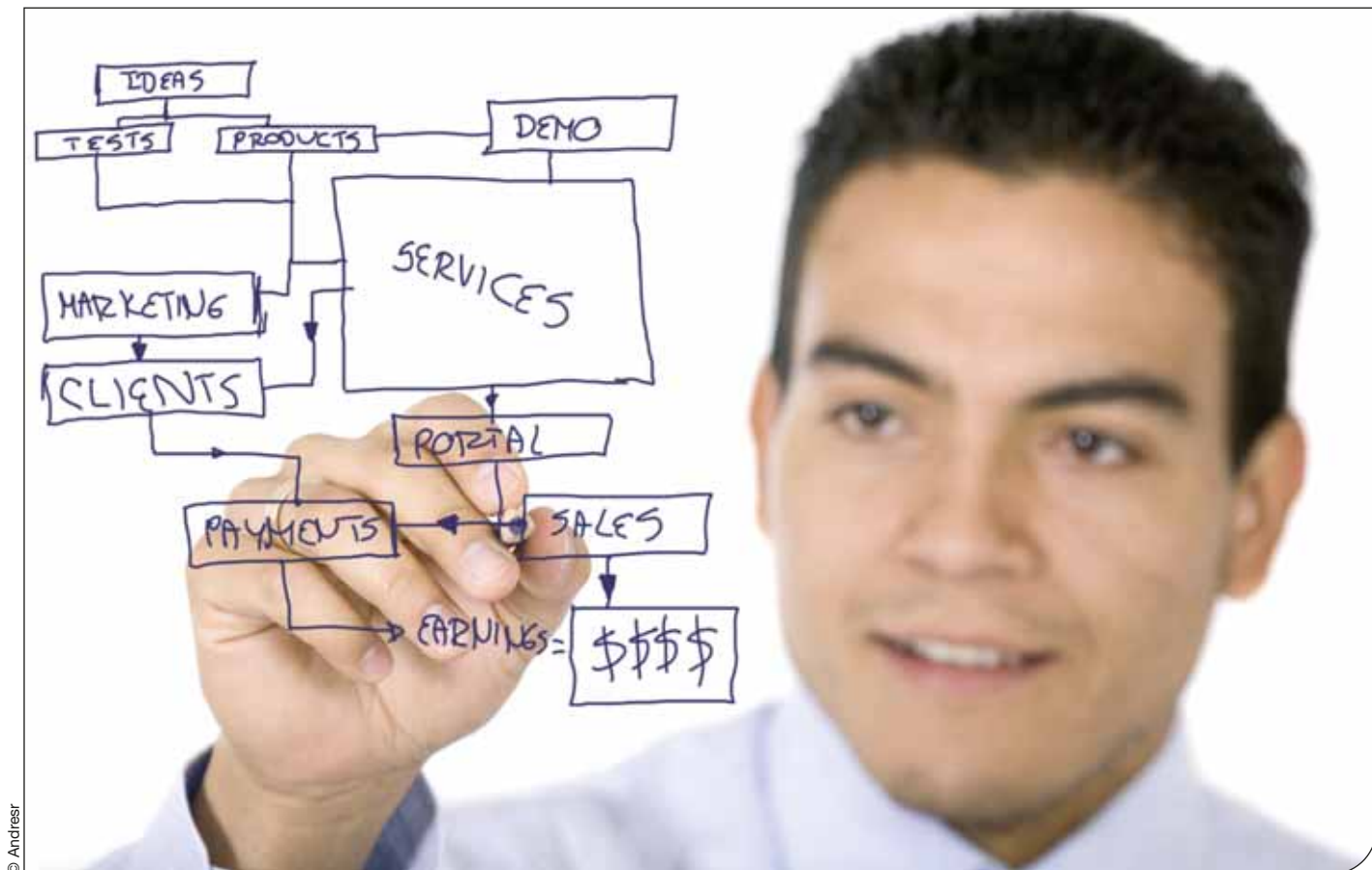
C'est PRATIQUE !

- Le forum : modéré par la rédaction et les auteurs de Programmez!, rejoignez les forums techniques de programmez.com
- Les tutoriels : une solution en quelques clics !
- Le téléchargement : récupérez les nouveautés.

www.programmez.com



Modélisation et développement : à chacun son modèle

2^e partie

© Andress



Fathi Bellahcene,
Consultant/Formateur
chez Winwise.
7 ans d'expérience dans
le développement logiciel.
Son Blog :
<http://blogs.codes-sources.com/fathi>

UML et les développeurs

Les développeurs d'aujourd'hui doivent de plus en plus être capables de spécifier, de modéliser et de concevoir avant de réaliser leurs projets. Pour ma part, j'utilise l'approche 'Top-Down' basée sur UML que l'on peut résumer en trois étapes : analyse, design et réalisation. La première étape consiste à collecter le maximum d'informations pour produire les diagrammes UML de haut niveau comme les Cas D'utilisations (que l'on précise avec les diagrammes d'activités et de séquences). Ainsi les différents acteurs et la liste des fonctionnalités à produire sont bien identifiées. Dans la phase de Design, on

produit des diagrammes de séquences et de classes afin de matérialiser de manière statique et dynamique le stockage et la circulation de l'information. On s'attache à respecter les principes SOLID et à utiliser au maximum les Design Patterns. Puis, on effectue des regroupements logiques (diagramme de composants), on décide sous quelle forme physique ces composants vont être réalisés (webservice, dll,...) et comment les flux circulent via les diagrammes de déploiement. Enfin, dans la dernière étape, l'utilisation de logiciel qui génère du code automatiquement (Entreprise Architect, Modelio, VS2010) est un

atout considérable pour la productivité. Outre le fait de nous faire gagner du temps, il permet aussi de limiter l'écart entre le code produit et le modèle et de limiter l'impact des modifications apportées au modèle sur la qualité du code. Pour finir, on retranscrit au maximum les diagrammes de haut niveau sous forme de tests unitaires ou fonctionnels. Bien que principalement utilisée sur des projets classiques tels 2TUP ou RUP, il est aussi possible de l'appliquer à des projets agiles. On applique cette méthode d'analyse aux items définis dans le Product Back log et pendant le sprint on effectue les étapes uniquement sur les cas sélectionnés. Bien évidemment, on reste pragmatique en réalisant uniquement les diagrammes qui ont une valeur ajoutée pour nos tâches. ■

Une semaine d'immersion dans une équipe d'ingénierie par les modèles

Lundi matin, il est 9 heures je termine mon café avec mon équipe et nous voilà reparti pour une belle journée. Mon rôle ? Je dirige un projet assez significatif – environ 5 000 points de fonction – de développement spécifique pour une application métier d'une grande entreprise. Quoi de neuf sous le soleil ?

Une particularité ... pas une seule ligne de code n'a été écrite. Toute l'application est générée à partir de sa modélisation en utilisant un générateur commercial français. Et je dois dire que la productivité est au rendez-vous, tout comme le plaisir.

Voici notre histoire sur ce projet, au cours d'une semaine standard.

DÉBUT DE SEMAINE

Lundi

La semaine débute, nous procédons à la livraison d'un lot d'évolution. Et comme souvent, le week-end a été propice à la créativité. Notre maîtrise d'ouvrage nous sollicite : la liste des utilisateurs s'ouvre à une nouvelle catégorie : des usagers des filiales ou des cadres en déplacement avec leur seul smartphone comme point d'accès.

Une réunion a lieu dans la foulée pour cerner les besoins exacts et voir quand et comment intégrer cette demande ?

L'avantage de ces réunions est que nous travaillons dans un mode « What you see is what you get ». Nous travaillons par maquettage, la maîtrise d'ouvrage nous décrit l'ergonomie des écrans dans un format directement utilisable par notre générateur d'application (page xhtml [web]ou xml[RIA]) - . Nous passons en revue les nouveaux champs et constatons que tout est bon dans l'ensemble, il suffit d'ajouter une règle de validation et de faire une modification du modèle de données. Nous passons donc à la définition d'une nouvelle règle de gestion métier. La MOA nous exprime cette règle de gestion, sous des aspects uniquement métiers : étapes du processus, objets

manipulés. Pendant ce temps nous modélisons ce processus dans un diagramme UML simplifié. Après revue l'ergonomie est validée, la navigation entre écrans également, ainsi que les processus appelés et les données qui seront manipulées. L'évolution s'avère simple, je confirme donc qu'elle sera incluse dans la livraison de la semaine. Fin de la réunion, elle a duré 30 mn et la MOA n'a jamais eu à formuler ses besoins en UML.

De retour dans notre open space, je demande que les nouvelles fonctions soient implémentées immédiatement. Gros avantages de notre Atelier MDA : l'intégralité de l'application est modélisée, tout le code est généré. L'analyste de dépendance n'identifie aucun impact sur la modification du modèle d'entité mais en trouve une sur une règle de navigation. Nous savons ce que nous avons à faire, nous ne rateons rien ; en fin de matinée tout est bouclé. Le validateur de modèle nous indique qu'il n'y a pas d'erreur de modélisation, nous passons aux tests fonctionnels. Nous activons notre model debugger qui permet de placer nos points d'arrêts dans la modélisation et de co-exécuter modèle et application générée. Nous basculons en vue pas à pas dans les règles de gestion. Pas de bug, tout se passe correctement, nous sommes passés dans toutes les branches de l'algorithme (pratique la vue graphique pas à pas dans l'exécution du modèle). Par conséquent l'application est déployée pour validation par la MOA. Nous n'avons pas besoin de les accompagner à ce stade car les écrans, la navigation et les processus sont rigoureusement identiques aux éléments utilisés lors de notre réunion

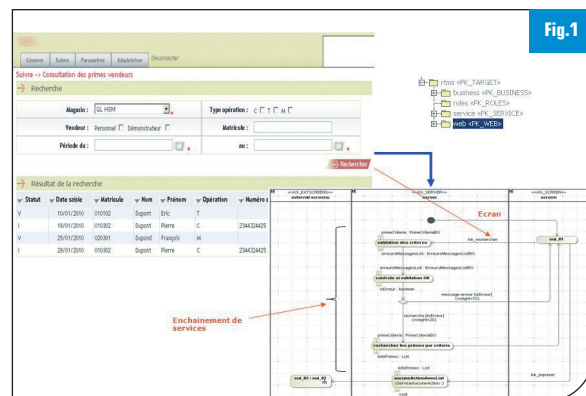


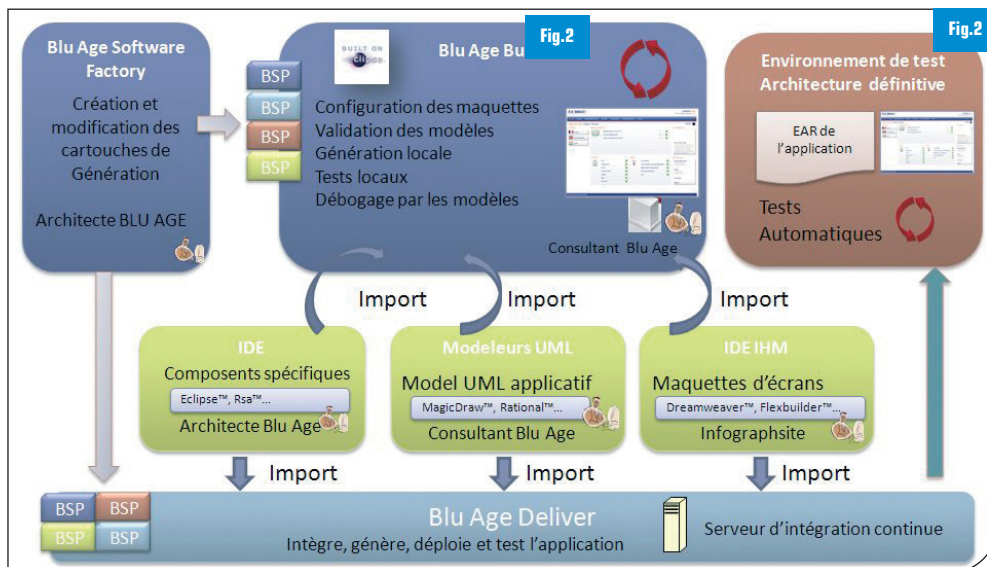
Fig.1

d'expression des besoins. Le modèle et les maquettes sont par définition exactement identiques à l'application produite. Le vrai dicton américain est d'ailleurs « What you see is what you get ... and you get what you see ». Cela n'a jamais été aussi vrai [Fig.1].

BIENTÔT LA LIVRAISON

Mardi

La livraison approche. Concrètement cela veut dire que nous validons que toute la chaîne d'intégration continue est à jour. En effet, notre atelier MDA est composé en deux univers. Un environnement poste de travail pour travailler en local. Sur ce poste nous modélisons, et enrichissons les maquettes d'écran pour les rattacher aux modèles. Une fois notre travail validé nous procédons à une génération locale de l'application. Cet IME (Integrated Modeling Environment) nous permet de valider le modèle et de l'exécuter pas à pas avec le modèle débogueur. C'est là qu'intervient la solution serveur qui implémente la chaîne d'intégration continue de tous nos projets. L'ensemble des modèles y sont gérés (versionning, branches, gestion collaborative et check in/out avec la granularité souhaitée) ainsi



que l'ensemble des maquettes et l'ensemble des configurations d'architecture. Tous ces éléments sont rattachés par projet. Ceci nous permet chaque nuit de procéder à la génération, au déploiement et à l'exécution automatisée de nos cas test. Je me connecte donc à l'interface d'administration. Je commence par le tableau de bord « nightly built ». L'application est générée et déployée. L'ensemble des cas tests est bon. Il y a deux tests qui sont à analyser de plus près, mais je poursuis la planification de la génération de la release. Je choisis donc l'architecture, ou plutôt les architectures. En effet cette application doit marcher à la fois de manière centrale (WEB RIA) et en mode autonome déconnecté. Notre architecte a bien défini les deux workflows d'architecture. J'associe donc la version 4.1 du modèle de l'application à ces deux workflows et programme une double génération pour la nuit de mercredi à jeudi. Comme la modélisation de l'application est indépendante de l'architecture, je n'ai pas besoin d'adapter les entrants (spécification de l'application) en fonction de l'architecture. C'est un vrai soulagement de ne plus avoir à contrôler l'alignement fonctionnel de différents codes sources d'une même application en fonction de son architecture de déploiement [Fig.2].

Puisque tout est prêt, je vais m'assurer que les derniers tests sont automatisés et que la qualité de code est toujours bonne (PMD, Checkstyle,

CAST et Sonar). Toujours les mêmes résultats : que de très bonnes notes. Vraiment nos architectes ont bien fait leur boulot en ajustant nos car-touches de génération. Des notes maximales en respect des patterns, en performance, en maintenabilité. D'ailleurs cela avait été déterminant dans le passage de Flex3 à Flex4, Cairngorn avait fortement évolué et il avait fallu tout repenser [Fig.3].

Mercredi

Nous sommes à 24 heures de la génération de notre release. Bien que fan de la série (NDLR : série TV 24 heures), je n'aime pas trop vivre mes projets avec le même suspens. Mais c'est bien la chose qu'un outil ne peut pas contrôler. Quand le métier a besoin de quelque chose il faut le faire !

























Généralement l'imminence d'une livraison est source de quelque stress, ça booste les schémas mentaux et les idées surgissent. Ou l'imprévu qui était caché jusque là en profite pour sortir tel le diable à ressort de mon enfance. Bref il faut rester prêt à réagir.

Cette fois je ne y coupe pas. Il s'avère qu'une de nos filiales a décidé de basculer sa flotte vers Apple et va équiper ses cadres itinérants d'Ipad. On me précise qu'il n'est pas envisageable de décaler la mise à disposition de la release pour cette filiale. J'essaye d'expliquer que c'est trop tardif. On me rétorque que la migration Flex3 vers Flex4 s'était faite par simple changement de cartouche ... bref l'enjeu technologique n'est pas comparable à l'urgence du métier. Il faut y aller.

Dernière vérification

Nous vérifions quelles sont les cartouches de génération que nous avons achetées auprès de l'éditeur. Il s'avère que nous sommes en mode locatif, ce qui nous permet d'ajuster le nombre de licences en fonction de l'utilisation mensuelle. En complément nous avons une liberté d'usage de toutes leurs cartouches de génération. Je télécharge donc deux cartouches pour tester rapidement et faire un choix : une cartouche Objective-C spécialisée pour Apple et une cartouche HTML5 plus générique. Nous générons rapidement une appli-

Fig.3

ADMIN	BEN	CAL	CYC	Dictionary	FIN	GEN	GRP	HMO jobs	HMO screens	IMD	LOG	MIS	PLAN	SERVICE COMPOSITE	SUB	TRG	Tous	Utilities	WKS
S	W	Tâche :								Dernier succès				Dernier échec					
		BEN - SCREEN - BEBKT								9 mo. 8 j (#4)				N/A					
		BEN - SCREEN - BEBLK								9 mo. 8 j (#5)				N/A					
		BEN - SCREEN - BECAT								9 mo. 8 j (#7)				10 mo. (#1)					
		BEN - SCREEN - BEDET								9 mo. 8 j (#7)				10 mo. (#1)					
		BEN - SCREEN - BELST								9 mo. 8 j (#6)				9 mo. 17 j (#4)					
		BEN - SCREEN - BENAR								9 mo. 8 j (#6)				N/A					
		BEN - SCREEN - BESEQ								9 mo. 8 j (#4)				9 mo. 17 j (#3)					
		BEN - SCREEN - BMENU								9 mo. 8 j (#7)				10 mo. (#1)					
		BEN - SCREEN - BRFF								9 mo. 8 j (#7)				9 mo. 10 j (#5)					
		BEN - SCREEN - RVNAR								9 mo. 8 j (#4)				N/A					
		BEN - SCREEN - RXREF								9 mo. 8 j (#6)				N/A					
		BEN - SCREEN - DISBN								9 mo. 8 j (#4)				N/A					
		BEN - SCREEN - EECAT								9 mo. 8 j (#6)				N/A					
		BEN - SCREEN - EECBG								9 mo. 8 j (#6)				9 mo. 17 j (#4)					
		BEN - SCREEN - EMAP								9 mo. 8 j (#6)				N/A					
		BEN - SCREEN - LFBN								9 mo. 8 j (#6)				N/A					
		BEN - SCREEN - MEDBN								9 mo. 8 j (#7)				10 mo. (#1)					

cation prototype, que nous avons conçue pour ce genre de situation, elle nous permet de faire de la validation de framework quand nous faisons évoluer nos cartouches de génération. Les tests fonctionnels sont bons dans les deux cas.

Aussitôt, nos architectes mettent au point deux nouveaux workflows de génération (Objective-C, html5). 15 minutes après c'est prêt. Je me reconnecte, crée une copie du projet pour ne pas faire d'interférence et déclenche une génération immédiate vers ces deux cibles d'architecture. Je décide de n'exécuter que les tests des services unitaires et pas les tests basés sur les écrans car je n'ai pas le temps de faire adapter les pages à la taille de l'écran Iphone.

Avec un « sachant » fonctionnel, nous testons donc quelques écrans représentatifs de l'ergonomie et de la complexité de l'application. Après une heure nous voyons que l'architecture Objective-C nous satisferait plus mais elle requiert plus d'ajustement des maquettes d'écran. L'architecture HTML5 nous permet d'avancer plus rapidement mais les écrans conçus pour des écrans de PC seront à l'étroit sur un Iphone.

Nous faisons une réunion avec la maîtrise d'ouvrage pour montrer le résultat. Les deux solutions sont viables. Pour ne pas prendre des risques sur les tests nous proposons de générer et déployer cette semaine l'application Flex, et de déployer l'application en HTML5 la semaine prochaine, le temps d'ajuster quelques écrans. Nous déciderons par la suite si nous maintenons le choix HTML5 ou si nous passons sur Objective-C. Nous tombons d'accord. Ce fut chaud. La livraison de cette semaine n'est pas impactée.

Jeudi matin

Aujourd'hui c'est double ration. Je dois m'occuper de passer toute l'application en validation avec la MOA et je dois m'occuper de lancer le chantier HTML5.

Je vais voir Thomas à qui je confie le chantier HTML5. Thomas est un architecte Java à qui nous avons confié la responsabilité d'être le référent tech-

nique sur l'ensemble de nos projets (il s'occupe surtout de la modélisation des données à des fins de performance et du coaching). Il a pour mission d'évaluer la charge de travail pour adapter l'ergonomie la semaine prochaine et voir si nous aurons besoin de scinder des processus sur plus d'écrans. Il va être force de suggestion pour utiliser au mieux la petite taille des écrans Iphone. Evan, lui aussi issu du chaudron à fabriquer les archi-

tectes JEE, va se pencher sur le choix définitif entre HTML5 et Objective-C, ainsi que sur les patterns mis en œuvre par les cartouches que nous venons de télécharger. Il validera également quels outils intégrer pour automatiser les tests fonctionnels pour Iphone et Ipad. Tout deux ont construit notre technique de validation de framework. En effet, partant du principe que la même modélisation produit l'application sur l'architecture de notre choix ils ont réalisé qu'il nous était possible, premièrement de réduire considérablement le nombre de versions différentes des mêmes framework java au sein de notre système d'information et, deuxièmement de générer une application déjà existante -sans la modifier- sur une nouvelle version de nos Frameworks afin de faire des tests de non régression d'architecture.

Nous en avons tiré quatre bénéfices majeurs :

- Le temps est fini ou le triplet version hibernate/version Spring, version de la couche de présentation était unique pour chaque application.
- Lors de la migration Flex3 vers Flex4 nous avons immédiatement constaté des problèmes (ex : transfert de données vers une pop-up).
- Thomas et Evan ne sont plus les pompiers sur nos projets et se consacrent à l'ensemble des projets (performance du modèle de données, des requêtes, respect des bonnes pratiques de modélisation) et font un travail d'enrichissement de nos patterns pour avoir une qualité de code sans précédent et des architectures souples et perfor-

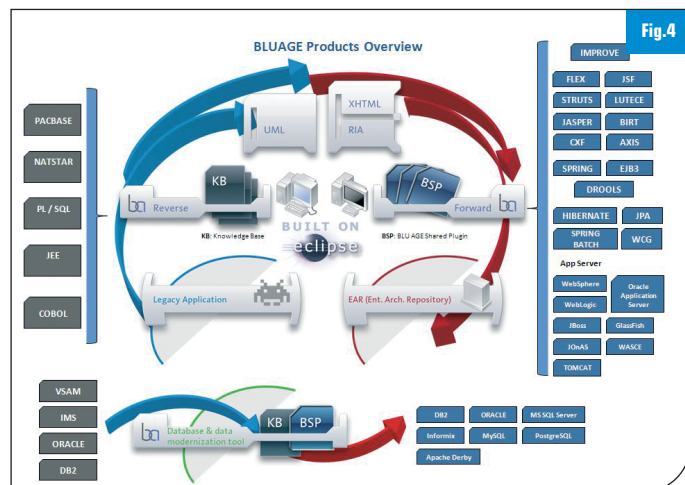


Fig.4

mantes. Notre application la plus significative compte quand même 4 000 utilisateurs simultanés.

- Les métiers n'entendent plus le mot web service à chaque réunion, nos architectes sont satisfaits et deviennent de véritables experts, à la fois métier et technique. Nous avons même lancé un chantier de factorisation pour faire une architecture de type SCA, non par les composants mais par la modélisation.

LE WEEK-END APPROCHE

Vendredi

Finalement la semaine se termine aussi bien qu'elle a commencé. La validation se passe bien, le passage en pré production sera visiblement prononcé ce soir. Je peux donc confirmer aux architectes qu'ils peuvent factoriser nos éléments de modélisation dans notre base de connaissance commune de composants métiers. J'ai pu faire mon bilan de charge sur cette release. Mon boss me demande si je tiens les objectifs de réduction de coûts et de délais habituels depuis que nous sommes MDA. Je lui réponds avec un grand sourire. En guise de conclusion je me permettrais de citer un développeur Java d'un certain talent, qui m'avait dit après son premier projet BLU AGE (2 000 points de fonction) : « Je fais enfin mon métier ».

■ Alexis Henry

Directeur Opérations - BLU AGE

www.bluage.com

Modélisation : simplifier le problème du développeur

Dans cet article, nous allons voir ensemble comment l'utilisation d'un générateur de code piloté par un modèle métier peut simplifier la vie d'un développeur. A titre d'exemple nous utiliserons le produit CodeFluent Entities, une fabrique logicielle pilotée par les modèles éditée par SoftFluent pour créer un site ASP.NET de gestion de contacts. Le site créé dans cet exemple est volontairement simpliste pour, premièrement, illustrer ce qui est possible avec de la modélisation combinée à de la génération de code et, deuxièmement, montrer une démarche que chaque développeur peut dès aujourd'hui mettre en œuvre dans ses applications.

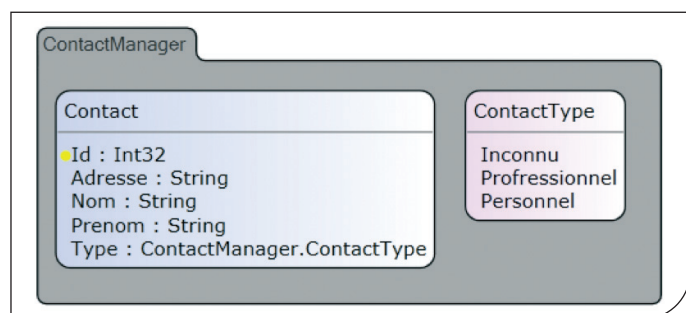
CRÉATION DE L'APPLICATION

Modéliser l'application

La première étape est bien sûr de modéliser mon application. La modélisation d'une application avec CodeFluent Entities implique deux étapes :

- La déclaration des éléments fonctionnels de mon application,
- La définition de mon architecture cible ainsi que des technologies utilisées.

La déclaration des éléments fonctionnels consiste à définir les concepts métier de mon application (un contact, un type de contact), leurs propriétés (un contact a un nom, prénom, une adresse), leurs relations (un contact ne peut être que d'un seul type) et leurs règles métier (je ne peux pas avoir d'adresse vide par exemple).



Une fois la logique métier de mon application définie, il ne me reste plus qu'à définir son architecture (ici client/serveur), et les technologies à utiliser (SQL Server comme base de données, et un modèle objet en C#). Cette étape est ainsi cruciale puisqu'elle consiste à définir la sortie de ma génération. Pour CodeFluent Entities, ceci se fait via la définition de producteurs qui sont des modules opérant la traduction de mon modèle indépendant de toute plateforme en code spécifique et optimisés pour une plateforme.

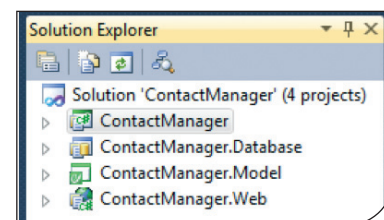
De cette manière, nous avons dissocié le métier de l'application et les technologies utilisées, ce qui est capital à la pérennité d'une application puisque le cycle de vie des technologies est bien plus court que celui du métier de l'application.

La fabrique logicielle va ensuite me générer des composants fonctionnels et prêts à l'emploi. Ainsi, d'une part toute la tuyauterie de

l'application a été générée, mais en plus le code produit implémente toutes les interfaces .NET et respecte toutes les bonnes pratiques nécessaires pour que je puisse facilement créer mon site web par-dessus. De ce fait, je vais pouvoir concentrer mes efforts sur la plus-value de mon application : sa logique métier et son interface graphique.

Utiliser le code généré

Mon application correspond à une solution Visual Studio, elle-même composée de quatre projets :



- un projet CodeFluent Entities dans lequel j'ai modélisé mon application,
- un projet Database dans lequel j'ai généré mes scripts de base de données,
- un projet C#, lequel contient le modèle objet métier généré,
- et enfin un projet ASP.NET contenant l'interface graphique de mon application.

Le projet ASP.NET référence mon modèle objet métier et je vais changer le contenu de la page par défaut de mon site (default.aspx) pour qu'elle liste tous mes contacts. Comme les classes de mon modèle objet prennent en charge la liaison de données je n'ai plus qu'à lier mon écran à mon objet métier, qui lui-même consomme la base de données. Voici donc le code de ma page ASP.NET :

```
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
  <h2>
    Welcome to ASP.NET!
  </h2>
  <p>
    <asp:ObjectDataSource runat="server" ID="DataSource"
      DataObjectTypeName="ContactManager.Contact"
      TypeName="ContactManager.ContactCollection" SelectMethod=
        "LoadAll" />
    <asp:GridView runat="server" DataSourceID="DataSource" Auto
      GenerateColumns="false">
      <EmptyDataTemplate>No data!</EmptyDataTemplate>
    </asp:GridView>
  </p>
</asp:Content>
```

```

<Columns>
    <asp:BoundField runat=»server» DataField=»Id»HeaderText
=>Id» />
    <asp:BoundField runat=»server» DataField=»Prenom» Header
Text=»Prénom» />
    <asp:BoundField runat=»server» DataField=»Nom» Header
Text=»Nom» />
    <asp:BoundField runat=»server» DataField=»Adresse» Header
Text=»Adresse» />
</Columns>
</asp:GridView>
</p>
</asp:Content>

```

Ma page ASP.NET est maintenant liée à mon objet métier, il faut désormais spécifier une chaîne de connexion à mon application pour que ma couche métier sache où puiser ses données.

```

<configuration>
  <configSections>
    <section name=»ContactManager» type=»CodeFluent.Runtime.Code
FluentConfigurationSectionHandler, CodeFluent.Runtime» />
  </configSections>
  <ContactManager connectionString=»server=(local);database=
ContactManager;Integrated Security=true» />
  (...)
</configuration>

```

FAIRE ÉVOLUER MON APPLICATION

Evolution du métier

Maintenant que nous avons une version de base de notre application, nous allons la faire évoluer et c'est justement là que le développement piloté par les modèles s'avère particulièrement bénéfique. Il se trouve que maintenant mon application doit gérer des numéros de téléphones. Sans une fabrique logicielle pilotée par les modèles il faudrait donc que je repasse sur ma table pour ajouter la colonne correspondante, sur le code d'accès aux données, sur la classe correspondante et bien sûr sur mes écrans pour ajouter le champ en question.

Ici, il me suffit d'ajouter un champ « Téléphone » à mon modèle, de générer à nouveau et toutes mes couches vont automatiquement être mises à jour.

Plus complexe, mais dans la même lignée je peux ajouter une nouvelle fonctionnalité à mon application : je veux pouvoir trier mes contacts par nom et prénom. Tout comme pour la nouvelle propriété, il me suffit de mettre à jour mon modèle, et la procédure stockée de chargement de données va être mise à jour pour prendre en compte un tri sur la colonne indiquée par les couches supérieures.

```

CREATE PROCEDURE [dbo].[Contact_LoadAll]
(
    @_orderBy0 [nvarchar] (64) = NULL,
    @_orderByDirection0 [bit] = 0
)
AS
SET NOCOUNT ON
SELECT [Contact].[Contact_Id], [Contact].[Contact_Prenom], [Contact]
.[Contact_Nom], [Contact].[Contact_Adresse], [Contact].[Contact_

```

```

_Type], [Contact].[Contact_Telephone], [Contact].[_trackLast
WriteTime], [Contact].[_trackCreationTime], [Contact].[_track
LastWriteUser], [Contact].[_trackCreationUser], [Contact].
[_rowVersion]
FROM [Contact]
ORDER BY CASE
    WHEN @_orderBy0 = '[Contact].[Nom]' AND @_orderByDirection0
= 0 THEN [Contact].[Contact_Nom]
END ASC,
CASE
    WHEN @_orderBy0 = '[Contact].[Nom]' AND @_orderByDirection0
= 1 THEN [Contact].[Contact_Nom]
END DESC,
CASE
    WHEN @_orderBy0 = '[Contact].[Prenom]' AND @_orderByDirection0
= 0 THEN [Contact].[Contact_Prenom]
END ASC,
CASE
    WHEN @_orderBy0 = '[Contact].[Prenom]' AND @_orderByDirection0
= 1 THEN [Contact].[Contact_Prenom]
END DESC
RETURN
GO

```

Il ne me reste plus encore une fois qu'à mettre à jour ma page ASP.NET pour qu'elle indique à ma procédure stockée le sens du tri à appliquer ainsi que la colonne sur laquelle l'appliquer et le tour est joué.

Evolution technique

De la même manière que nous avons pu facilement modifier le métier de l'application, la modélisation pilotée par les modèles permet facilement de changer l'architecture de l'application ou de prendre en charge une nouvelle plateforme. Pour cela, il suffit d'ajouter le producteur voulu et le tour est joué : le producteur va générer le code nécessaire pour que mon application prenne en charge la nouvelle plateforme. Par défaut, une quinzaine de producteurs sont disponibles, tels que Oracle Database, WCF, SharePoint, et bien d'autres.

Conclusion

Nous avons vu dans cet article et par le biais du développement d'une application de gestion de contact ce que le développement piloté par les modèles peut apporter au développeur : c'est d'abord un gain de productivité puisque le développeur peut maintenant concentrer ses ressources sur les points clés de l'application, un gain de qualité puisque la majeure partie du code est maintenant générée et par conséquent garantie par la fabrique logicielle, et enfin un gain de flexibilité puisque le métier est dissocié de la technique et prendre en charge une nouvelle technologie revient à ajouter un nouveau producteur.

Même si l'exemple montré est simple, il illustre ces gains de manière très concrète. Ces gains sont en fait encore plus significatifs dans le cas d'une application complexe, car, sans une approche pilotée par les modèles, les besoins et évolutions se combinent et les coûts se multiplient.

■ Carl Anderson (*SoftFluent*)

Créer son outil de modélisation de processus métier en utilisant Workflow Foundation 4

Lorsque le besoin d'automatiser des processus métier se fait sentir, il est fréquent de vouloir créer une application et des Workflows associés qui répondent à toutes les problématiques métier. Cependant il est parfois difficile pour un développeur de comprendre toutes les subtilités métiers ainsi que de prévoir tous les cas possibles sans avoir à effectuer des changements très fréquemment. Pourquoi ne pas prendre le problème sous un autre angle et fournir de préférence aux utilisateurs les outils pour pouvoir générer leurs propres Workflows ?

De quoi vont avoir besoin nos utilisateurs ? Un outil de modélisation de Workflows, simple et facile à prendre en main, et sans l'obligation d'utiliser Visual Studio. Ils auront également besoin, pour modéliser leur processus, d'activités très simples à utiliser et qui ne demandent aucune connaissance technique.

Dans cet exemple, nous allons développer des outils permettant de modéliser des Workflows liés au recrutement. Pour cela, nous allons mettre en place une architecture comprenant :

- Une base de données permettant de stocker l'état de nos processus (les nouvelles demandes, les tâches de chaque utilisateur)
- Un service (WCF) pour réaliser des opérations sur la base (créer une nouvelle demande, mettre à jour le résultat d'un entretien, etc.)
- Un jeu d'activités WorkflowFoundation personnalisées
- Des designers pour ces activités afin d'avoir un rendu graphique agréable pour l'utilisateur final et une ergonomie adaptée au besoin
- Une application WPF permettant aux utilisateurs de modéliser les nouveaux processus.
- Un service pour héberger nos processus
- Une application cliente pour manipuler et faire avancer nos processus

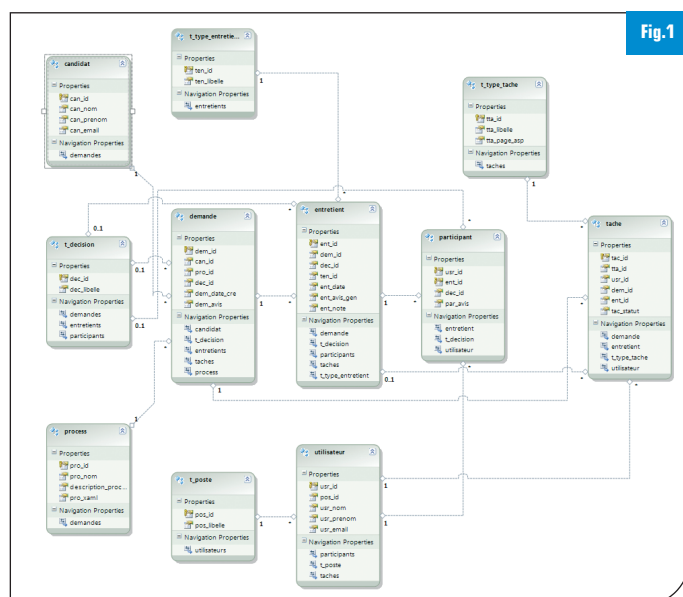
La base de données [Fig.1]

Comment se déroule généralement un processus de recrutement ? Un candidat va se présenter, ce qui va générer une nouvelle demande. Ce candidat va ensuite passer un ou plusieurs entretiens. Chaque entretien va être affecté à un ou plusieurs participants qui vont à la fin donner leur accord.

En terme « Workflow » chaque processus créé aura son propre identifiant et sera stocké sous sa forme déclarative (en XAML). Chaque utilisateur qui sera affecté à un entretien aura une « tâche » qu'il devra terminer pour faire avancer le processus.

Créer des activités personnalisées

Nous allons maintenant rentrer dans le vif du sujet et réaliser nos activités. Tout d'abord nous allons créer un nouveau projet de type « Activity Library » dans Visual Studio, puis rajouter un nouvel élément de type « Activity ». Nous allons créer une première activité personnalisée nommée « Nouvelle demande » qui va être l'activité de



départ pour tous nos Workflows. Cette première activité doit faire plusieurs choses :

- Démarrer un nouveau Workflow
- Être accessible pour qu'une application cliente puisse démarrer un nouveau Workflow
- Créer une nouvelle demande dans la base
- Retourner l'ID de la nouvelle demande qui vient d'être créée

Pour les deux premiers points nous allons utiliser l'activité « Receive » qui va nous permettre d'exposer notre Workflow comme un service WCF. Cette activité est très simple à utiliser. Il suffit de saisir l'OperationName (Start fera très bien l'affaire), ainsi que les différents paramètres (par exemple nom et prénom du candidat). Ne pas oublier de cocher la case « CanCreateInstance » afin qu'une nouvelle instance de Workflow soit créée lors de l'appel à cette méthode.

Maintenant, nous allons ajouter une activité permettant d'insérer une nouvelle demande dans la base. Ici pas question d'utiliser du code ! Dans WorkflowFoundation, il nous suffit de rajouter une réf-

rence à un service WCF existant pour générer automatiquement des activités qui permettent d'appeler ce service directement ! Aucun paramétrage spécifique n'est requis [Fig.2].

Enfin nous allons ajouter un argument de sortie à notre activité « Nouvelle demande » qui sera bindé au résultat de l'activité « CreateDemande » généré précédemment. Il suffit de cliquer sur l'onglet « Arguments » en bas du designer et de rajouter un argument qui aura pour direction « OUT » et de type « Int32 » [Fig.3].

Et voilà notre activité est terminée ! [Fig.4].

Passons maintenant à la suite ! Un peu comme on factorise du code, nous allons créer nos blocs d'activités afin de pouvoir les assembler par la suite. Il nous faut une activité qui va nous permettre de créer une tâche de type « planifier un entretien ». Pour cela, nous allons créer une deuxième activité personnalisée qui va effectuer les opérations suivantes :

- Créer un nouvel entretien en base
- Créer une nouvelle tâche en base
- Utiliser une activité Receive pour la réalisation de cette tâche
- Mettre à jour l'entretien après réalisation de la tâche
- Clôturer la tâche en base

Quand on expose un Workflow en tant que service WCF nous devons avoir un moyen de différencier chaque instance différente de notre Workflow. Le client ne va avoir accès qu'aux différentes opérations définies avec nos activités Receive. Dans le cas présent :

- Une opération Start
- Une opération PlanifierEntretien

Lorsque l'on fait appel à l'opération Start, une nouvelle instance de notre Workflow sera créée. Par la suite nous allons devoir spécifier un paramètre à l'opération PlanifierEntretien pour cibler la bonne instance. Cette problématique est gérée dans WorkflowFoundation par l'intermédiaire de la corrélation. Nous allons spécifier à notre activité Receive un paramètre qui sera notre indicateur de corrélation. Pour rester dans la simplicité, nous allons utiliser l'id de la tâche nouvellement créée en base comme paramètre de corrélation. Pour cela, il faut utiliser l'activité « InitializeCorrelation » et indiquer dans la propriété « CorrelationData » la variable à utiliser. De même, dans l'activité Receive nous allons indiquer sur quel paramètre effectuer cette corrélation. Il suffit de cliquer sur la propriété « CorrelatesOn » dans Visual Studio et sélectionner la variable IdTask [Fig.5]. Notre activité est terminée ! [Fig.6].

Ensuite, en suivant exactement le même modèle, nous allons créer une troisième activité personnalisée « EntretienAvis », qui va créer une tâche pour chaque participant afin qu'il puisse donner son avis sur un candidat suite à un entretien.

Une fois les activités de base créées, nous allons pouvoir composer des activités de plus haut niveau comme par exemple une activité de type « EntretienRH ». Cette activité sera uniquement composée des deux sous-activités réalisées plus haut [Fig.7].

Nous pouvons par la suite créer autant d'activités que l'on souhaite en fonction des besoins utilisateurs. Pour ma part j'ai créé une activité supplémentaire appelée « EntretienAutre », qui utilise une activité « ParallelForEach » et qui prend en argument d'entrée une liste d'Id utilisateurs afin d'affecter une tâche en parallèle à plusieurs utilisateurs.

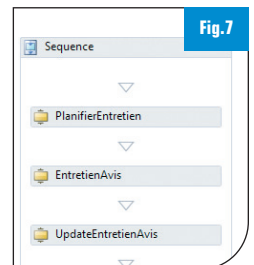
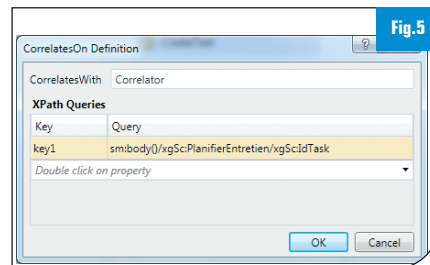
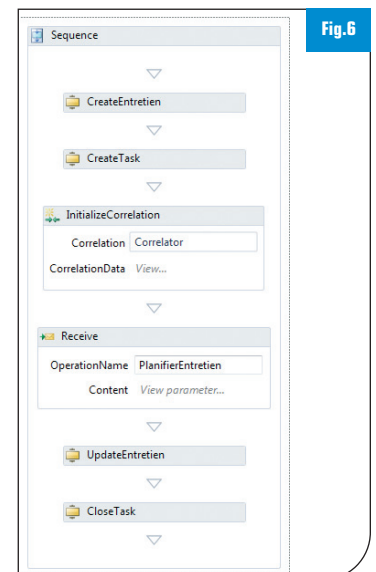
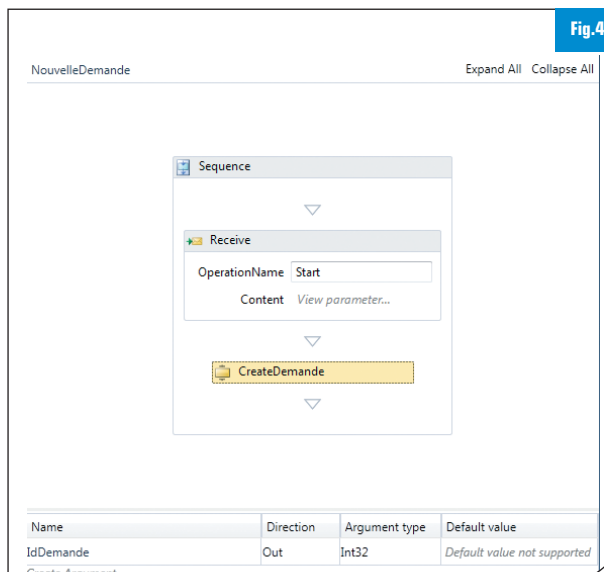
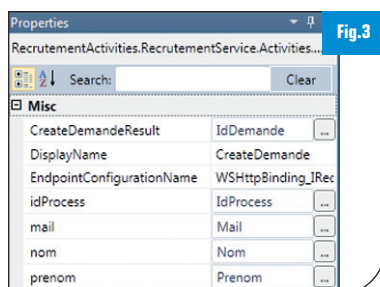
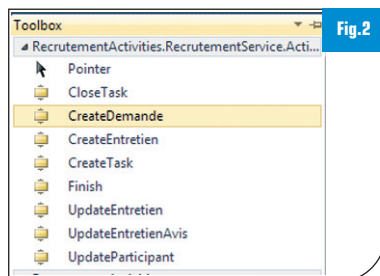
Maintenant que nous commençons à avoir une bibliothèque d'activités assez conséquente, nous allons pouvoir passer à l'étape suivante pour modifier le rendu graphique de nos activités !

Les designers d'activités

Le design d'une activité par défaut est très basique, juste une icône et un nom. Tous les paramètres doivent être définis à l'aide de la fenêtre « propriétés » de Visual Studio. Il est cependant possible de surcharger ce modèle par défaut, afin d'enrichir l'ergonomie et l'aspect visuel de nos activités.

Il suffit de créer un nouveau projet de type « Activity Designer Library ». Nous allons créer un designer pour une activité appelée « Entretien Autre » qui permet de définir un entretien avec un nombre illimité de participants. La liste des participants sera renseignée directement à l'aide de notre nouveau designer.

Nous restons en terrain connu puisque dorénavant tout l'aspect graphique de WorkflowFoundation est réalisé en WPF. Nous allons



gros plan \ modélisation

donc surcharger dans un premier temps la partie XAML afin de modifier complètement le rendu de notre activité une fois qu'elle sera « posée » sur notre surface d'édition du Workflow.

Le seul point à noter est que le contrôle WPF racine est un « WorkflowViewElement ». Le contrôle graphique de base de toute activité. Nous sommes libres de modifier le contenu à notre guise.

Aucune particularité. Juste une grille pour définir le layout et quelques ComboBox et boutons pour pouvoir ajouter ou supprimer des participants.

Regardons un peu du côté du code behind. Il y a une méthode qu'il est souvent intéressant d'implémenter. Il s'agit de la méthode « ActivityDesigner_Loaded ». Nous souhaitons complètement affranchir l'utilisateur final de la notion de variable ou d'argument. Nous allons donc paramétrer dynamiquement notre activité au moment où elle sera chargée dans le designer. Nous avons accès aux propriétés de notre activité par l'intermédiaire de l'objet « ModelItem ». Nous l'utilisons de cette façon :

```
private void ActivityDesigner_Loaded(object sender, System.Windows.RoutedEventArgs e)
{
    this.RestoreSelection();

    InArgument<int> idDemande = new InArgument<int>()
    {
        Expression =
        new Microsoft.VisualBasic.Activities.VisualBasicValue<int>
        {
            ExpressionText = « IdDemande »
        }
    };

    this.ModelItem.Properties[« IdDemande »].SetValue(idDemande);

    OutArgument<int> idDecision = new OutArgument<int>()
    {
        Expression =
        new Microsoft.VisualBasic.Activities.VisualBasicReference<int>
        {
            ExpressionText = « IdDecision »
        }
    };

    this.ModelItem.Properties[« IdDecision »].SetValue(idDecision);
}
```

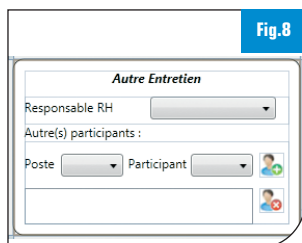


Fig.8

Les propriétés (ici les arguments d'entrées et de sorties) de l'activité sont en fait des expressions Visual Basic. Ici nous utilisons les classes « VisualBasicValue » et « VisualBasicReference » pour définir par code l'expression définie dans les deux arguments de notre activité (IdDemande et IdDecision). Notre designer est terminé ! [Fig.8].

Rehosting de designer

Nous allons maintenant développer l'outil permettant aux utilisateurs de modéliser leurs Workflows. Une sorte de Visual Studio Light qui contiendra :

- Une toolbox avec une librairie d'activités
- Un menu permettant de sauvegarder, charger leurs Workflows
- Une surface de design pour modéliser

Cette fois-ci nous allons partir d'un projet WPF standard :

```
<Window x:Class="Recrutement.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:workflow="clr-namespace:System.Activities.Presentation.Toolbox;assembly=System.Activities.Presentation"
        xmlns:local="clr-namespace:Recrutement"
        xmlns:sys="clr-namespace:System;assembly=mscorlib"
        Title="Recrutement" Height="768" Width="1024"
        Initialized="Window_Initialized" WindowStartupLocation="CenterScreen"
        Icon="/Recrutement;component/Icones/Personne01.ico" WindowState="Maximized">
    <Window.Resources>
        <sys:String x:Key="AssemblyName">System.Activities, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35</sys:String>
    </Window.Resources>
    <Grid Name="grdMain">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="2*" />
            <ColumnDefinition Width="8*" />
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto" />
            <RowDefinition Height="Auto" />
        </Grid.RowDefinitions>
        <MenuGrid.Column="0" Grid.Row="0" Grid.ColumnSpan="2" Height="25" HorizontalAlignment="Left" Name="mnuMain" VerticalAlignment="Top" Width="1024">
            <MenuItem Name="mnuXamlNew" Header="Nouveau Processus" Click="mnuXamlNew_Click" />
            <MenuItem Name="mnuXamlLoad" Header="Charger Processus" Click="mnuXamlLoad_Click" />
            <MenuItem Name="mnuXamlSave" Header="Sauvegarder Processus" Click="mnuXamlSave_Click" />
            <MenuItem Name="mnuXamlQuit" Header="Quitter" Click="mnuXamlQuit_Click" />
        </Menu>
        <BorderGrid.Column="0" Grid.Row="1" Name="brdWkfToolBox">
            <workflow:ToolboxControl>
                <workflow:ToolboxCategory CategoryName="Entretiens">
                    <workflow:ToolboxItemWrapper AssemblyName="RecrutementActivities" ToolName="RecrutementActivities.EntretienRH" DisplayName="Entretien RH" />
                    <workflow:ToolboxItemWrapper AssemblyName="RecrutementActivities" ToolName="RecrutementActivities.EntretienTechnique" DisplayName="Entretien Technique" />
                    <workflow:ToolboxItemWrapper AssemblyName="RecrutementActivities" ToolName="RecrutementActivities.EntretienAutre" DisplayName="Entretien Autre" />
                </workflow:ToolboxCategory>
            </workflow:ToolboxControl>
        </BorderGrid.Column>
    </Grid>
</Window>
```

```
<workflow:ToolboxItemWrapperAssemblyName="{StaticResourceAssemblyName}" ToolName="System.Activities.Statements.FlowDecision"
DisplayName="Decision"/>
<workflow:ToolboxItemWrapperAssemblyName="RecrutementActivities"
ToolName="RecrutementActivities.Fin" DisplayName="Fin"/>
</workflow:ToolboxCategory>
</workflow:ToolboxControl>
</Border>
<BorderGrid.Column="1" Grid.Row="1" Name="brdWkfDesigner"></Border>
</Grid>
</Window>
```

Il faut noter ici l'utilisation du contrôle « ToolboxControl » qui, comme son nom l'indique, va nous permettre d'afficher et de gérer la Toolbox de notre application. Il est nécessaire de lui indiquer les activités qui vont être incluses dans notre boîte à outil. Notez aussi la présence d'une bordure vide « brdWkfDesigner » qui va contenir notre surface de design. Nous devons la définir dans le code behind comme ceci :

```
this.m_oWflDesigner = new WorkflowDesigner();
this.brdWkfDesigner.Child = this.m_oWflDesigner.View;
```

Notre application est presque prête à être utilisée. Il ne nous manque que quelques détails. Il faut associer nos designers d'activités personnalisés de cette manière :

```
private void Window_Initialized(object sender, EventArgs e)
{
    // Référence des métadonnées du Designer Workflow dans WPF

    (new DesignerMetadata()).Register();

    // Surcharge du designer pour les customactivités XAML

    AttributeTableBuilder oAtb = new AttributeTableBuilder();
    oAtb.AddCustomAttributes(typeof(EntretienTechnique),
        new DesignerAttribute(typeof(EntretienTechDesigner)));
    oAtb.AddCustomAttributes(typeof(EntretienAutre),
        new DesignerAttribute(typeof(EntretienAutreDesigner)));
    oAtb.AddCustomAttributes(typeof(EntretienRH),
        new DesignerAttribute(typeof(EntretienRHDesigner)));
}
```

```
oAtb.AddCustomAttributes(typeof(Fin),
    new DesignerAttribute(typeof(FinDesigner)));
oAtb.AddCustomAttributes(typeof(NouvelleDemande),
    new DesignerAttribute(typeof(DebutDesigner)));
MetadataStore.AddAttributeTable(oAtb.CreateTable());
}
```

Ce code permet d'associer chaque Designer à son activité correspondante en utilisant les attributs de Metadata.

Un jeu de service lié au designer nous permet de manipuler l'aspect graphique de notre surface de modélisation. Nous n'allons autoriser que le Zoom pour que l'utilisateur ne puisse pas être tenté de modifier les variables de notre Workflow qui sont gérées dynamiquement par nos activités.

```
this.m_oWflDesigner.Context.Services.GetService<System.Activities.Presentation.View.DesignerView>().WorkflowShellBarItem
Visibility =
System.Activities.Presentation.View.ShellBarItemVisibility.Zoom;
```

Pour sauvegarder notre Workflow, rien de plus simple ! Il suffit d'appeler la méthode Save du designer. De même pour les charger, il suffit d'appeler la méthode Load :

```
this.m_oWflDesigner.Save(p_sFilename);

this.m_oWflDesigner.Load(p_sFilename);
```

[Fig.9]

Hébergement des Workflows

Dernière étape, il nous faut un service pour héberger nos Workflows. Chaque type de Workflow sera accessible par une url spécifique. Pour cela nous allons utiliser la classe WorkflowServiceHost qui va se charger d'héberger nos Workflows. Il nous faut également paramétrer la persistance de nos Workflows en base. Il suffit d'exécuter les scripts fournis dans le répertoire « C:\Windows\Microsoft.NET\Framework\v4.0.30319\SQL\en », nommés « SqlWorkflow InstanceStore-Logic.sql » et « SqlWorkflow InstanceStore Schema.sql », soit sur une base de données déjà existante ou sur une nouvelle. Ensuite nous allons parcourir notre base de données afin de récupérer l'ensemble des Workflows qui ont été créés par les utilisateurs et démarrer un nouveau WorkflowServiceHost comme ceci :

Dans cet exemple, les Hosts sont hébergés dans une application console. Il est tout à fait possible de les héberger dans un service Windows, directement dans IIS ou en utilisant Windows Server AppFabric. Nos Workflows sont prêts à être appelés ! Nos Workflows sont accessibles directement par WCF donc depuis n'importe quel type de client. Il suffira d'appeler l'opération Start pour démarrer un nouveau Workflow, requêter notre base de données pour avoir la liste de tâches correspondant à l'utilisateur connecté et appeler les opérations adéquates pour faire avancer nos Workflows.

Maintenant, il ne tient plus qu'à vous de développer vos propres activités, d'enrichir vos designers et de proposer la meilleure interface possible pour vos utilisateurs !

■ Julien Balouka

Technical Leader-Winwise

Julien.balouka@winwise.fr

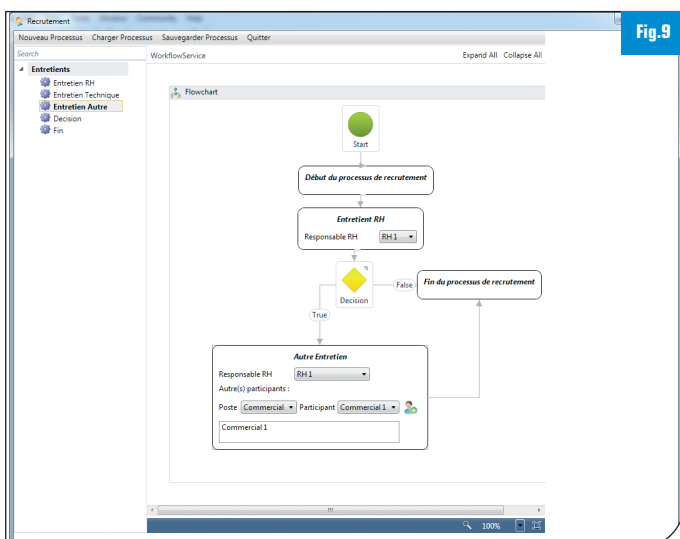


Fig.9



Les forks

nuisent-ils à l'open source ?

Dossier réalisé par François Tonic

Depuis quelques mois, le monde ouvert / libre est traversé par une « épidémie » de forks, ces dérivés d'un projet open source. Il ne s'agit pas d'une première pour MySQL, mais c'est une nouveauté en ce qui concerne OpenOffice et Mandriva.

Naturellement dans les gènes de l'open source, le fork d'un projet a toujours existé. Mais pour MySQL et OpenOffice, cela apparaît comme une réaction au rachat de Sun par Oracle et aux orientations, pas toujours limpides de l'éditeur face à l'open source. Le fork d'OpenOffice, LibreOffice, est clairement hostile et une partie non négligeable des développeurs, responsables et de la communauté sont partis vers LibreOffice qui est allé jusqu'à demander à Oracle, la cession pure et simple de la marque OpenOffice [sic]. Pour conforter le fork, une fondation a été créée à l'instar de Mozilla ou Apache pour assu-

rer l'indépendance et le financement du projet.

Cependant, le succès d'un fork n'est pas assuré. Par le passé, de multiples dérivés se sont créés mais rapidement se sont avérés être des échecs. Le défi du fork est de créer une communauté, car souvent, la communauté d'origine ne suit pas ou partiellement, sauf crise grave. Il faut attirer les développeurs, trouver des utilisateurs. Et des sociétés et éditeurs pouvant proposer des services, du support autour du fork.

Le cas Mandriva

Ensuite il y a différentes manières pour le projet initial de voir le fork. Soit il le prend mal et c'est la guerre, soit il fait bonne figure et tente de composer. Dans le cas de Mandriva, par exemple, Arnaud Laprèvote (CEO et directeur technique de Mandriva) considère que : « *Ce n'est pas le premier fork [de Mandriva]. Mandriva, c'est à la fois une entreprise et un projet.* ». Et à un moment donné, il faut faire des choix : soit on privilégie le projet, soit l'entreprise.

Et dans ce dernier cas, il faut trancher dans les stratégies à prendre, notamment sur la rentabilité et le choix des produits que l'éditeur open source veut vendre... Et le CEO de poursuivre : « ceux qui sont partis, considéraient qu'ils étaient Mandriva ». Il y a un facteur économique et un facteur communauté / développement. Pour Mandriva, le constat est simple : les solutions desktop ne se vendaient pas sur nos marchés. Des choix devaient être faits. C'est pour cela que Mandriva a voulu se concentrer, ici, sur la partie entreprise et serveur, même si le desktop continuera à être soutenu et développé (mais pas en France). Et pourtant, le CEO nous a précisé que des discussions et propositions avaient été faites aux développeurs qui sont finalement partis. Mais rien ne dit que le fork Mageia aura du succès. Il est bien trop tôt pour le savoir, de même que les conséquences sur Mandriva et la communauté.

Finalement, comme le dit Arnaud Laprèvote, « que le meilleur gagne ! ».



Fork ou pas fork ?

Le point de vue de l'AFUL

Laurent Séguin, vice-président de l'AFUL (association francophone des utilisateurs de logiciels libres) nous livre ses commentaires suite aux récents évènements par rapport à MySQL, OpenOffice, Mandriva, et le fork en général.

Qu'est-ce qu'un fork ? Comment le définir ? Son rôle et son rapport au « projet mère » et à la communauté.

Un fork est un programme développé à partir des sources d'un autre. Il existe différents types de forks, créés pour diverses raisons et qui répondent à des besoins différents. On peut citer par exemple :

- la continuation d'un programme abandonné qui permet de faire vivre un logiciel tout en s'affranchissant des soucis de marque du projet original (on peut donner en exemple Kompozer qui est un fork de NVU)
- une évolution d'un programme dans une direction différente (Firefox est originellement un fork de la partie navigateur de la Mozilla Suite avant d'en devenir un produit) ;
- une « libération » d'un programme sous licence propriétaire (le rachat des sources de Blender pour les mettre sous licence libre en est un exemple)
- une « propriété » d'un logiciel libre (Cedega est un fork propriétaire de Wine)
- une divergence dans la gouvernance du projet (X.org est un fork de XFree86 suite à un désaccord sur la licence du logiciel)

Les effets sur le projet mère peuvent n'avoir aucune incidence (par exemple FreeBSD est toujours là malgré les nombreux forks de ce système d'exploitation), peut le tuer (Sodipodi est totalement abandonné alors que Inkscape est très actif), ou même parfois permet aux deux projets de « se tirer la bourre » sur la qualité (exemple Frugalware Linux qui est un fork à suivre de Slackware Linux) ou n'avoir aucune incidence car le fork se meurt

De nombreux développeurs se sont essayés (par challenge personnel ou défi technique) à maintenir un fork d'une distribution majeure de GNU/Linux et ont arrêté au bout d'un moment. Cependant, et dans tous les cas, un fork affaiblit forcément le projet mère qui perd des contributeurs, qu'ils soient historiques ou potentiels. Bref, tout dépend de comment ceux qui maintiennent le projet-mère se comportent, de la volonté et de la résistance au temps de celui ou ceux qui « forkent », de l'adhésion des utilisateurs au fork, etc. Il est totalement impossible de prévoir à l'avance l'impact qu'aura le fork sur le projet initial ou sur la communauté.

Je tiens à préciser que le « forkage » n'est pas forcément une spécificité du logiciel libre. En effet, il existe bon nombre de forks de logiciels propriétaires souvent méconnus voire totalement inconnus (secret industriel des différentes parties) suite au rachat de l'entreprise éditrice par une autre ou carrément un achat du code source d'un logiciel pour le développer dans une autre direction.

Bien évidemment, une licence libre simplifie grandement les problèmes relatifs aux droits d'auteurs et, de par l'ouverture des codes sources, est visible de tous.

De nombreux entrepreneurs craignent le fork et hésitent à se lancer dans l'aventure du logiciel libre. Parfois c'est à raison (tout dépend du marché visé, de la concurrence, du business model, etc.), mais le plus souvent c'est à tort.

Dans le monde du libre, le fork est l'exception, pas la règle. A ceux qui ont ces craintes, je leurs dis « Lan-

cez-vous ! » Si vraiment vous craignez cette épée de Damoclès, voyez là comme une obligation de qualité, d'innovation et d'écoute des besoins des utilisateurs plus que comme une menace. Si vous faites le job et que vous jouez les règles du jeu, personne n'aura intérêt à forker !

Les forks de grands projets se multiplient depuis quelques mois : MySQL, OpenOffice, Mandriva, pour les principaux. Comment le comprendre ? Comment l'appréhender ?

Les motivations menant à un fork sont uniques. Chaque fork a son histoire (pas forcément réjouissante) mais certains sont nécessaires. Pour parler de ceux que vous citez voilà ce que j'en pense :

- Il y a de nombreux forks de MySQL. On verra s'ils tiennent la route et arrivent à prendre des parts de marché, il est trop tôt pour faire de réelles prévisions. Cependant, il faut voir que MySQL est critique pour bon nombre d'entreprises utilisatrices (notamment celles qui ont choisi la version commerciale et le support qui va avec) et ces entreprises craignent de perdre le support de la part d'Oracle et se voir un jour imposer le SGBD Oracle en lieu et place. Il y a donc un marché à prendre sur le support de MySQL et c'est ce qu'ont fait les fondateurs de SkySQL.

Je vois cela d'un bon œil, c'est très bien qu'il y ait un peu de concurrence, y compris sur le support bas niveau d'un même logiciel, cela stimule tout l'écosystème. Au passage, à part quelques cas très spécifiques, il n'est pas très compliqué de

migrer MySQL vers PostgreSQL ;-)
 - OpenOffice est probablement le fork qui fait parler le plus de lui et pour encore un bon moment. Cette suite bureautique est devenue tellement importante et tellement critique pour le monde du libre et les différents acteurs gravitant autour, qu'il était vraiment plus raisonnable de sortir de chez l'éditeur pour continuer le projet dans une optique de développement communautaire pur. Cela à l'avantage de rendre ce logiciel plus résilient aux stratégies industrielles des uns et des autres, tout en leur permettant de travailler ensemble sur un même projet (ce qui est plus problématique quand le projet est hébergé principalement par une entité commerciale) et donc d'assurer sa pérennité. Je souhaite donc une belle et longue vie à la Document Foundation et j'espère qu'elle saura réussir ce beau pari sur l'avenir et devenir aussi emblématique et indépendante financièrement que la Mozilla Foundation ou la Linux Foundation.

- Pour ce qui est de Mageia, fork de Mandriva (qui est, pour mémoire, un fork de Red Hat), le fork est toujours en cours de réalisation, c'est donc difficile d'en parler. Ce fork est principalement issu d'anciens employés remerciés par l'éditeur et de quelques contributeurs, suite à l'arrivée d'un nouvel investisseur qui a imposé une restructuration des dettes assortie d'une simplification de l'organisation. Cela rend les relations avec l'entité commerciale relativement difficiles. Du moins jusqu'à ce que l'émotion tombe, que l'état des lieux soit fait et que chacun définisse son périmètre d'évolution. Je pense que ce fork peut être bénéfique pour Mandriva et aux futurs utilisateurs de Mageia, tout comme Fedora est bénéfique à Red Hat. Il est à mon avis important pour Mandriva de s'appuyer enfin sur sa base d'utilisateurs.

Signe d'indépendance par rapport aux éditeurs ? Les éditeurs ont payé et contribué aux développements de ces projets.

Si les éditeurs ont certes payé et développé ces projets, il ne faut pas

oublier de valoriser l'implication des utilisateurs et des bénévoles autour du projet. OpenOffice par exemple est disponible dans plus de 40 langues (combien de langues pour Microsoft Office ? - 50 pour Office, ndr), quel coût cela a eu pour l'éditeur ? Quel est le coût des dictionnaires de correction orthographiques ? Quel est le coût du support et d'aide aux utilisateurs ? Quelle est la valorisation des remontées de bugs et de patch qui arrivent gratuitement ? Il faut bien comprendre qu'un logiciel libre n'appartient pas à son éditeur originel, il appartient à ses utilisateurs et contributeurs. Certains l'apprennent tardivement ou le redécouvrent...

En même temps un fork est un gage de succès du logiciel, cela veut dire que la base existante est bien faite et qu'elle vaut l'implication nécessaire de quelques personnes pour continuer de la maintenir et de la faire évoluer plutôt que de repartir à zéro.

Comment assurer le financement des projets ?

Ah l'éternelle question des modèles économiques et de la rémunération du code. Je pense qu'il y a moyen de faire de l'argent et de stabiliser le financement de bon nombre de logiciels libres. Cependant il faut que tout le monde joue le jeu pour pérenniser les projets. À chaque projet ses modèles économiques applicables. L'AFUL maintient une liste des différents modèles économiques liés au libre :

<http://aful.org/professionnels/modeles-economiques-logiciels-libres>

Cependant, vu de mon prisme, la stabilité financière d'un projet libre ne peut se faire qu'avec les acteurs qui font du business avec ou qui y ont intérêt. Typiquement pour OpenOffice, trop d'intégrateurs ou sociétés de service facturent du support technique niveau 3 sans reverser au projet ni code ni argent.

Espérons qu'avec LibreOffice et la Document Foundation ces intégrateurs reverseront leur bout de code, voire commanderont des prestations de développement à la fondation si elles n'en ont pas les compétences. C'est un modèle gagnant x 4 : le client

à une suite qui fonctionne conformément à ses besoins, l'intégrateur gagne en image de marque et en expertise (sans compter les rapports privilégiés avec la fondation), la fondation reçoit des contributions en nature ou financière, tous les autres utilisateurs bénéficient du code produit et payé par un autre. Les utilisateurs peuvent également mettre la main à la poche pour quelques euros de temps en temps afin de soutenir la structure à but non lucratif qui maintient le projet.

Avant d'utiliser un fork, que doit-on regarder ?

Cela dépend du fork et de l'utilisateur ;-) Dans tous les cas, il faut faire très attention à son patrimoine numérique et il faut s'assurer que ses données originelles soient toujours lisibles par le fork et que les données produites par le fork le soient également du projet mère. Bien évidemment, tout ce beau monde se doit d'utiliser des standards ouverts (c'est très rare que ce ne soit pas le cas avec un logiciel libre). Le plus important à mes yeux, avant même l'ouverture du code et les libertés accordées par la licence, ce sont les libertés, la pérennité et l'interopérabilité des données de l'utilisateur.

Du point de vue de l'utilisateur :

- si l'utilisateur est un utilisateur lambda d'une distribution GNU/Linux (coucou maman), qu'il fasse confiance aux choix effectués par les développeurs de la distribution lors des mises à jours. C'est très rare qu'ils soient totalement à côté de la plaque ;-)
- si l'utilisateur est un utilisateur lambda sur Windows ou MacOS, comme d'habitude, faites confiance à votre instinct et aux retours des utilisateurs plus chevronnés que vous associez à quelques saines lectures dans la presse spécialisée
- si l'utilisateur souhaite s'impliquer un peu dans le projet, qu'il choisisse celui qui va mieux prendre en compte ses remontées de bug
- si l'utilisateur est un core developer, bah... a-t-il vraiment besoin de mes conseils ? ;-)

NOUVELLE
VERSION

16

Elu «Langage le
plus productif du
marché»

VERSION
EXPRESS
GRATUITE

Téléchargez-la !

WINDEV

DÉVELOPPEZ
10 FOIS
PLUS VITE

996
NOUVEAUTÉS

WINDEV 16 est l'environnement de développement totalement intégré, qui couvre l'intégralité du cycle de développement.

Pour le découvrir, vous pouvez télécharger la version Express, demander le numéro spécial «01 Informatique» de 100 témoignages, ou encore mieux: demander le dossier complet gratuit (avec DVD).

Et vous aussi, développez 10 fois plus vite.

ACTUALITE WINDEV

- Opération «1 PC pour 1 Euro de plus»
- «Tour de France» des versions 16

Voir les annonces dans ce magazine, merci.

Votre code est unique: Windows, Internet, Java, .Net, PHP, J2EE, XML, Ajax, Webservice, Mac, Linux, Android, Windows Phone 7, SaaS,...



Fournisseur Officiel de la Préparation Olympique

www.pcsoft.fr



► Dossier gratuit 200 pages sur simple demande. Tél: 04.67.032.032 info@pcsoft.fr

MariaDB, SkySQL ou MySQL

Le choix entre l'original et le fork

Depuis quelques années, l'actualité de MariaDB est toujours très chargée. Son apparition, l'an dernier, sur la scène a été remarquée, et saluée par la communauté : on attendait plus de vitalité et d'innovations techniques. Le pari est tenu, et cette année, c'est SkySQL qui débarque. Comme le présente Kaj Arnö, son CEO, c'est la séparation de l'église et de l'état : SkySQL assure la commercialisation des services de pointe pour MariaDB, et Monty Program, assure les aspects techniques. Nos collaborateurs ont pu rencontrer Kaj Arnö de SkySQL et Rasmus Johansson de Monty Program pendant le Forum PHP 2010 organisé par l'AFUP (l'Association Française des Utilisateurs de PHP) pour faire le point.



Rasmus Johansson



Kaj Arnö

Historique

Avant de rentrer dans le vif du sujet, il est important d'effectuer un petit rappel concernant l'origine de MySQL. La base de données MySQL est définie comme un système de gestion de base de données (SGBD) sous licence propriétaire et libre. Elle est la base de données la plus répandue au niveau des serveurs d'hébergement Web.

Elle a été rachetée début 2008 par la société SUN, rachetée à son tour par la société Oracle en 2009 et validée début 2010 par la Commission Européenne.

Depuis que la société Oracle a effectué cette acquisition, l'écosystème autour de la base de données a été énormément perturbé avec l'apparition du fork MariaDB, suivi cette année d'une offre de services proposée par SkySQL.

A présent, la base de données MySQL se voit offrir 2 solutions :

- La première concerne l'offre classique existant depuis de nombreuses années avec quelques modifications de la société propriétaire.
- La deuxième est une offre alternative, c'est-à-dire des services presque identiques mais avec une solution libre.

Actuellement, les différentes offres que proposent ces 2 sociétés vont s'orienter autour de plusieurs axes :

- Les serveurs
- Le monitoring
- Un éditeur de requête au niveau software
- Une assistance technique

MariaDB, fork de MySQL

A l'origine, MariaDB est un fork de MySQL. Cette dernière, passée dans le giron d'Oracle, a un avenir incertain. Coincée entre la compétition interne avec les produits phare d'Oracle, et le passage à des pratiques commerciales adaptées au monde propriétaire, elle accumule les contraintes. La licence GPL est toujours disponible aux côtés des licences commerciales : elle a permis de démarrer un nouveau projet à partir du code de la fameuse base de données.

Le fork complet du projet permet de garantir à la communauté la sauvegarde du projet. Les innovations pour MySQL passent actuellement par les moteurs de tables, qui sont nombreux et dynamiques (Open Query Graph, Xpath, Aria, sphinxse, pbxt...), mais leur support par le moteur SQL est une condition de survie à long terme.

L'autre condition du fork est le support par la communauté. Le transfert de cette dernière de MySQL vers MariaDB est surveillé de près et même fortement incité : MariaDB se présente actuellement comme un remplacement standard de MySQL. La communauté technique sera séduite par les nouveaux moteurs de tables : qui remplacent d'anciens de MySQL, comme XtraDB qui remplace

InnoDB; des améliorations des standards, comme les moteurs HEAP ou MyISAM qui est relevé par Aria; des moteurs qui sont disponibles sur les deux bases, comme PBXT, ou encore de pures nouveautés, comme OQGraph, un moteur de graphe, issu de la vague NoSQL. Du côté des services, c'est SkySQL qui prend le relais, et propose les services de monitoring de serveurs, de conseils et formations. On est donc dans un environnement connu et classique.



La différence MariaDB

MariaDB entend proposer une offre de remplacement à MySQL. Pour se différencier, la nouvelle base a plusieurs atouts dans sa manche. Le premier, et non des moindres, est l'avantage technique. Le cycle de développement de MariaDB est beaucoup plus court que celui affiché par Oracle/MySQL. Aucun des deux n'est régulier, mais on devrait voir MariaDB sur un cycle rapide à 6 mois, tandis

qu'Oracle/MySQL proposera la stabilité avec un cycle stable de quelques années. Il y aura donc plus de nouveautés côté MariaDB, et plus de corrections tant attendues de la part de la communauté.

La différence de cette implication se traduit aussi dans les risques : le partage se fait entre un risque technique avec MariaDB, et un risque politique pour MySQL. Dans le premier cas, l'ouverture de la base permet d'explorer de nouvelles possibilités, aussi loin que les capacités techniques de chacun le portent. La stabilité du code MySQL élimine ce risque en le transformant en un risque politique : MySQL est désormais à la merci d'un changement de cap du vaisseau amiral. Ce type de situation va donc automatiquement séparer les communautés MySQL et MariaDB. Les premiers seront ceux qui peuvent s'offrir les licences, ou souhaitent un cycle de développement de même vitesse que le leur, tels les éditeurs et les institutions. Ceux qui sont plus rapides ou souhaitent se porter sur des innovations opteront naturellement pour MariaDB.

Communauté

Les 2 projets Oracle/MySQL et SkySQL/MariaDB ont les yeux tournés vers la communauté car c'est un des critères importants qui va rendre populaire une base de données. Depuis les derniers rebondissements et dernières annonces effectuées par la société Oracle, une vague de contributeurs, autour de la base de données MySQL se sont déportés vers la solution libre.

Le résultat obtenu, crée un doute sur le choix de la sélection auprès des utilisateurs finaux car même si l'assistance technique va jouer en faveur de l'un ou de l'autre, il est certain que l'aspect commercial sera aussi un des points importants.

Or, pour mesurer le succès d'un produit, les indicateurs sont toujours difficiles à mettre en place. Actuellement l'analyse de popularité se portera sur



le nombre de téléchargements, mais aussi auprès des réseaux sociaux et des communautés (ou associations) présentes dans les différents pays (Ndlr : En France par l'intermédiaire du LeMug).

Par ailleurs, les communautés des différents langages du marché ont aussi leur rôle à jouer sur internet. Le langage PHP est celui qui est le plus impliqué, mais les autres langages ont commencé aussi à suivre le projet MariaDB pour proposer les mêmes portabilités que MySQL.

Toutefois, les utilisateurs peuvent être intéressés de s'orienter vers l'entrepôt alternatif (exemple : NoSQL) pour garder l'aspect libre et la facilité de mise en place. Cependant d'après Kaj et Rasmus : « *il ne s'agit pas de bases de données relationnelles, mais plutôt d'un complément* », c'est pourquoi il est important de se faire conseiller par les professionnels avant de vouloir s'éloigner de MySQL ou de MariaDB.

Comment le fork pourra-t-il survivre ?

A l'heure actuelle, la différence entre l'original et le projet forké est encore minime. MariaDB met un point d'honneur à inclure les modifications publiées par Oracle/MySQL, afin de conserver la compatibilité. Ce dernier n'a pas encore profité de son statut d'éditeur propriétaire, pour diffuser des fonctionnalités MySQL spécifiques à la version commerciale, et publiées sous licence propriétaire : il s'agirait ainsi de séparer nettement les deux projets en scindant de manière définitive les fonctionnalités disponibles. Pour le moment, MariaDB a l'avantage technique, en suivant son rival, et en incluant des innovations Open Source : il sera diffi-

cile de retourner à MySQL quand on a goûté aux joies des nouvelles tables.

Pour assurer son avenir, il faudra que MariaDB se fasse un prénom. Par exemple, en apparaissant dans tous les produits où MySQL existe, et sous son nom. Par exemple, le Zend Framework ou Drupal, disposent d'un support spécifique à MariaDB, et on en attend d'autres dans les semaines à venir. Mais il faudra probablement attendre une killer application, tel phpMyAdmin, pour faire la différence : phpMariaAdmin pourrait-il prendre la place de l'application PHP la plus populaire de tous les temps ?



Enfin, et inévitablement, les deux projets prendront des directions différentes, jusqu'à être deux projets distincts. Cela prendra quelques années : nous n'avons encore rien vu !

Conclusion

Comme nous avons pu le constater dans cet article, le choix entre la base de données propriétaire et le fork montre très peu de différences. Les prochaines versions des 2 bases de données, pourraient modifier l'analyse d'aujourd'hui. Cependant les moyens mis en place par le fork montre qu'il est possible de garder la configuration actuelle tout en bénéficiant des nouvelles fonctionnalités.

■ Christophe Villeneuve

Consultant pour Alter Way solutions, auteur du livre « PHP & MySQL-MYSQLi-PDO, Construisez votre application », aux Éditions ENI. Rédacteur pour nexen.net, Trésorier AFUP et membre du LeMug.fr, PHPTV, PHPteam...

■ Damien Seguy

Dirige le pôle d'expertise en logiciels libres du groupe Alter Way. Damien est l'auteur de plusieurs livres, le père des elePHPants, co-auteur des certifications Zend et co-fondateur de l'AFUP, de PHP Québec.

Mageia, un nouveau challenge pour Mandriva

Depuis plusieurs mois, la communauté Mandriva est secouée par les rumeurs, les remaniements. Après les éventuels rachats, un fork, Mageia, a été lancé. Romain d'Alverny, de Mageia, fait le point pour nous.

Pouvez-vous nous rappeler le contexte de ce fork et la situation de Mandriva à cet instant ?

Le 2 septembre de cette année, le Tribunal de Commerce de Paris a mis la société Edge-IT (filiale du groupe Mandriva) en liquidation judiciaire avec cessation d'activité, suite à une déclaration de cessation des paiements par sa direction ; entraînant par là le licenciement du personnel. La situation était déjà tendue depuis de nombreux mois et la plupart des employés étaient inquiets pour le projet Mandriva et pour un esprit et un mode de management qu'il devenait de plus en plus intenable à défendre en interne. Une grande partie des salariés d'Edge-IT étaient des ingénieurs travaillant directement à la direction, la réalisation et au support de Mandriva Linux. Attachés à cette plateforme technologique, à ce que l'on peut faire avec, et considérant que le groupe Mandriva n'était plus forcément, à nos yeux, le meilleur endroit

pour la faire s'épanouir, nous (ex-salariés, contributeurs) avons considéré que c'était le bon moment pour lui donner une nouvelle perspective, dans un autre lieu, une autre forme, en particulier en donnant totalement les clés du projet à la communauté. C'était aussi le moment où diverses rencontres, discussions et réflexions personnelles et collectives, entamées plusieurs mois auparavant entre employés, contributeurs et observateurs se sont assez naturellement emboîtées. Le projet Mageia est né de cette convergence d'événements.

Mandriva parle de deux entités : l'entreprise et le projet communautaire. Mageia est-il une réaction à la nouvelle stratégie plus orientée entreprise, serveur en France et en Europe et non plus axée sur le Linux Desktop ?

Cela fait plusieurs années que la stratégie de Mandriva n'est pas axée exclusivement sur le poste de travail,

loin s'en faut : Corporate Server existe depuis 2001 et a connu 4 versions majeures par la suite en 2002, 2004, 2007 et 2009 (MES5) ; le Multi-Network Firewall a connu deux versions majeures en 2002 et 2004 ; plus récemment, Pulse 2 (suite de Pulse démarré en 2005), Directory Server (suite de Linbox Directory Server) et LRS. De même, Mandriva s'est intéressé aux offres OEM, à la formation, aux services et au support à destination des entreprises depuis 2002. Néanmoins, nous avons constitué Mageia sans demander l'avis de Mandriva. C'est plutôt, comme déjà dit, l'aboutissement de plusieurs éléments : plusieurs personnes clés autour de la distribution Mandriva Linux ont été libérées de

MANDRIVA : Focus sur le Serveur et sur les pays en développement

On sait que Linux ne s'est pas imposé dans le desktop, mais a gagné de fortes parts de marché dans le serveur. Ce marché d'entreprise, dominé par Red Hat, recèle de forts potentiels commerciaux en licences et surtout en services. Mandriva veut donc se concentrer sur ce marché en Europe, en apportant des arguments de facilité de mise en œuvre et d'exploitation.

Les pays en développement, en revanche, découvrent Linux pour les postes de travail. Le marché brésilien présente une particularité : de très fortes ventes de Linux auprès du grand public. Mandriva distribue au Brésil entre 500 000 et un million de boîtes par an, se réjouit Arnaud Laprêvote, Directeur Général. En plus du grand public, Mandriva

découvre un potentiel important dans l'Éducation nationale et a signé avec ce pays un marché de 600 000 postes sur 3 ans, dans les écoles. Le fait que le nouvel actionnaire russe ait également acquis Pingwin Software qui ferait le support pour 50 000 écoles russes,

“ **Mandriva est en train de construire un axe étonnant : France-Brésil-Russie.** ”

renforce la compétence du nouveau groupe Mandriva dans ce secteur. « Mandriva est en train de construire un axe étonnant : France-Brésil-Russie. » remarque A. Laprêvote.

Pour les pays en développement, Mandriva veut jouer aussi l'argument de l'indépendance par rapport aux éditeurs majeurs, tous anglo-saxons. Après la Russie et le Brésil, le monde en développement est vaste...

■ J.K.

Mandriva dévoile sa roadmap 2010-2011

La crise terminée, l'éditeur a dévoilé mi-novembre, sa roadmap 2010-2011. Tout d'abord, la cadence annuelle des mises à jour va se ralentir pour ne proposer qu'une version majeure par an, au lieu de deux. Par contre, développeurs et utilisateurs auront droit à deux versions fin décembre et courant janvier 2011 :

- Mandriva 2010.2 sortant le 22 décembre. Cette version mettra à jour les packages logiciels, le kernel, intégrera les derniers patches de sécurité.

- Mandriva 2011 : une version alpha sera disponible fin janvier 2011. Pour cette version nous ne connaissons pas encore les détails techniques. La version finale est prévue pour fin mai.

Pour se tenir au courant : <http://blog.mandriva.com/fr/>

leurs obligations, l'absence de visibilité à long terme sur la pérennité du projet, le besoin exprimé et ressenti depuis de nombreuses années, pour le projet, dans l'entreprise et la communauté, d'une structure et d'une gouvernance économiquement indépendantes et d'un souffle nouveau et enfin l'envie de faire autre chose que "juste une distribution Linux".

N'avez-vous pas peur d'une fragmentation trop grande de la communauté ? Comment a-t-elle réagi ?

Une fragmentation trop grande ? Nous ne pensons pas. Ce qui est essentiel, c'est de réunir des personnes capables de collaborer dans un climat de grande confiance, portées par un projet commun assumé, de bien le réaliser et de le savourer. L'annonce du fork a été globalement bien accueillie, bien au-delà de nos attentes ; que ce soient des encouragements, des félicitations, des ralliements (d'ex-employés, de contributeurs, d'utilisateurs, ndlr) ou les dons à l'association. Depuis un mois, environ 400 personnes se sont inscrites comme contributeurs au projet sur notre wiki ; en pondérant et en considérant l'activité actuelle sur nos groupes de travail, on peut considérer qu'il y a une bonne quarantaine de personnes très actives à ce jour pour poser les bases et coordonner.

Mageia est supporté et développé par les développeurs qui étaient au coeur de Mandriva. Quelles sont vos ambitions ? Vos désirs ? Quelle est votre roadmap ?

Nous voulons produire une plateforme technologique de référence et pour ce faire, fédérer et coordonner différentes communautés d'individus, de sociétés, d'associations. Nous avons défini la gouvernance du projet de telle manière que celui-ci ne dépende que de sa communauté, exclusivement et sans ingérence : plusieurs équipes de contributeurs/acteurs collaborent (développeurs, packageurs, testeurs, traducteurs, graphistes, marketing, web, communication, ndlr) et réalisent les différents projets ; faire partie d'une équipe suppose d'être coopté par les membres existants, un

conseil réunissant les représentants est élu tous les ans (par équipe) ; ce conseil dirige et coordonne la réalisation de la distribution, un bureau élu par tiers chaque année, parmi les anciens membres du conseil, dirige et représente l'association et le projet ; et tranche en cas de besoin les décisions du conseil. Ne pourront donc parvenir au conseil et au bureau (et donc à la direction du projet) que des personnes qui auront été reconnues par leurs pairs pour leur contribution et leurs qualités. Et non des personnes placées là par un jeu économique ou relationnel. Ce qui fera l'intérêt, c'est le succès et le climat de la collaboration au sein du projet. Notre distribution Linux en sera l'exemple plus abouti, progressivement, mais nous souhaitons aller plus loin, explorer d'autres réalisations complémentaires. Il ne s'agira plus de faire "juste une distribution Linux", mais de mieux comprendre et s'ouvrir à ce qui entoure une personne dans son environnement quotidien où qu'il soit, quoi qu'il fasse - et de mettre en commun ces connaissances et les technologies, concepts qui peuvent en découler, au sein du projet Mageia. Charge à chacun des participants d'y puiser par la suite. L'informatique a des implications grandissantes dans notre vie de tous les jours ; et depuis plusieurs années, le libre s'étend aussi à d'autres domaines que la seule informatique. Il est donc plus que pertinent de s'intéresser à la convergence de tout cela et de réfléchir, d'offrir à chacun les moyens de comprendre et garder le contrôle sur

sa propre existence, sa société, ses productions, ses outils, son environnement. Concrètement, nous comptons publier une version stable de notre distribution Mageia en avril 2011, avec des releases techniques de test intermédiaires, dont la première en décembre. D'ici là, la route est ambitieuse, tant sur le plan technique qu'humain. Nous organiserons notre première assemblée générale lors du FOSDEM à Bruxelles en février 2011, et serons présents à Solutions Linux à Paris en mai, au LinuxTAG à Berlin en juin, aux RMLL à Strasbourg, au Chaos Communication Camp 2011 près de Berlin en août et probablement à d'autres événements.

Si LibreOffice apparaît comme en concurrence frontale avec OpenOffice, comment se positionne Mageia par rapport à Mandriva ?

MandrakeSoft (puis Mandriva) était le lieu où sont nées une partie de la technologie et de la philosophie dont dérive Mageia ; et où plusieurs personnes de l'équipe fondatrice de Mageia ont officié pendant plusieurs années. C'est tout. Mageia n'est en aucune façon lié ou subordonné à la société Mandriva (ni à aucune autre société), ni financièrement ni par organisation.

Y aura-t-il un enrichissement mutuel notamment sur les développements ?

C'est ouvert et toujours souhaitable évidemment. Libre à des salariés de la société de venir participer au projet. Cette invitation est valable pour toutes les personnes et entreprises intéressées par le projet. ■

Jean-Noel de Galzain : « pourquoi tirer sur l'ambulance » ?

Jean-Noel de Galzain, dirigeant de Wallix a renoncé à son rôle d'administrateur de Mandriva : « nous avons sauvé l'entreprise de la faillite, mais les investisseurs ne nous ont pas suivi davantage ». Il est convaincu qu'avec son nouvel actionnaire russe et un gros contrat signé avec l'Education Nationale au Brésil, Mandriva devrait trouver dans les pays émergents un important potentiel de développement.

Avec le recul, et un peu amer, il estime : « l'open source est un sujet de plus en plus sociétal. Il n'y a pas de business model du logiciel libre. Il s'est structuré uniquement en offrant un espace de liberté aux développeurs. » L'ancien administrateur est sévère au vu de la home page de Mageia : « Quel est leur but véritable, à part taper sur Mandriva ? Leur démarche paraît surtout inspirée par la rancune, donc par le passé ! Les anciens employés ont un devoir de réserve et de confidentialité. Ils devraient arrêter de tirer sur l'ambulance, ce n'est pas éthique, ce n'est pas acceptable ! » ■ Jean Kaminsky

Etes-vous un développeur durable ?

Créez des logiciels “verts”

Une forte tendance actuelle concerne le GreenIT, une informatique plus écologique, un traitement des déchets de hautes technologies, une économie des performances énergétiques. En réalité, le GreenIT ou informatique verte recouvre une réalité très variée.

Les constructeurs ont été parmi les premiers acteurs à travailler sur l'optimisation énergétique des machines : comment mieux utiliser l'énergie consommée tout en consommant moins. Un énorme effort est fait sur les serveurs, les datacenters. S'il faut travailler au niveau matériel, il manque tout de même un élément vital : le logiciel. Et derrière le logiciel, il y a le développeur.

Depuis 20 ans, la course aux fonctionnalités, a considérablement alourdi le logiciel, pensez donc, plusieurs Go pour installer MS Office, plus de 7 Go pour Visual Studio 2001 Ultimate, etc. Le moindre logiciel prend tout de suite 300, 500 Mo. Cette inflation doit s'arrêter. Car la gourmandise des logiciels en puissance machine (CPU, GPU, RAM, etc.) n'est plus tolérable.

Que faire alors de l'équation GreenIT et développeur ? « Le développeur n'est pas un axe prioritaire des entreprises. Elles mettent l'accent plutôt

sur les performances des Datacenters, le travail à distance. Le GreenIT est davantage lié à l'organisation. Nous avons une inflation de puissance des logiciels. C'est là un très gros problème. Cependant, Windows 7 consomme moins de ressources que Vista. C'est un bon signe »,

commente Thomas de Lacharrière (responsable de l'offre Greentech chez Devoteam Consulting). Les éditeurs prennent peu à peu conscience du problème et cherchent à optimiser, à réduire la consommation en puissance matérielle et en énergie. C'est là que le développeur joue un rôle essentiel !

Dans ce dossier, nous allons voir comment le développeur peut appréhender le GreenIT et une programmation « éco-responsable » et l'impact que cela aura sur le code, les technologies. Cependant, nous n'en sommes qu'au début de la réflexion, un énorme travail reste à faire.

■ François Tonic

Le GreenIT, moteur de l'entreprise

Dans son étude GreenIT publiée en novembre 2010, Devoteam tente de comprendre les tendances de l'informatique verte au cœur des entreprises : où elle se situe, quelles sont les mesures prises, voire, les freins.

Les technologies au service du développement durable

La diminution des impacts de l'informatique sur l'environnement est le volet le plus largement visible du Green IT. Cependant, un second volet majeur est la contribution des technologies à la mise en place d'une économie plus responsable. Les directions informatiques doivent être parties prenantes de la politique de développement durable en fournissant des outils aux fonctions support et aux directions métier.

Les chiffres montrent un e-commerce largement répandu, qui illustre ce rôle d'innovation. Les stratégies de développement durable des entreprises, quant à elles, tirent de plus en plus parti d'outils de gestion du carbone et de l'énergie. Enfin, la création de produits plus respectueux de l'environnement est facilitée par les logiciels d'analyse de cycle de vie et d'éco-conception.

Véhicule électrique, smart-meters, géolocalisation, bâtiments intelligents ou dématérialisation, l'actualité regorge de projets innovants vers une économie plus verte. On estime que les technologies de l'information peuvent engendrer plus d'économies de CO₂ que leurs propres émissions. Il est temps pour les directions informatiques de devenir force de proposition afin de développer de nouvelles oppor-

tunités écologiques et économiques.

Les chiffres clés à retenir sont :

- 45% des organisations utilisent des catalogues électroniques ou des outils d'e-commerce.
- 24% des organisations ont déployé des logiciels de gestion du carbone et de l'énergie.
- 21% des organisations travaillent avec des logiciels d'éco-conception et d'analyse du cycle de vie.

Les raisons d'initier une démarche GreenIT

Dans la grande majorité des réponses faites à l'enquête, la démarche GreenIT est vue comme un enjeu important pour le développement durable et un moyen de répondre aux attentes des clients. Le message est optimiste, puisque seulement 14% des répondants qualifient le Green IT d'effet de mode. Dans l'ensemble, ces résultats confirment notre perception du marché, avec une nuance en ce qui concerne les enjeux d'image et de communication qui semblent quelque peu sous-évalués [Fig.A].

Le Datacenter : un nid d'économie et d'optimisation

L'énergie est aujourd'hui au cœur de la gestion des datacenters. Elle concentre les principaux enjeux de disponibilité des systèmes, d'émission de gaz à effet de serre et de maîtrise



« Contribution au réchauffement climatique, pollution des sols et des eaux, exploitation minière dans des sites sensibles, les technologies de l'information ont leur part de responsabilité dans la fragilisation des écosystèmes et nous devons aujourd'hui contrôler et réduire cet impact. Pourtant, on

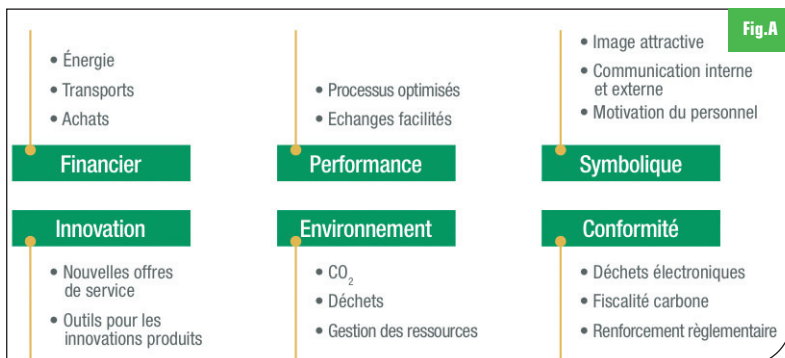
ne peut pas se limiter à ce constat. De l'autre côté de l'échiquier, les techniques d'imagerie satellite, de simulation, de géolocalisation et bien d'autres sont en train de révolutionner notre capacité à comprendre et à protéger notre environnement. Pour définir notre feuille de route, gardons à l'esprit ces deux visions antagonistes : si les technologies sont une part du problème, elles sont aussi au cœur de la solution ! »

Thomas de Lacharrière, Responsable de l'offre GreenTech, Devoteam Consulting

« si les technologies sont une part du problème, elles sont aussi au cœur de la solution ! »

des coûts. Dans le secteur tertiaire, elle peut représenter jusqu'à 25% des émissions de CO₂ d'une entreprise. Pour preuve, 65% des entreprises connaissent la facture électrique de leurs datacenters contre moins de 40% pour les postes de travail et les infrastructures réseaux. Souvent noyées dans une facture électrique globale, ces deux dernières sont plus complexes à suivre. Les bonnes pratiques les plus répandues ne sont généralement pas impulsées uniquement par une politique Green IT. L'agilité, la résilience, la maîtrise des coûts sont des motivations tout aussi importantes que les économies d'énergie dans un projet de rationalisation ou de virtualisation. Toutefois, l'émergence d'outils de supervision énergétique montre une volonté d'agir dans la durée. Ces outils procurent des fonctions de suivi et d'alerte, éléments importants d'un projet de maîtrise énergétique.

Retrouvez les résultats complets de cette étude sur : www.devoteam.fr



Le développeur est la clé du Green IT

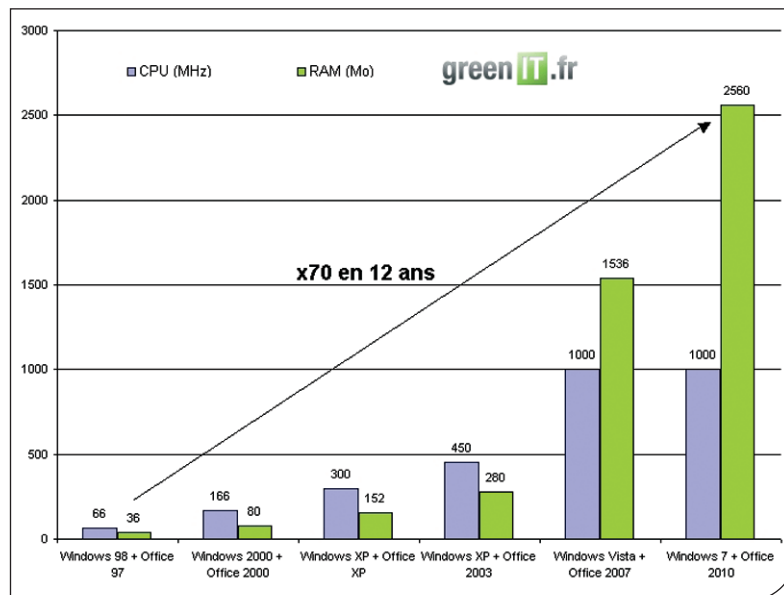
En trouvant un juste compromis entre une bonne productivité et le respect de contraintes en ressources matérielles, il peut contribuer à réduire l'empreinte écologique de l'informatique. Démonstration.

La fabrication et la fin de vie du matériel informatique concentrent les principaux impacts environnementaux : épuisement des ressources non renouvelables, émissions de CO₂, et pollutions chimiques. Pour réduire ces impacts, il faut allonger la durée de vie du matériel. Or, la durée d'utilisation d'un ordinateur a été divisée par 4 en 20 ans. Entre 1985 et 2007, elle est passée de 10,7 ans à 2,5 ans.

La couche logicielle joue un rôle prépondérant dans l'accélération continue du rythme de renouvellement du matériel informatique. En 12 ans, entre 1998 et 2010, la mémoire vive nécessaire pour écrire un texte a été multipliée par 70, passant de 36 Mo sous Windows 98 et Office 97 à 2.560 Mo sous Windows 7 et Office 2010. Soit un doublement des ressources matérielles nécessaires à l'exécution des nouvelles versions de logiciels tous les 2 ans. Ce constat n'est pas propre aux logiciels de Microsoft. Il se vérifie chez

presque tous les éditeurs. Imaginez que votre voiture nécessite aujourd'hui un moteur 70 fois plus puissant qu'il y a 12 ans pour parcourir le même nombre de kilomètres, à la même vitesse. Impensable. C'est pourtant bien ce qui se passe dans l'informatique.

L'augmentation continue des ressources matérielles nécessaires (processeurs, mémoire vive, etc.) n'est pas directement liée à l'ajout de nouvelles fonctionnalités, mais plutôt à l'empilement d'un nombre croissant de frameworks et de composants non optimisés les uns par dessus les autres. Les premiers développeurs ne pouvaient pas se permettre un tel



comportement car ils étaient contraints par la faible puissance des machines : difficile d'empiler les frameworks lorsque l'on dispose en tout et pour tout de 64 Ko de mémoire vive !

Le développeur dans la mine

Le développeur moderne est un acteur clé du Green IT. Il est le seul à pouvoir infléchir la tendance actuelle en s'appuyant sur des « green » design patterns et des bonnes pratiques lorsqu'il conçoit son logiciel. L'idée n'est pas de recommencer à coder un logiciel entier en assembleur « from scratch », mais plutôt de trouver le meilleur compromis possible entre la productivité apportée par l'assemblage de composants et de frameworks, et une limite raisonnable en terme de consommation de ressources matérielles (processeur, mémoire vive, etc.).

Facebook a récemment démontré, avec son projet HipHop for PHP, que ce compromis est possible, économi-

quement très intéressant, et viable dans le temps. HipHop for PHP transforme du code PHP en code C++ qui est ensuite compilé à l'aide de g++. 90% des pages de Facebook reposent aujourd'hui sur HipHop. Le gain en production est colossal puisque l'équipe technique de Facebook a divisé par 2 l'empreinte en ressources matérielles : il faut aujourd'hui 2 fois moins de serveurs pour faire fonctionner le réseau social. Les gains économiques et pour l'environnement sont évidents. Et ces gains n'ont pas été obtenus au détriment de la souplesse et de la rapidité de développement puisque les développeurs de Facebook continuent à coder en PHP.



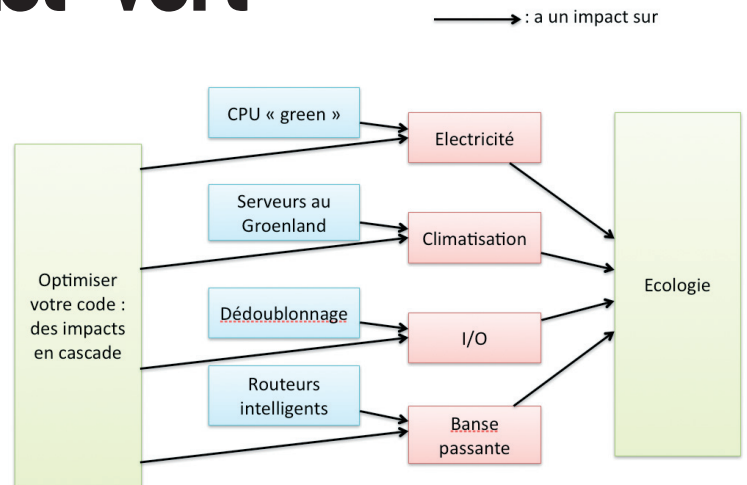
■ Frédéric Bordage

Après avoir été développeur et architecte logiciel, Frédéric est aujourd'hui un des experts reconnus du Green IT en France. Co-auteur de nombreux ouvrages et guides de bonnes pratiques sur le sujet, il anime depuis 4 ans la communauté de référence, GreenIT.fr, qui fédère chaque mois les 35 000 acteurs français du Green IT.



Le Lean et le logiciel "vert"

Les écologistes de longue date savent que le produit le moins polluant est celui qui n'a pas été produit. Avant de parler de facilité de désassemblage, de recyclage ou de dépollution, avant même de parler de matières premières, de ressources énergétiques, d'impact du transport, le mieux est tout simplement de ne pas construire le produit en dehors d'un besoin justifiable.



Le mouvement Lean, qui s'inscrit dans les architectures agiles, insiste sur l'importance de réduire au minimum les fonctionnalités produites. On a d'ailleurs vu des conférences dédiées spécifiquement à l'élimination des gaspillages lors de l'Agile Tour. Cette approche permet de réduire la sur-architecture, ennemi numéro un de la maintenabilité et de la performance des logiciels. Elle aide également à une meilleure satisfaction du client, en se concentrant sur les aspects qui lui apporteront immédiatement le plus de valeur ajoutée. Elle contribue au final à une informatique plus verte, en restreignant les processus à ceux strictement nécessaires.

Simplifier le code

Le parallèle entre ressources naturelles et informatiques est frappant. Aux premiers pas de l'informatique, ces dernières étaient rares, et les informaticiens habitués à composer avec cette rareté, réalisaient des prouesses sur quelques Ko. Puis, comme lors des Trente Glorieuses pour la consommation de masse, nous avons vécu dans un monde informatique basé sur l'abondance. Quelle que soit la lourdeur d'un calcul, le prochain CPU le rendait suffisamment rapide. Quelle que soit la taille d'une information, le nouveau réseau accélérât son transfert. Aujourd'hui, au même titre que nous cherchons à

réduire notre impact sur l'écosystème, il est important de réduire notre consommation de ressources informatiques.

Si on souhaite intervenir le plus en amont possible, et ainsi avoir le maximum d'impact, sur quoi travailler ? Pour une suite d'instructions données, mieux vaut-il acheter des processeurs « verts », déporter les calculs dans le Cloud, ou vers une salle blanche refroidie naturellement au Groenland ? Pas du tout : le plus efficace est de tout simplement ne jamais exécuter ces instructions. Et pour cela, optimiser le code source de façon à ce qu'il ne réalise que les instructions absolument nécessaires.

Simulation de l'économie liée à une optimisation logicielle

Vous trouverez facilement sur internet des benchmarks de processeurs avec les courbes de consommation en fonction de la charge. Elles montrent des écarts de 20 à 50% en fonction des modèles. Prenons l'exemple d'une ferme de huit serveurs pour laquelle une optimisation logicielle fait tomber de 80% à 60% l'utilisation des CPU. L'enveloppe thermique perd environ 20%, soit une consommation qui tombe de 85 à 65 Watts. 20 Watts économisés pour chacun des huit serveurs, soit 160 Watts à multiplier par les 365 x 24 heures d'une année. En ne travaillant

que sur votre code, vous économisez 125 € d'électricité (0,09 € le kWh). Pas énorme... Mais là où ça devient intéressant, c'est que vous allez désormais pouvoir consolider vos serveurs en une ferme de six machines seulement. Et économiser alors les deux fois 2 000 € d'achat des machines, et la totalité de leur consommation, bref on parle cette fois d'économiser les deux fois 450 Watts de consommation, soit au total 710 €, et donc une économie totale de 4 710 €.

Bien sûr, ce calcul appellera contestation, rectification, ajustement des chiffres, etc. Mais il en ressort tout de même que les économies sur un algorithme sont bien de l'ordre de grandeur de la centaine d'euros sur un an, et pas de l'ordre de la dizaine ou même des unités, comme on aurait pu le penser sur une première estimation.

L'optimisation en pratique

Mais ce genre d'algorithme peu performant est-il légion ? Nos applications ne sont-elles pas soigneusement optimisées ?

Sérieusement, qui peut prétendre que le traitement de texte utilisé pour taper ce petit article a réellement besoin de 18 Mo de RAM pour fonctionner, alors que je pouvais réaliser exactement le même travail avec son ancêtre sur mon premier PC sans arriver au bout de ses 1024 Ko ?

D'expérience, il est très rare de ne pas trouver de gisements importants de performance sur un code qui n'a jamais été audité ou optimisé auparavant. Tout simplement parce que, même avec des développeurs certifiés et qualifiés, il n'est pas dans les mœurs de penser les développements avec la performance en ligne de mire. Rappelons-nous que nous en sommes toujours à diffuser le pilotage des développements par les tests ! Et les performances viennent naturellement après la validité fonctionnelle.

Si vous pensez que tous les algorithmes standard sont maîtrisés et que 20% de CPU est une économie inatteignable, prenez l'exemple - pourtant simplissime - du calcul de la médiane d'une série de valeurs. Effectuer un tri préalable puis prendre la valeur du milieu est-il plus performant qu'un algorithme de type partitionnement récursif ? La majorité d'entre nous, lecteurs de Programmez ! et donc développeurs chevronnés, ne le sait pas. Et nos applications regorgent pourtant d'algorithmes maintes fois plus complexes...

Si vous n'êtes pas convaincu, prenons un autre exemple, cette fois-ci sur la consommation de bande passante. Lorsqu'on crée un service web, la plupart des frameworks sérialisent par défaut la totalité des objets renvoyés. Il n'est pas rare qu'un développeur oublie cette fonctionnalité bien pratique. Alors que l'interface ne requiert que l'envoi d'un nom et d'un montant, on se retrouve avec la totalité de la définition d'un objet et ses sous-objets véhiculée par HTTP, qui plus est sous une forme extrêmement verbeuse, à savoir XML. Sur l'interface locale d'une machine de développement, c'est invisible. Mais lorsque ce comportement est multiplié par des dizaines de milliers d'utilisateurs sur des années, on se rend compte qu'une partie, faible peut-être, mais belle et bien existante, des équipements réseau aurait pu être économisée.

Question de point de vue

Le gaspillage en informatique est beaucoup plus prégnant que dans le monde matériel. Malheureusement, son aspect abstrait le masque, alors

que les conséquences sont pourtant bien réelles et mesurables. Un projet informatique abandonné au bout de plusieurs années (et les exemples sont encore légion), ce sont des dizaines de machines devenues obsoètes, une consommation de ressources purement et simplement gaspillée, et pourtant cela nous interpelle moins que de voir de la nourriture gâchée, par exemple.

Quelles réponses ? Quelles méthodes ?

Une première partie de la réponse est d'adopter une attitude minimaliste, prônée par le Lean. Cette méthode a grandement fait ses preuves dans l'industrie, et s'adapte particulièrement bien au domaine informatique. En particulier, elle modélise sept sources de gaspillage : surproduction, inventaire, transport, traitement, temps d'attente, interaction, défaut de qualité. Sur la plupart de ces points, le développeur possède un levier d'action : ne pas créer de fonctionnalités qui ne servent pas, ne pas réaliser de traitements inutiles, diminuer le temps d'exécution et la consommation de ressources, créer du code de qualité le plus en amont possible, etc. Nous avons tous entendu parler du coût d'un bug qui augmente exponentiellement en fonction du retard avec lequel il est pris dans la chaîne de production : il faut être conscient que le coût écologique va systématiquement dans le même sens que le coût financier, car il procède de la même raison : la surconsommation de ressources. Le second point est plus un conseil pratique à

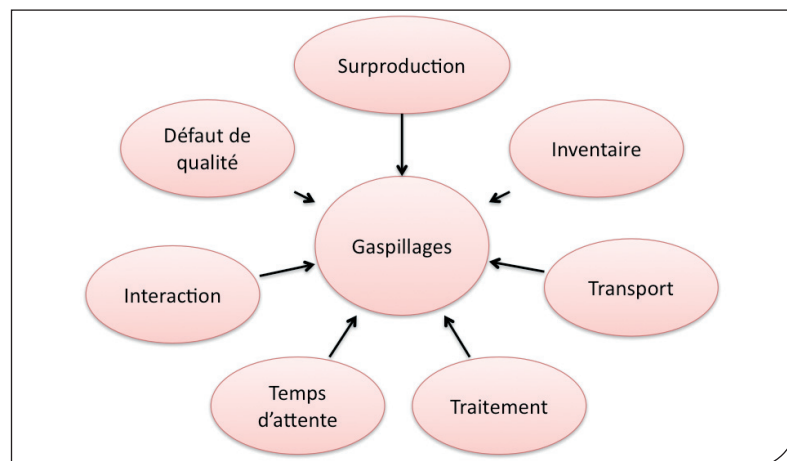
des développeurs : prenez le réflexe de réaliser de temps en temps une passe de profileur sur votre code. De nombreux outils existent, à des prix très abordables, l'opération ne vous prendra que quelques minutes, et pratiquement tout le temps, les résultats seront riches d'enseignements. Vous ne trouverez pas à chaque fois une grosse surconsommation de mémoire ou des pics CPU qu'il faudra expliquer puis réduire, mais même si cela ne vous permet que de mieux comprendre l'impact de vos développements sur le fonctionnement global de l'application, vous aurez rentabilisé votre temps.

Conclusion

Travailler sur le code source vous garantit que, pour un investissement de ressource virtuellement nul, vous aurez l'impact maximal sur l'ensemble de la chaîne informatique, car chaque instruction économisée se répercute sur l'achat de machines, la consommation électrique, la climatisation, etc.

Vous, développeurs, aurez toujours au bout des doigts la capacité de diviser les ressources consommées uniquement au prix de votre activité cérébrale, sans impact écologique ! Rien d'étonnant que le Lean et les méthodes agiles, qui privilégient l'humain sur les processus, soient aussi liés à l'élimination des gaspillages.

■ **Jean-Philippe Gougoux**
architecte logiciel en charge de la R&D chez MGDIS. Il a écrit une thèse sur la conception en vue du recyclage qui a été récompensée du prix Cherry de l'Université de Cranfield (GB). <http://blogs.dotnet-france.com/jeanphilippe/>



Programmation java : des Green Patterns ?

Le Green Challenge a fait émerger des patterns de développement permettant de réduire la consommation énergétique des applications Java : les "Green Patterns".

C'était un concours ouvert, lancé dans le cadre de l'Université du SI 2010, en partenariat avec GreenIT.fr. Il a mobilisé 15 équipes, soit une cinquantaine de développeurs, pendant 2 mois. Les gagnants ont remporté des iPads, iPods et des places pour l'USI.

A propos des QRcodes

Les QR Codes sont des codes-barres à 2 dimensions qui permettent à tout téléphone muni d'un appareil photo numérique d'accéder simplement au Web. En photographiant un QR Code, puis en le décodant, le téléphone peut récupérer le contenu d'une carte de visite ou se rendre sur un site Web.

Pourquoi travailler sur les QRcodes? Nous avons choisi de faire un challenge autour des QR Codes car le décodage de ces images consomme beaucoup de temps CPU. Il nous permet donc de mesurer une amélioration de la sobriété de l'application, chose que nous aurions eu du mal à quantifier avec une application peu consommatrice en CPU.

Le périmètre du Challenge

Pour identifier les "greens patterns", le challenge proposait de faire baisser la consommation sur un exemple d'application: QRDecode. L'objectif de l'application QRDecode est de décoder des QRcodes, d'afficher les coordonnées des contacts auxquels correspondent ces codes barres et de positionner ces contacts sur une carte. Une implémentation de référence était proposée, en voici une capture d'écran [Fig.1].

Schématiquement, si l'on se limite aux flux, le rôle de cette application est de transformer plusieurs dizaines de QR Codes représentés par des fichiers .QRC transmis à l'application en :

- Des cartes de visites,
- Des images (la représentation visuelle du QR Code),
- Des coordonnées GPS,
- Une carte affichant ces coordonnées [Fig.2].

Les différentes fonctionnalités de l'application QR Decode sont donc :

1. Le décodage des fichiers QR Code en une représentation carte de visite (en l'occurrence VCard),



2. La transformation des QR Codes en images,
3. La géolocalisation des adresses contenues dans les VCard pour obtenir des coordonnées GPS,
4. Le positionnement des points GPS sur une carte graphique,
5. L'affichage des cartes de visites et de la carte graphique.

L'implémentation de référence de l'application QR Decode a été réalisée en Java et se décompose en deux parties: une partie serveur qui s'exécute sur Google App Engine et une partie client qui s'exécute sur le navigateur. Les deux parties de l'application sont instrumentées pour récupérer le temps CPU consommé. Pour la partie serveur cela se réalise via des API spécifiques fournies avec le projet, pour la partie client cela se réalise par l'intermédiaire d'un plug-in FireFox développé pour l'occasion, GreenFox.

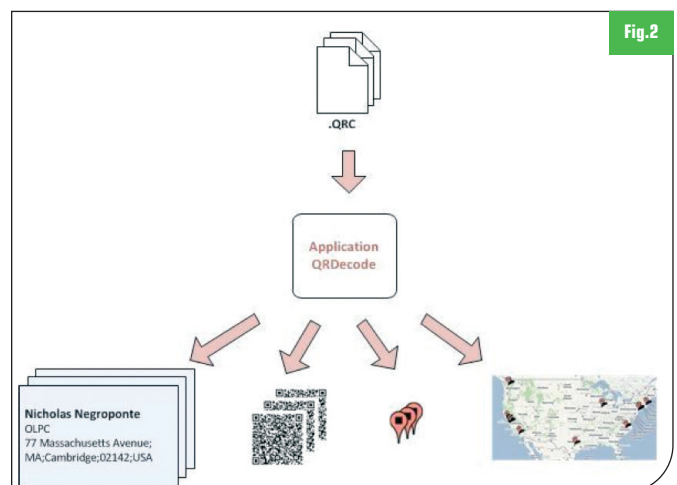
Présentation de GreenFox

GreenFox mesure le temps CPU consommé par l'utilisation de Firefox. Il fonctionne sous Firefox 3.6, pour l'instant sous Windows uniquement, à partir de Windows XP SP3. GreenFox s'installe sous forme de toolbar en bas de la fenêtre de votre navigateur. Il mesure le temps CPU consommé par le process FireFox, entre le moment où vous cliquez sur "start" et le moment où vous cliquez sur "stop". Il s'agit du temps CPU consommé par tout le process Firefox et pas uniquement de la fenêtre de votre application web. Donc il est recommandé de désactiver toutes les autres extensions afin de ne pas perturber la mesure. Chaque mesure est envoyée au serveur de collecte du Challenge [Fig.3].

Les équipes étaient jugées en fonction de la réduction de la consommation qu'elles apportaient à l'application de référence. Les paragraphes suivant décrivent les méthodes mises en œuvre par les différents participants.

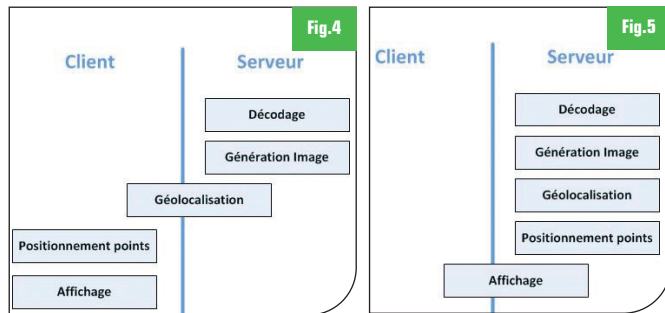
L'architecture et répartition des traitements

La bonne répartition des traitements entre le serveur et le client est un des éléments majeurs d'optimisation des performances



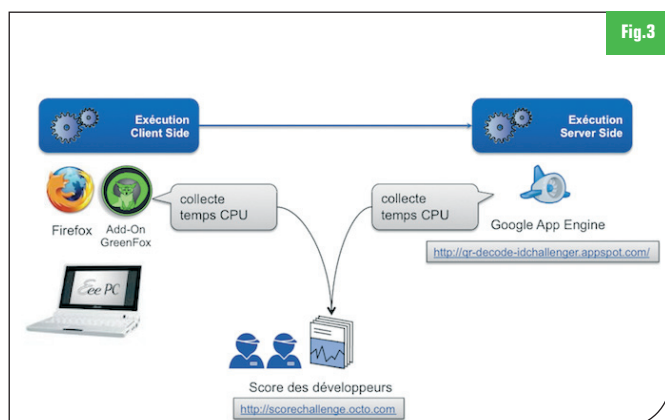
énergétiques de l'application. En effet, le code qui s'exécute côté serveur chez Google dispose de conditions énergétiques idéales (une plateforme hautement mutualisée, un PUE optimisé) alors que le code qui s'exécute sur la machine cliente, même dans les conditions minimales que nous avons choisies (un netbook de type Asus EeePC) reste celui d'un PC individuel avec une importante déperdition d'énergie. Dans l'implémentation de référence, les traitements se répartissaient comme suit : [Fig.4]

La première optimisation consistait donc à décaler le plus possible de traitements côté serveur. Voici la répartition "idéale" proposée par les deux premières équipes sur le podium [Fig.5].



La **géolocalisation** est effectuée par appel d'un traitement Google. Néanmoins le fait de le réaliser via un appel JavaScript est nettement plus coûteux que lorsqu'il est intégré dans le code serveur. De plus, il est effectué pour chaque adresse, ce qui est pénalisant. Une optimisation consiste à réaliser un appel batch sur le serveur pour réaliser la totalité du traitement en un seul appel. C'est possible via l'API Google mais cela impose des limites en nombre de requêtes lancées, un des gagnants a donc choisi d'utiliser MapQuest à la place.

Le **positionnement** des points sur la carte est également réalisé dans l'application de référence par l'API JavaScript de Google Maps. C'est encore une fois très coûteux en CPU. Deux stratégies ont été utilisées pour éviter cet écueil. Dans les deux cas il s'agissait de supprimer les appels JavaScript. La première stratégie consiste à réaliser une génération complète sur le serveur, cela est possible en utilisant la version statique de Google Maps qui génère une seule image intégrant la carte et tous les points [Fig.6]. Une autre stratégie était l'utilisation d'un Canvas HTML 5 dans la page avec un positionnement des points en relatif sur un simple fond de carte [Fig.7]. Dans les deux cas cela limite les fonctionnalités de l'application, ce qui était autorisé par le règlement.

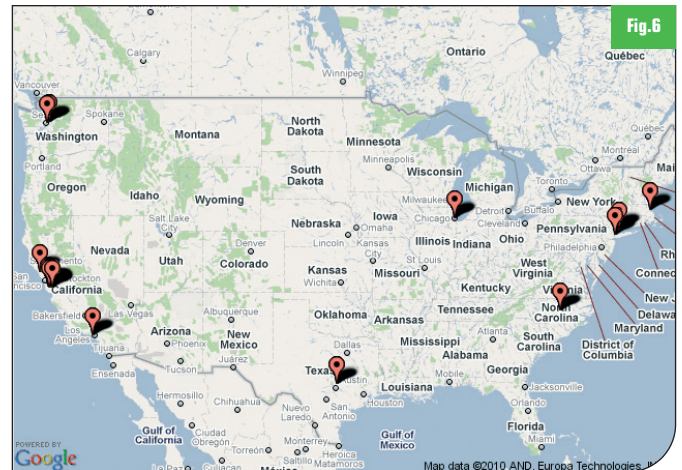


L'affichage se fait dans le navigateur et est donc obligatoirement côté client. Néanmoins, la plupart des candidats ont choisi d'alléger les traitements d'affichage en préparant le travail côté serveur. Ainsi, deux des équipes gagnantes ont directement encodé les images dans la page HTML (soit en base64 soit en passant par le data URI Scheme). Voir l'exemple ci-dessous.

```

```

L'affichage HTML des cartes de visites a également été généré côté serveur, ce qui permet d'éliminer le JavaScript qui génère le code HTML à partir de la représentation mémoire/JSON des données.



Optimisations « Green »

La plupart des équipes ont également travaillé sur l'optimisation des traitements côté serveur. Dans un premier temps, l'idée était d'effectuer un profiling de l'application afin d'identifier les lignes de code sur lequel le processeur passe le plus de temps. Plusieurs outils Java ont été utilisés pour cela : **Netbeans Profiler**, **Java VisualVM** ou **JavaProfiler** [Fig.8].

Voici les différentes optimisations réalisées :



Décodage des QRcodes

Le traitement de décodage des QR Codes est clairement le traitement le plus coûteux en temps processeur sur la partie serveur.

Dans l'implémentation de référence, le traitement s'appuyait sur la **librairie Open Source de Yusuke Yanbe** [Fig.9].

Deux des équipes sur le podium ont fait le choix de chercher et de benchmarker une autre librairie. Ils se sont donc appuyés sur la **librairie Zebra Crossing** qui offre plus de fonctionnalités (support d'un plus grand nombre de types de codes barres) et qui est surtout beaucoup plus performante que la librairie de départ. Une autre stratégie, plus complexe, consistait à optimiser la librairie existante. Pour cela un profiling plus fin et des optimisations sur le code Java ont été mises en œuvre. En particulier :

Fig.8

Call Tree - Method	Time (%)	Time	Invocations
qrbench.QrDecode.processDecode (String, jp.sourceforge.qrcode.QRCodeDecoder)		22757 ms (99.9%)	120
jp.sourceforge.qrcode.QRCodeDecoder.decode (jp.sourceforge.qrcode.data.QRCodeData)		17939 ms (78.8%)	120
jp.sourceforge.qrcode.QRCodeDecoder.decode (jp.sourceforge.qrcode.data.QRCodeData)		17937 ms (78.7%)	120
jp.sourceforge.qrcode.reader.QRCodeImageReader.getQRCodeSymbol (jp.sourceforge.qrcode.QRCodeImage)		13354 ms (58.6%)	120
jp.sourceforge.qrcode.pattern.FinderPattern.findFinderPattern (boolean)		6605 ms (29.3%)	120
jp.sourceforge.qrcode.pattern.FinderPattern.findLineAcross (boolean)		5842 ms (25.7%)	120
jp.sourceforge.qrcode.pattern.FinderPattern.findLineCross (jp.sourceforge.qrcode.pattern.FinderPattern)		723 ms (3.2%)	120
jp.sourceforge.qrcode.pattern.FinderPattern.getCenter (jp.sourceforge.qrcode.pattern.FinderPattern)		11.2 ms (0.0%)	120
jp.sourceforge.qrcode.pattern.FinderPattern.getAngle (jp.sourceforge.qrcode.pattern.FinderPattern)		8.45 ms (0.0%)	120
Self time		8.44 ms (0.0%)	120
jp.sourceforge.qrcode.pattern.FinderPattern.calcExactVersion (jp.sourceforge.qrcode.pattern.FinderPattern)		7.33 ms (0.0%)	120
jp.sourceforge.qrcode.pattern.FinderPattern.getWidth (boolean[])		2.35 ms (0.0%)	120
jp.sourceforge.qrcode.pattern.FinderPattern.sort (jp.sourceforge.qrcode.pattern.FinderPattern)		1.15 ms (0.0%)	120
jp.sourceforge.qrcode.pattern.FinderPattern.calcRoughVersion (jp.sourceforge.qrcode.pattern.FinderPattern)		0.283 ms (0.0%)	120
jp.sourceforge.qrcode.pattern.FinderPattern.<init> (jp.sourceforge.qrcode.pattern.FinderPattern)		0.051 ms (0.0%)	120
jp.sourceforge.qrcode.reader.QRCodeImageReader.getQRCodeMatrix (jp.sourceforge.qrcode.reader.QRCodeImageReader)		5015 ms (22.3%)	120
jp.sourceforge.qrcode.data.QRCodeSymbol.<init> (boolean[])		981 ms (4.3%)	120
jp.sourceforge.qrcode.reader.QRCodeImageReader.filterImage (int[])		388 ms (1.7%)	120
jp.sourceforge.qrcode.reader.QRCodeImageReader.setSmallestGrid (int)		251 ms (1.1%)	120



- Eviter les copies de blocs mémoires (voir exemple de code ci-dessous),
- Limiter le nombre d'objets à instancier et favoriser la réutilisation des instances,
- Passer en variable Static des tableaux de valeurs,
- Limiter les conversions de types,
- Limiter l'utilisation d'objet nécessitant de la synchronisation (ThreadSafe).

A noter que l'objectif de ces optimisations n'est pas de limiter l'usage mémoire (peu impactant énergétiquement) mais de limiter le nombre d'opérations pour alléger la CPU.

```
int[][] imageToIntArray(QRCodeImage image) {
    int width = image.getWidth();
    int height = image.getHeight();
    int[][] intImage = new int[width][height];
    for (int y = 0; y < height; y++) {
        for (int x = 0; x < width; x++) {
            intImage[x][y] = image.getPixel(x, y);
        }
    }
    return intImage;
}
```

Optimisation de la génération des images

Un autre traitement coûteux était la génération de l'image graphique du QR Code à partir de l'adresse.

Une implémentation était fournie dans l'application de référence. Deux équipes ont fait le choix de la substituer par un appel à une API GoogleChart qui propose cette fonctionnalité. Cela permet en un simple appel HTTP de disposer de l'image.

```

```



Même si cette option génère une économie énergétique, le jury a pris la décision de pénaliser ces deux équipes car l'appel à ce traitement n'est pas visible à travers nos instruments de mesure et pénalisait donc les autres équipes.

La plupart des autres équipes ont réalisé des optimisations sur le code de génération des images. L'optimisation la plus simple était de "rogné" la taille pour éviter le traitement de pixels inutiles.

Différentes optimisations ont également été réalisées sur le traitement pour éviter de manipuler des pixels de couleurs alors que l'image du QR Code est nécessairement en noir et blanc. Enfin, l'encodage de l'image en bitmap a été privilégié en évitant de passer par les API AWT qui sont peu performantes.

Optimisation de l'IHM

L'optimisation de l'affichage consistait d'abord à éviter l'utilisation

du JSP qui provoque un overhead, surtout au premier chargement. La plupart des équipes ont également pris la décision de construire le code HTML directement en utilisant des `StringBuilder` (voir exemple ci-dessous).

```
public String decode(StringBuilder sbCards, StringBuilder sbAlerts,
StringBuilder sbImages, File file, int id) throws Throwable {
    ...
    // HTML of vcard
    sbCards.append("<li class = \"vcard\">");
    // preparing the HTML5 canvas
    sbCards.append("<canvas id=\"canvas\"");
    sbCards.append(id);
    sbCards.append("\" width=\"");
    sbCards.append(wOut);
    sbCards.append("\" height=\"");
    sbCards.append(hOut);
    sbCards.append("\"></canvas>");
    sbCards.append("<span class=\"name\">");
    sbCards.append(sName);
    sbCards.append("</span>");
    sbCards.append("<span class=\"orga\">");
    sbCards.append(sOrga);
    sbCards.append("</span>");
    sbCards.append("<span class=\"addr\">");
    sbCards.append(sAddress);
    sbCards.append("</span>");
    sbCards.append("</li>' + r\n'");
    ...
}
```

La plupart des équipes ont aussi fait en sorte d'éviter les aller/retours qui nécessitent des traitements de connexions côté client et côté serveur. Une des équipes a fusionné dans la page HTML: le code HTML, la feuille CSS, les images et le JavaScript pour limiter les échanges à un seul aller/retour.

Le code HTML a été optimisé (suppression des espaces inutiles) par plusieurs équipes. Le code JavaScript a également été optimisé pour limiter le nombre d'appels AJAX qui impliquent des traitements (et donc de la CPU) pour réaliser les connexions. Le JavaScript a aussi été offensé pour limiter le parsing. La meilleure stratégie était néanmoins d'éviter complètement le JavaScript !

Autres optimisations

D'autres optimisations ont été mises en œuvre. La principale concerne la gestion des traces et du code de débogage. L'idée

étant de réaliser ces traitements de manière conditionnelle pour ne pas consommer du temps d'exécution inutile. Par exemple le code de trace :

```
canvas.println("Adjust AP(" + x + "," + y + ") to d(\"+dx+\", \"+dy+\")");
```

est remplacé par:

```
if (canvas.isPrintlnEnabled())
    canvas.println("Adjust AP({},{}) to d({},{})", x, y, dx, dy);
```

Enfin, plusieurs équipes ont mis en œuvre les options de gestion du cache côté navigateur qui peuvent être intéressantes si l'application s'exécute plusieurs fois. Néanmoins le jury portait systématiquement d'un cache vide avant chaque mesure.

Conclusion : les Green Patterns

Le premier enseignement que l'on peut tirer de ce challenge est que l'optimisation énergétique d'une application est une réalité. Les gains obtenus entre l'application de référence et les équipes gagnantes vont de 20% pour la partie serveur à un gain de plus de 600% sur la partie cliente.

On constate ensuite que les stratégies gagnantes pour limiter la consommation sont :

- Réaliser le maximum de traitements côté serveur quitte à réduire l'ergonomie à l'essentiel côté client,
- Profiler l'application pour identifier les traitements fortement consommateurs de CPU,
- Optimiser ces traitements pour limiter le nombre d'opérations réalisées ou remplacer ces traitements par des bibliothèques plus efficaces,
- Pré-générer les affichages sur le serveur et éviter le code interprété (JavaScript ou JSP).

C'est à ce prix que l'on aura des applications "vertes" mais aussi, globalement, de meilleure qualité.

Un grand merci à tous les participants qui nous permettent d'appréhender ces bonnes pratiques.

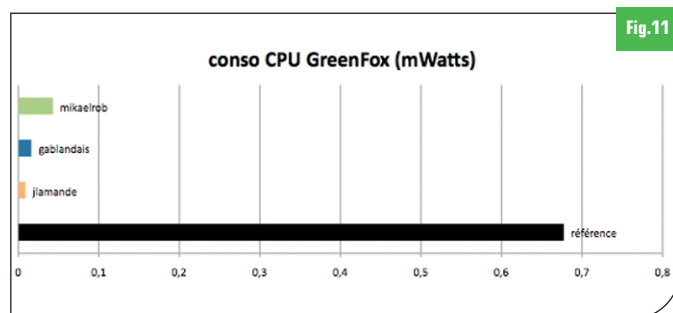
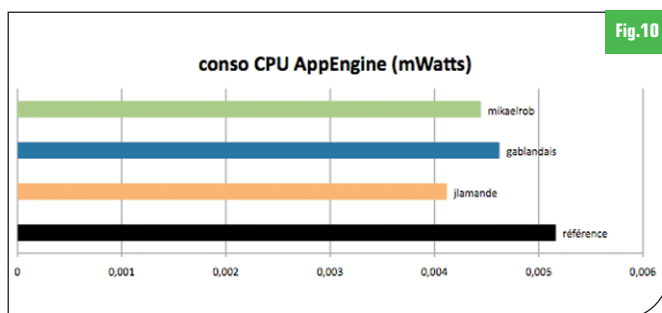
Pour en savoir plus sur le challenge et GreenFox :

<https://sites.google.com/a/octo.com/green-challenge/>

■ Lionel **Laske**, C2S, Groupe Bouygues

■ Frédéric **Bordage**, www.greenit.fr

■ Guillaume **Plouin**, OCTO Technology



Comment créer des applications plus performantes et moins gourmandes ?

La consommation, voilà bien un mot utilisé partout et pour tout. Lorsque vous achetez une TV, vous aller chercher celle qui consomme le moins et qui a la meilleure qualité d'image. Lorsque vous achetez un lave-vaisselle, même combat : plus il a de programmes et moins il consomme, mieux c'est !

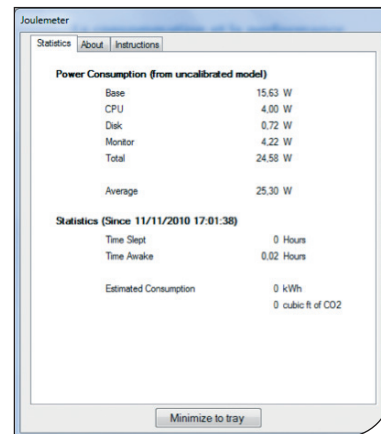
Si on regarde les différents consommateurs énergétiques d'une TV, on a bien sûr le type d'écran. Il est clair qu'une TV LED consomme moins qu'une plasma (en général 20% de moins). On peut également regarder les différents composants électroniques. Certains consomment plus que d'autres. Ainsi nous avons vu apparaître des processeurs de moins en moins gourmands en énergie.

Diminuer la consommation n'a pas pour unique but de diminuer la facture énergétique du client. On peut remarquer que le combat des constructeurs d'ordinateurs se base entre autres sur la durée de vie des batteries. On observe le même phénomène sur les mobiles. Souvenez-vous lorsque la fonction première de votre portable était de téléphoner, combien de temps durait votre batterie ? Plusieurs jours, voire plusieurs semaines ! Aujourd'hui, nos appareils de poche se connectent à internet pour regarder nos mails, nous permettent d'écouter de la musique en streaming et j'en passe. Depuis l'iPhone et son écosystème d'applications, notre téléphone devient un véritable couteau suisse. Malheureusement, l'évolution des batteries ne suit pas celui des usages, c'est pourquoi l'appareil se doit de consommer le moins possible.

De plus en plus de contraintes et de législations forcent les sociétés à créer des appareils de moins en moins énergivores. Le simple fait de consommer moins qu'un autre peut avoir un impact significatif sur votre produit. Certains secteurs bénéficient de primes, ce qui rend votre produit

plus compétitif. Si le matériel influence beaucoup sur la consommation d'énergie, ce n'est pas le seul facteur. Les développeurs, de par la manière dont ils utilisent ce matériel, peuvent influencer la consommation. Un des arguments d'Apple pour ne pas autoriser Flash sur l'iPhone / iPad est d'ailleurs sa consommation. Ceux qui ont Android avec Flash le savent bien. Si nous revenons au tout début de l'informatique, le langage de programmation était fort lié au hardware. Ainsi nous développions en assembleur. Les programmes étaient alors très performants et peu consommateurs (au niveau logiciel) mais très spécifiques et très longs à développer. Nous avons vu apparaître par la suite des langages de programmation plus faciles à réutiliser et un peu moins spécifiques comme le langage C. Mais l'informatique a évolué de plus en plus vite. Le marché réclamait des nouveautés toujours plus rapidement. Dans un même temps, le hardware grand public est devenu de plus en plus puissant et les contraintes logicielles se sont atténuées, permettant le développement de langage d'abstraction énergivore mais tellement plus simple et plus facile à utiliser et réutiliser : la programmation orientée objet. Ce type de programmation est en complète contradiction avec la consommation énergétique mais en complète adéquation avec le marché : time-to-market. Pendant longtemps la programmation respectueuse de la consommation était dédiée à des secteurs spécifiques avec des contraintes énergétiques réelles.

La programmation orientée objet est devenue une norme. Pas seulement



parce que le code est plus facilement réutilisable mais également parce qu'il apporte plus de sécurité, notamment grâce à l'encapsulation. Les langages les plus utilisés sont entièrement ou partiellement sous ce modèle et ceci ne devrait pas changer malgré les problèmes énergétiques auxquels les développeurs sont confrontés.

La consommation et la performance

On peut dire sans se tromper que la consommation est intimement liée à la performance. Un programme qui consomme peu d'énergie a plus de chance d'être également un programme performant.

Le logiciel ne consomme pas directement de l'énergie. Celle-ci est dépensée au travers du hardware. Ainsi un programme qui utilise beaucoup de CPU va entraîner une sur-utilisation des processeurs. Ceux-ci consomment de l'énergie et chauffent les autres unités, ce qui va entraîner l'augmentation de l'utilisation des ventilateurs, eux aussi consommateurs d'énergie. L'utilisation du CPU est une bonne mesure de la performance du

logiciel. Un programme utilisant moins de CPU et de mémoire aura plus de chance d'aller plus rapidement qu'un programme utilisant 100% de votre CPU et 100% de votre mémoire. Néanmoins, la consommation est plus intimement liée au CPU qu'à la mémoire. Le caching est dès lors un bon moyen de diminuer sa consommation malgré l'usage de la mémoire. Pour suivre l'évolution de la consommation de votre ordinateur, vous pouvez télécharger « Joulemeter » développé par Microsoft Research. Pour vous le procurer, suivez le lien : <http://research.microsoft.com/en-us/downloads/fe9e10c5-5c5b-450c-a674-daf55565f794>.

Comme on le voit sur la capture d'écran, la mémoire n'est pas une grande consommatrice comparée au moniteur et au CPU. Lorsque j'ai pris cette image, l'utilisation de mon CPU n'était que de 7%. En ouvrant mon logiciel de développement (ici en l'occurrence Visual Studio 2010), ma consommation CPU a eu des piques à 11 W, soit quasiment le triple !

Par contre, ouvrir plusieurs occurrences d'un même programme cause moins de soucis, probablement grâce à des systèmes de caching.

Si la puissance des ordinateurs a fait oublier aux développeurs les bonnes pratiques de développement où 1 millième de seconde avait de l'importance, le besoin de mobilité actuel et donc l'utilisation adéquate de la batterie le rappelle à l'ordre. A titre d'exemple, en C#, l'utilisation de liste générique de « string » va s'avérer tantôt plus performante tantôt moins performante qu'une StringCollection. Notamment lorsque votre but est de rassembler des listes contenant des milliers d'éléments. Ce simple changement dans votre programme peut largement diminuer la consommation tout en augmentant la performance.

La programmation orientée objet et consommation

La programmation orientée objet est un ensemble de concepts. En POO, tout type de fonction ou de donnée réside dans une classe. Lorsque le développeur veut utiliser une classe, il

doit l'instancier, c'est-à-dire créer un objet. Il existe différents types de conteneurs en C#. Typiquement nous pouvons comparer d'une certaine manière struct et class. Le mot clef struct est fort utilisé pour définir un ensemble de variables ayant un lien. A titre d'exemple, nous trouvons sur MSDN ceci :

```
public struct Book
{
    public decimal price;
    public string title;
    public string author;
}
```

En effet, un livre est souvent lié à un prix, un titre et un auteur. On aurait pu créer la même chose avec une classe :

```
public class Book
{
    public decimal price;
    public string title;
    public string author;
}
```

Si on mesure différents paramètres pour déterminer la consommation comme :

- User Processor Time
- Privileged Processor Time
- Total Processor Time
- Memory usage

On remarque très vite que l'utilisation de struct consomme 40% de plus !

Un autre concept bien connu du POO est l'héritage. En C#, cela nous donne la notion de classe abstraite et d'interface. Bien sûr, une classe peut hériter d'une autre mais nous n'allons pas analyser cette partie. Une classe abstraite, à la différence d'une interface, peut contenir une définition d'une méthode. Ces deux concepts sont très largement utilisés dans les programmes actuels. Si nous effectuons le même type qu'auparavant,

on remarque une légère différence en faveur de la classe abstraite.

Vous vous en doutez très certainement, une variable static consomme beaucoup moins qu'une variable dynamique. Ceci est important à noter car même si la plupart des développeurs le savent, ils oublient souvent de rendre la variable static.

Par contre lorsque l'on parle de méthode, il n'en va pas de même. Les méthodes static consomment 50% de plus que les méthodes normales (public ou private).

Avec la venue de LINQ en .NET, de plus en plus de personnes se sont habituées à utiliser les méthodes dynamiques, notamment avec les delegate, les actions ou les fonctions (Func). Sachez que ces méthodes anonymes consomment quant à elles 80% de plus qu'une méthode normale. On peut dès lors obtenir un gain de performance et de consommation en utilisant moins ce type de méthode. La plus part du temps les développeurs peuvent éviter ce genre d'écriture. On l'utilise souvent à tort pour diminuer les lignes de code.

La notion d'héritage de la POO a amené aussi le concept d'attributs et méthodes protégés (protected). Souvent, les développeurs utilisent ce mot clef sur toutes les méthodes et variables d'une classe dont ils savent d'avance qu'elle va être héritée. Rarement des variables vont être redéfinies. Leur utilisation engendre une sur-consommation de 40%. Pour ce qui est des méthodes, elles ne sont pas plus consommatrices que d'autres.

Vous pouvez trouver ci-dessous un tableau représentatif des choix à effectuer lorsque vous programmez. Tout ceci ne doit néanmoins pas vous empêcher de garder les bonnes pratiques en tête.

■ Loïc Bar

A faire	1er choix	2e choix	3e choix
Liste d'attributs (variables)	Class	Struct	
Définition	Abstract class	Interface	
Type d'attribut	Statique	Dynamique	
Type de méthode	Dynamique	Statique	Anonyme
Accessibilité d'attribut	Private	Public	Protected
Accessibilité de méthode	Protected	Public	Private

Les outils des Décideurs Informatiques

*Vous avez besoin d'info
sur des sujets
d'administration,
de sécurité, de progiciel,
de projets ?
Accédez directement
à l'information ciblée.*

Cas clients

Actu triée par secteur

Avis d'Experts



Actus / Evénements / Newsletter / Vidéos

www.solutions-logiciels.com



☐ **OUI, je m'abonne** (écrire en lettres capitales)

Envoyer par la poste à : Solutions Logiciels, service Diffusion, 22 rue René Boulanger, 75472 PARIS - ou par fax : 01 55 56 70 20

1 an : 30€ au lieu de 36€, prix au numéro (Tarif France métropolitaine) - Autres destinations : CEE et Suisse : 36€ - Algérie, Maroc, Tunisie : 36€ - Canada : 48€ - Dom : 45€ - Tom : 60€
6 numéros par an.

☐ M. ☐ Mme ☐ Mlle Société

Titre : Fonction : ☐ Directeur informatique ☐ Responsable informatique ☐ Chef de projet ☐ Admin ☐ Autre

NOM Prénom

N° rue

Complément

Code postal : Ville

Adresse mail

☐ Je joins mon règlement par chèque à l'ordre de SOLUTIONS LOGICIELS ☐ Je souhaite régler à réception de facture

Certains **profils** sont précieux pour les recruteurs



Expectra, filiale du Groupe Randstad France, spécialisée dans la sélection et la mise à disposition en CDI et Intérim de profils « hautes compétences, vient d'éditer le « Guide 2010 des métiers en tension ». Fruit d'une enquête réalisée en partenariat avec les deux sites d'emploi LesJeudis.com et RegionsJob.com, ce guide analyse 19 profils. Nous en avons extrait trois fiches métiers.

Source :
Baromètre
des salaires
Expectra 2010
Lesjeudis.com

L'étude Expectra témoigne d'un paradoxe : malgré la crise et l'augmentation du nombre de chercheurs d'emploi, les recruteurs peinent toujours à recruter dans plusieurs filières. Informatique, ingénierie, finance, commercial,... autant de domaines dans lesquels les candidats manquent à l'appel. Incompatibilité entre compétences recherchées et profils disponibles, salariés préférant rester en poste plutôt que de tenter leur chance ailleurs, entreprises peu enclines à ouvrir leurs critères de recrutement... les raisons des tensions sont multiples et complexes. Le point sur ces métiers toujours en pénurie en 2010.

Exigence : des qualités de communication et de gestion de projet

Dans l'informatique et les télécoms, les recruteurs exigent à présent de bonnes capacités en communication et parfois en gestion de projets, en plus des compétences techniques. D'où une difficulté de recrutement accrue sur certains profils, en particulier pour les techniciens support de niveau 3 (appelés à résoudre les incidents complexes), les techniciens télécoms, les ingénieurs sécurité, les chefs de projet NTIC et les ingénieurs développement.

« Les ingénieurs JAVA font partie des plus durs à trouver, surtout en provin-

ce », précise Evelyne Achache, administratrice du Syntec Recrutement. Sur Paris en effet, les recherches se révèlent un peu moins compliquées, la région étant plus attractive pour ces professions.

Recruter des généralistes et leur donner des formations sur-mesure

La solution pour parer à la pénurie ? Recruter des profils généralistes et les former aux besoins spécifiques des entreprises, selon Evelyne Achache : « de plus en plus, les SSII mettent en place des systèmes de tutorat dans ce but ».

Lire l'étude complète : <http://www.expectra.fr/552/metiers-en-tension-expectra-devoile-son-etude-2010>

Ingénieur sécurité Gardien du capital immatériel

Un expert doté d'une vision stratégique

L'ingénieur sécurité est avant tout un technicien de haut niveau, qui maîtrise à la fois les réseaux, les bases de données, les systèmes d'exploitation et sait repérer les failles dans tous les recoins du système informatique. Menant souvent sa mission seul au sein de la Direction des systèmes d'information, il doit user de diplomatie et de pédagogie pour convaincre ses collègues

informaticiens mais aussi les salariés de l'entreprise de respecter certaines règles. Enfin, sa fiabilité et sa loyauté envers l'entreprise constituent bien entendu des critères de sélection particulièrement importants.

Rémunération	
Mini	28870 €
Médian	35370 €
Maxi	46840 €

Une somme de compétences qui font de ce professionnel une personne rare !

Les compétences les plus recherchées en 2010 :

Maîtrise des outils comme Checkpoint, Juniper, Fortinet..., compétences techniques générales sur les infrastructures de sécurité : infrastructure à clés publiques (PKI), chiffrement, authentification, systèmes Windows, Unix, bilingue Anglais.

Ingénieur développement Fabricant d'applications informatiques

Profil type

- Âge moyen : 33 ans
- Expérience : 6 ans
- Formation : Bac +5

Les + salaire

- Expérience en multimédia ou dans les outils de bug tracking (Mantis, Bugzilla)
- Expérience d'au moins 3 ans à l'international
- Maîtrise de technologie GED (Sharepoint)

Un expert très opérationnel

Les domaines en tension actuellement concernent les développeurs sur de nouvelles technologies ou langages (Java, JEE, .Net, C#, PHP, web services, Sharepoint, HTML).

Reportés avec la crise, les projets de développement IT ont repris depuis début 2010 : les évolutions et modernisations des systèmes d'information sont plus que jamais stratégiques et l'évolution des technologies les rend incontournables. En particulier, les développements dans le secteur de la Banque finance, du cloud computing, de l'internet 2.0 et de la téléphonie mobile génèrent de très fortes demandes. Même s'il y a beaucoup d'ingénieurs développement, plutôt jeunes sur le marché, les profils plus seniors et donc immédiatement opérationnels, ne sont pas si nombreux et sont donc très recherchés.

Les compétences les plus recherchées en 2010 :

Expert des langages (objet, Flex, Web-services, UML), maîtrise des outils de gestion de version / configuration (Subversion, Perforce, Clearcase) et des bases de données relationnelles de type Oracle, SQL Server, expérience en systèmes embarqués et temps réel (RTOS, norme POSIX), bilingue Anglais.

Attirés par la conduite de projet

Près de 90% des ingénieurs développement seraient prêts à changer de poste et d'entreprise, dont 61,4% dans un avenir proche.

Travaillant essentiellement chez les

L'ingénieur développement utilise des langages et plateformes qu'il doit parfaitement maîtriser et applique les normes de programmation mises en place par le chef de projet. Il intervient également en maintenance corrective et évolutive des applications.

clients finaux et les SSII, les ingénieurs développement apprécient également de rejoindre les équipes des éditeurs de logiciels.

Près de 60% des développeurs souhaitent évoluer vers la conduite de projet. L'ingénierie logicielle ainsi que le consulting les attirent également. Ils aspirent également à une progression de carrière par l'encadrement, ce qui représente un tiers des désirs d'évolution.

Rémunération

Mini	27 510 €
Médian	33 490 €
Maxi	39 540 €

Les régions où il fait bon vivre

	Mini	Médian	Maxi
IDF	31 930 €	38 090 €	47 300 €
Rhône Alpes	31 320 €	35 640 €	40 100 €
Sud Ouest	29 840 €	35 300 €	38 790 €
PACA	28 000 €	35 050 €	39 750 €

Source : Baromètre des salaires Expectra 2010. Base annuelle brute.

Tendances

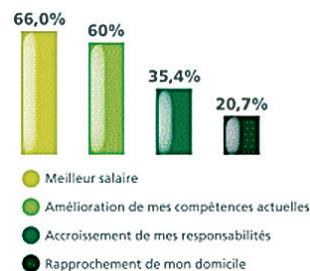
Les usines logicielles, une troisième voie durable ?

Traditionnellement recruté par les SSII, puis intégré au bout de quelques années chez un client final, l'ingénieur développement dispose aujourd'hui d'une troisième voie : intégrer l'un des nombreux centres de services créés par les SSII depuis quelques années. Situés en province, ils offrent à la fois la multiplicité des projets et une stabilité géographique que les ingénieurs apprécient.

Ces centres de services français – qu'on appelle nearshore – seront-ils progressivement transférés à l'étranger en offshore ?

La tendance de fond à la réinternalisation ou relocalisation des activités dans

Les critères de changement de poste ou d'entreprise



le secteur IT, devrait préserver l'emploi des ingénieurs et maintenir la forte demande. Par ailleurs, les besoins en développeurs des « anciennes technologies » (type Cobol) ne tarissent toujours pas, alors que les écoles ne les enseignent quasiment plus. Le développeur féru en anciennes et nouvelles technologies demeure une ressource rarissime !



Sophie Burgevin, Manager IT, Expectra Toulouse

« Les entreprises, en particulier les SSII, sont confrontées à un dilemme : d'un côté elles ont besoin de développeurs chevronnés, de l'autre les tarifs à la baisse ne leur permettent pas de les attirer avec des rémunérations suffisantes. La plupart des ingénieurs développement confirmés et spécialistes de Java ou de PHP sont actuellement en poste, et il est difficile de les convaincre de quitter leurs projets, souvent intéressants, et prendre le risque de rejoindre une nouvelle entreprise.

Les ingénieurs développement "seniors" très pointus dans leur domaine, apprécient, en premier lieu, avoir pour interlocuteur un recruteur qui connaît bien les technologies et leur univers. Ils ont besoin de se confronter à quelqu'un qui va pouvoir répondre à leurs questions et interrogations parfois assez techniques : le recruteur devra être au niveau **techniquement** pour rassurer le candidat. Ce premier contact peut être déterminant pour la suite du processus ».

Profil type

- Âge moyen : 37 ans
- Expérience : 12 ans
- Formation : Bac +5

Profil le plus recherché : connaissances en CD/CTI/CRM/Serveur Vocaux Interactifs/infrastructures systèmes/serveurs Windows/Unix/Citrix/bases de données Oracle et SQL server. Compétences managériales, très bon sens de la communication, rigoureux, organisé.

Chef de projet Pilote de l'avion multi-projet

Le chef de projet NTIC est en charge de la réalisation d'un projet informatique, dans les délais et les coûts prévus, avec un niveau de qualité optimal. Il recueille les besoins auprès de son client (interne ou externe), rédige les spécifications, planifie les phases de réalisation, puis coordonne l'équipe de développement et supervise les tests et la recette. Il est particulièrement recherché dans le domaine de la finance de marché et de la téléphonie sur IP.

Chef d'orchestre

Le rôle du chef de projet a fortement évolué depuis une dizaine d'années.

Autrefois meilleur technicien de son équipe, il en est aujourd'hui le meilleur manager, avec de fortes compétences techniques et métier.

En particulier, les dimensions liées à la conduite de projet – organisation, gestion des délais, reporting – sont devenues primordiales.

Communication, intermédiation, veille, gestion et partage de la connaissance, accompagnement du changement... les compétences requises sont très nombreuses.

Au-delà des savoir-faire techniques, la capacité à collaborer avec les équipes métiers et la connaissance des secteurs sont indispensables. Elles évitent en effet les problèmes de communication entre maîtrise d'ouvrage et maîtrise d'oeuvre, source de nombreux échecs.

Les compétences les plus recherchées en 2010 :

Expérience d'au moins 5 ans en conduite de projets informatiques, maîtrise des environnements Lan / Wan MPLS / IP / Voip / Toip, compétences sur les serveurs d'application Websphere / Weblogic.

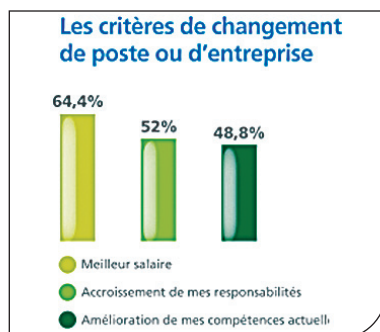
Rémunération			
Mini	30000€		
Médian	35000€		
Maxi	52680€		

Le match Paris/Province			
	Mini	Médian	Maxi
IDF	31730€	36890€	53490€
Province	28980€	32200€	46690€

Source : Baromètre des salaires Expectra 2010. Base annuelle brute.

Évoluer vers le consulting

Près de 95% des chefs de projet NTIC seraient prêts à changer de poste et d'entreprise, dont 72,5% dans un avenir proche.



Leurs objectifs ? Évoluer vers davantage de responsabilités d'encadrement ainsi que vers le domaine du conseil : ainsi, la moitié des chefs de projet seraient attirés par des fonctions de consulting et plus d'un quart d'entre eux par l'architecture IT.

S'ils apprécient d'intégrer une entreprise utilisatrice (46,6%), ils ne sont que 16,7% à souhaiter travailler en SSII, sociétés qui les recherchent pourtant très activement.

L'intérêt de leur mission constitue un critère majeur (49,4%), ainsi que la meilleure adéquation entre vie privée et professionnelle (22,3%). Un signe qui ne trompe pas sur les conditions stressantes du métier !

Tendances

Certification et management collaboratif

L'évolution fulgurante des technologies, en particulier celles de la convergence entre téléphonie mobile et informatique va fortement accentuer les besoins en chefs de projet formés

sur ces nouvelles technologies. Mais c'est l'aspect managérial qui reste le plus déterminant. Dépassement de délais et de budgets, démotivation des équipes ou incapacité à gérer des projets complexes avec des équipes à distance : les entreprises ne peuvent plus s'offrir le luxe de ces dérapages et envoient leurs chefs de projet en formation pour obtenir la certification PMI (Project Management Institute), méthode de conduite de projet éprouvée. Dans les nouveaux contextes, la fonction devient plus collaborative que hiérarchique, d'autant que l'éclatement géographique des équipes – en France comme à l'étranger – implique une excellente maîtrise du management interculturel.

Laurent Kermel, Manager référent IT, Expectra Paris



« Souvent anciens développeurs, les chefs de projet évoluent de la technique vers le métier et le management.

Ce chef d'orchestre doit être capable d'écouter les besoins des clients internes, de traduire et de vulgariser les contraintes IT. Le respect des délais et budgets est également essentiel.

Si les entreprises n'exigent pas systématiquement la certification PMI, elles sont en quête de profils qui connaissent la méthode et y sont sensibilisés.

En revanche les méthodes de type UML, et CMMi sont, dans des domaines différents, très demandées. Pour recruter le meilleur candidat, mieux vaut privilégier l'individu et ses qualités d'écoute et d'organisation plutôt que le technicien et donc ne pas hésiter à faire " monter " un développeur progressivement s'il fait preuve de motivation et de compétences de communication ».

egilia®

LEARNING

LE SPÉCIALISTE DE LA
FORMATION CERTIFIANTE
EN **INFORMATIQUE**
ET **MANAGEMENT**

Faire de vos succès
notre réussite

www.egilia.com

CONTACTEZ NOS CONSEILLERS FORMATION

 **N°National 0 800 800 900**

APPEL GRATUIT DEPUIS UN POSTE FIXE

ANVERS . LIEGE . PARIS . LYON . LILLE . AIX-EN-PROVENCE .
STRASBOURG . RENNES . BRUXELLES
TOULOUSE . BORDEAUX . GENEVE . LAUSANNE . ZURICH .

FraSCAti, prenez le contrôle de vos applications

2^e partie

Contrôler les applications en cours d'exécution n'est pas chose aisée. Nous présentons dans cet article différents moyens permettant de reprendre la main sur vos applications grâce à FraSCAti.

Dans le numéro précédent, nous avons vu comment SCA simplifie la réalisation d'applications orientées services tout en donnant un cadre architectural (SOA facile avec SCA). Nous allons ici nous intéresser à une autre préoccupation: comment observer une application en cours d'exécution, modifier sa configuration initiale, ou la faire évoluer sans la redéployer ? FraSCAti traite ces différentes problématiques en supportant la reconfiguration dynamique d'assemblages SCA. Nous les mettrons en pratique à l'aide de l'exemple introduit dans l'article précédent: MyWeather. Pour rappel, cet exemple permet d'interroger un compte Twitter afin de récupérer la localisation de l'utilisateur puis d'interroger un service météo pour connaître la météo à cette localisation. Nous compilerons cet exemple avec un script spécifique (compile, fourni avec les sources) afin de pouvoir développer un service technique (intent) intégré dans la plateforme FraSCAti.

Examiner les applications en cours d'exécution

FraSCAti Explorer est un outil capable d'observer, dans le détail, les applications en cours d'exécution. Il permet de naviguer dans l'architecture d'un assemblage SCA (les composants, leurs services, références, bindings, propriétés métiers et aspects techniques). Ceci prend tout son intérêt lorsque l'on sait que FraSCAti permet de faire évoluer cette architecture dynamiquement. FraSCAti Explorer est donc un véritable microscope vous permettant de visiter et faire évoluer votre application. Pour lancer FraSCAti Explorer, tapez `frascati explorer`. Chargez vos composites à l'aide d'un clic droit sur le domaine SCA et du menu contextuel Load. Naviguez dans votre système de fichiers jusqu'à l'archive `myWeather.jar`. Double-cliquez dessus pour naviguer à l'intérieur du jar et chargez le composite `myWeather.composite` [Fig.1].

L'explorer permet aussi de définir des plugins pour nos applications métiers. Dans cet exemple, nous fournissons un panel pour le service `tw`. Il sera affiché dans la partie droite lorsque le service sera sélectionné. Il nous permet d'invoquer le service par un simple clic

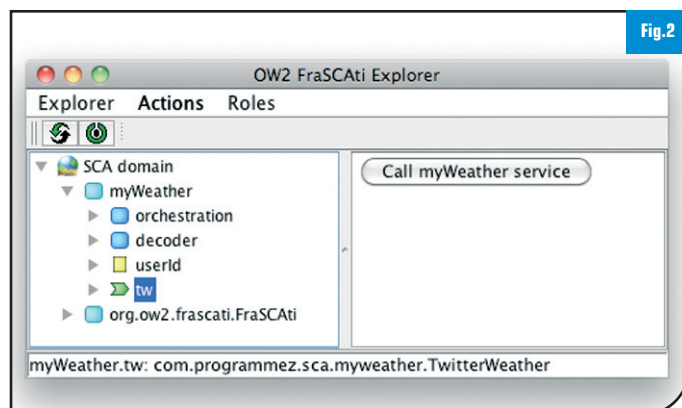
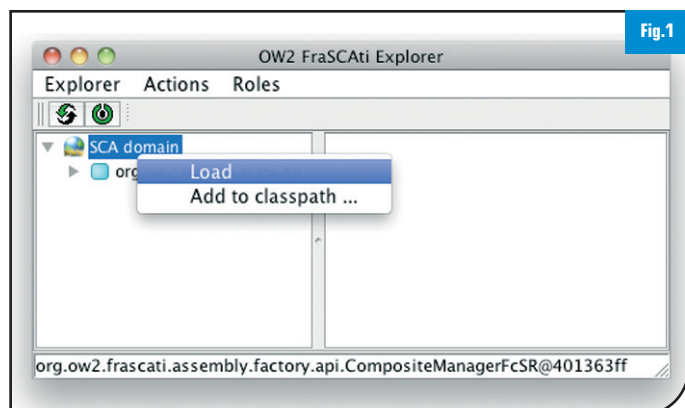
sur le bouton `Call myWeather service` et affiche le résultat de la requête [Fig.2]. Testons maintenant notre service en utilisant le panel. Vous devriez obtenir une réponse ressemblant à ceci :

```
Current weather in Lille:
<?xml version="1.0" encoding="utf-16"?>
<CurrentWeather>
  <Location>Lille, France (LFQQ) 50-34N 003-06E 52M</Location>
  <Time>Sep 28, 2010 - 07:00 AM EDT / 2010.09.28 1100 UTC</Time>
  <Wind> Variable at 2 MPH (2 KT):0</Wind>
  <Visibility> 2 mile(s):0</Visibility>
  <SkyConditions> overcast</SkyConditions>
  <Temperature> 57 F (14 C)</Temperature>
  <DewPoint> 57 F (14 C)</DewPoint>
  <RelativeHumidity> 100%</RelativeHumidity>
  <Pressure> 30.00 in. Hg (1016 hPa)</Pressure>
  <Status>Success</Status>
</CurrentWeather>
```

A noter que ce service n'est pas toujours disponible (saturation du serveur). Dans ce cas, l'application ne sera pas chargée et vous aurez le message d'erreur suivant: `WSDLException (at /html): faultCode=INVALID_WSDL`.

Reconfiguration dynamique

La visualisation est une chose mais l'interaction en est une autre. FraSCAti Explorer permet donc aussi d'interagir avec les applications. Il est possible, par exemple, de modifier la valeur d'une propriété sur un composant en cours d'exécution. Avec notre exemple, nous pouvons changer l'identifiant Twitter de l'utilisateur pour lequel nous souhaitons récupérer les informations. Pour vérifier que le changement a bien été pris en compte, il nous suffit d'invoquer à nouveau notre service grâce au panel disponible sur le service `tw` [Fig.3].



Avec FraSCAti Explorer, nous pouvons également ajouter (ou supprimer) des composants, mettre à jour une liaison (wire) ou un binding. Nous allons mettre en oeuvre ces techniques sur notre exemple. Supposons que le service météo que nous utilisons soit en panne. Nous allons reconfigurer notre application afin qu'elle utilise un autre service météo. En général, les API des services ne sont pas les mêmes. Pour contourner ce problème, nous ajouterons un composant adaptateur qui respectera l'interface du service météo utilisé par défaut, et traduira les appels vers notre nouveau service météo. Voyons l'implantation de ce composant :

```
public class WundergroundProxy implements GlobalWeatherSoap {
    /** Reference to the Weather Underground service. */
    @Reference private Wunderground weatherBackup;

    @Override
    public String getCitiesByCountry(String country) {
        // Not implemented
        return null;
    }

    @Override
    public String getWeather(String city, String country) {
        StringBuilder sb = new StringBuilder().append(city).append(
            ',').append(country);
        Forecast forecast = weatherBackup.getWeather( sb.toString() );
        return forecast.getTxt_forecast().getForecastdays().get(0)
            .getFcttext();
    }
}
```

Nous allons charger ce nouveau composite SCA myWeather-backup via le menu Load de FraSCAti Explorer. Une fois chargé, nous ajoutons le composant wunderground dans notre composite myWeather. Ceci se réalise en faisant un glisser/déposer du composant wunderground vers le composite myWeather.

Nous supprimons la liaison entre le composant myWeather et le service météo défaillant, en supprimant le "binding" web service sur la référence weather via un clic droit sur le binding (voir [Fig.4])

Ensuite, nous stoppons le composant orchestration en le sélectionnant, puis avec un clic droit nous choisissons l'action stop. Nous pouvons alors réaliser la liaison (wire) entre sa référence weather et notre service météo de secours wunderground. Notez que lorsque le composant orchestration est stoppé, tous les appels émis vers ce

composant sont mis en attente et seront traités une fois le composant redémarré. Ainsi, le service d'orchestration est toujours disponible vis-à-vis du client.

Toujours à l'aide du clic droit, mais cette fois sur la référence, nous sélectionnons le menu "Wire ...". Une boîte de dialogue s'ouvre, nous sélectionnons le service weather du composant wunderground qui se trouve maintenant dans le même composite que le composant orchestration. Nous validons en cliquant sur OK et nous redémarrons le composant orchestration, notre reconfiguration est terminée ! [Fig.5]

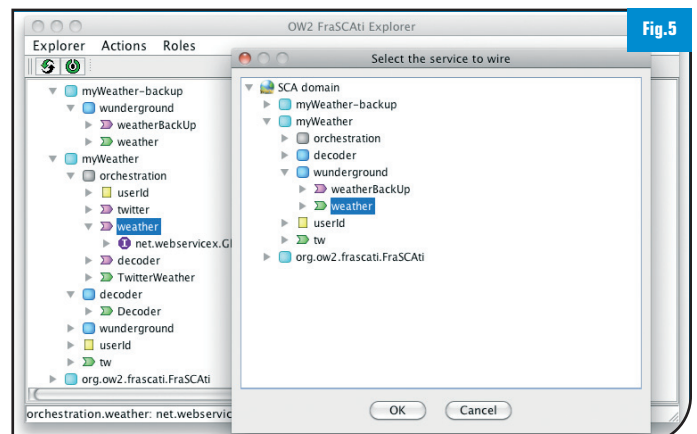
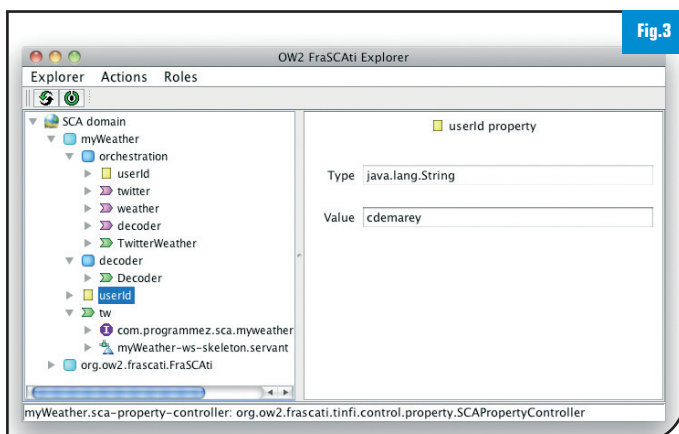
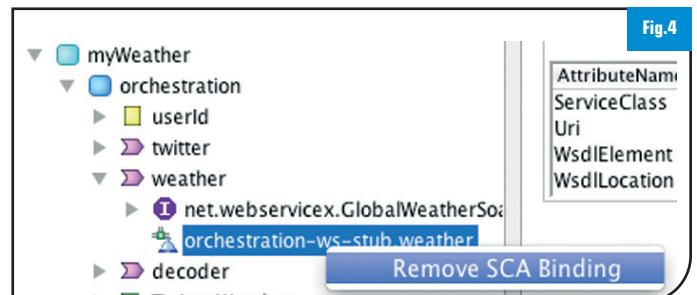
Appelons de nouveau notre service grâce au panel disponible sur le service tw. Nous voyons maintenant que nous obtenons toujours les informations météo mais légèrement différentes aussi bien au niveau des données que du format. C'est logique car nous avons changé de fournisseur de service météo.

FraSCAti FScript, langage d'interrogation et de reconfiguration d'architecture

Nous avons, pendant l'exécution, modifié l'architecture de notre application avec l'outil graphique fourni par FraSCAti. Ceci est très pratique pour le test ou réaliser de petites modifications d'architecture, mais cela est beaucoup moins envisageable pour des applications en production.

FraSCAti propose d'autres moyens de reconfigurer une application autrement que par son outil de visualisation: soit via une API Java spécifique, soit via un langage dédié particulièrement adapté aux interrogations d'architecture : FraSCAti FScript (extension SCA de FScript, <http://fractal.ow2.org/fscript>).

Ce langage possède une syntaxe similaire à XPath. Nous ne détaillerons pas toutes ses possibilités ici (plus d'informations sur <http://frascati.ow2.org/doc/current/ch08.html>). Nous ne retiendrons que quelques éléments essentiels de FraSCAti FScript : l'utilisation d'axes de navigation pour parcourir une architecture (scachild, scaservice, sca-property, scawire, scabinding, scaintent, etc.), la présence de



noeuds sources et cibles de chaque côté d'un axe, et la possibilité de définir des variables et des procédures. Pour mieux comprendre, prenons un exemple :

```
$domain/scachild::*
```

L'instruction ci-dessus demande au domaine SCA (variable prédéfini) la liste de ses fils (sans restriction sur le nom des fils). Le résultat est le suivant :

```
[#<scacomponent: myWeather>, #<scacomponent: myWeather-backup>,  
#<scacomponent: org.ow2.frascati.FraSCAti>]
```

Maintenant, si nous désirons obtenir un fils particulier, nous utilisons l'expression suivante :

```
orch = $domain/scachild::myWeather/scachild::orchestration;
```

Nous voyons ci-dessus que nous pouvons enchaîner les axes de navigation et stocker le résultat (des noeuds) dans une variable. Il existe aussi quelques raccourcis. L'expression ci-dessus est équivalente à :

```
orch = $domain/scadescendant::orchestration;
```

Pour obtenir toutes les références du composant orchestration, nous utilisons :

```
$orch/scareference::*
```

Après un bref aperçu de ce langage, intéressons-nous à un cas d'utilisation. Si nous reprenons la reconfiguration dynamique effectuée avec FraSCAti Explorer, nous pouvons la traduire avec la procédure FraSCAti Script suivante :

```
action switchToWundergroundService() {  
    orchestration = $domain/scadescendant::orchestration;  
    weatherRef = $orchestration/scareference::weather;  
    wsBinding = $weatherRef/scabinding::*;  
    remove-scabinding($weatherRef, $wsBinding);  
  
    wunderground = $domain/scadescendant::wunderground;  
    myWeather = $domain/scachild::myWeather;  
    add-scachild($myWeather, $wunderground);  
  
    set-state($orchestration, «STOPPED»);  
    wundergroundService = $wunderground/scaservice::*;
```

```
add-scawire($weatherRef, $wundergroundService);  
set-state($orchestration, «STARTED»);  
}
```

Cette procédure peut être enregistrée et réutilisée par le moteur de FraSCAti Script. Voyons comment. Tout d'abord, nous démarrons FraSCAti Explorer avec l'option -s afin d'activer le plugin FScript: `frascati explorer -s` [Fig.6].

Ensuite, nous chargeons les composites myWeather et myWeather-backup. A l'aide d'un clic droit sur le composite myWeather, choisissons FraSCAti Script console. Une console s'ouvre. Vous pourrez taper vos instructions FScript de manière interactive ou utiliser l'éditeur pour écrire des procédures, ouvrir des fichiers et charger des scripts dans le moteur. Nous allons cliquer sur Editor, taper la procédure décrite ci-dessus puis cliquer sur Register procedures. La procédure `switchToWundergroundService` est maintenant disponible. Cliquons à nouveau sur editor pour masquer la fenêtre et exécutons l'instruction suivante : `switchToWundergroundService()` La reconfiguration a été réalisée. Vous pouvez vous en assurer en rafraîchissant l'affichage de FraSCAti Explorer.

Le script de reconfiguration est bien plus rapide que les glisser/déposer réalisés précédemment avec FraSCAti Explorer. Mieux encore, nous aurions pu prévoir la panne du service météo et intégrer la gestion de cette panne dans notre application. Pour cela, deux possibilités s'offrent à nous : écrire un composant en Java et utiliser le moteur FraSCAti FScript depuis celui-ci (utile pour réaliser des traitements en plus de la reconfiguration) ou écrire un composant dont l'implantation sera un script FraSCAti FScript. Pour réaliser ce composant, il suffit de copier/coller l'action `switchToWundergroundService` dans un fichier que nous nommons `myreconfig.fscript`. Ensuite, dans le descripteur d'assemblage de myWeather-backup, ajoutons les lignes suivantes :

```
<component name="switch-to-wunderground-service">  
    <frascati:implementation.script script="myreconfig.fscript"/>  
    <service name="myReconfig">  
        <interface.java interface="com.programmez.sca.myweather.  
MyReconfig"/>  
    </service>  
</component>
```

Vous noterez que nous devons écrire une interface Java reflétant les procédures disponibles dans le script. Voici son implantation:

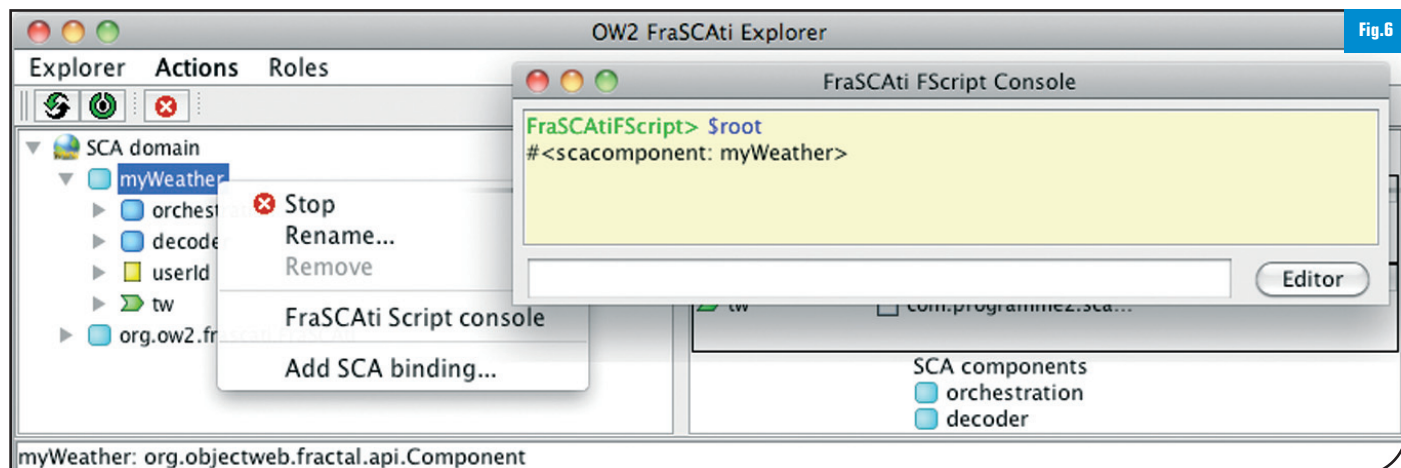


Fig.6


```
public interface MyReconfig
{
    void switchToWundergroundService();
}
```

Comme pour tout composant SCA, il est possible d'exposer l'action `switchToWundergroundService` via un protocole d'accès comme SOAP, RMI etc. Ainsi, l'architecture de notre application pourra être reconfigurée avec un simple appel distant.

En ajoutant un mécanisme qui contrôle la disponibilité du service, on obtiendra alors une application autonome, capable de s'adapter en cas de panne du service météo dont elle dépend.

Ajouter des aspects non-fonctionnels

La spécification SCA propose aux développeurs de composants de définir des intentions (intents SCA) sur les services et références des composants. Les intentions traduisent en général des préoccupations non fonctionnelles telles que authentification, confidentialité, etc. que l'on décrit à haut niveau. Celles-ci sont généralement implantées par la plateforme SCA et appliquées au déploiement des composants. Ici, FraSCaTi innove sur deux aspects: libre au développeur de fournir ses propres intents. Pour ne pas le perturber, il développera ces intents sous forme de composants SCA, même formalisme pour le métier et le non fonctionnel. Second point, vous l'aurez peut-être deviné, ces intents peuvent être appliqués au déploiement mais aussi dynamiquement, lorsque les applications s'exécutent, grâce à un mécanisme dérivé de la programmation orientée aspects (AOP). Lorsque l'on place un intent sur un service ou une référence, un intercepteur est généré et déroute l'appel vers le composant réalisant l'intent. Il est alors possible de réaliser des pré et post-traitements, de reprendre le cours d'exécution normal de l'application.

Écrire un intent dans FraSCaTi est très facile. Il suffit d'écrire un composant SCA proposant un service dont l'interface est `org.ow2.frascati.tinfi.control.intent.IntentHandler`, pour laquelle il n'y a qu'une seule méthode à implanter : `invoke`. Voyons ceci sur un exemple simple de journalisation des appels sur un composant. Nous allons écrire cet intent log. Voici sa description d'architecture :

```
<component name="log">
  <implementation.java class="com.programmez.sca.intent.Log" />
  <service name="intent">
    <interface.java interface="org.ow2.frascati.tinfi.control.
intent.IntentHandler"/>
  </service>
  <property name="header">[FRASCATI-BASIC-LOG] </property>
</component>
```

Et maintenant l'implantation de la classe Log :

```
public class Log implements IntentHandler {
    /** A configurable header to add to log traces. */
    @Property
    protected String header;

    @Override
    public Object invoke(IntentJoinPoint ijp) throws Throwable {
        // Before the current invocation.
```

```
        System.err.println(header + « Before proceed»);
        // Proceed the current invocation.
        Object ret = ijp.proceed();
        System.err.println(header + « result of proceed:» + (ret==
=null ? «null» : ret.toString()));
        // After the current invocation.
        System.err.println(header + « After proceed»);
        return ret;
    }
}
```

Nous pouvons maintenant le charger dans FraSCaTi Explorer comme un composant classique. Pour utiliser cet intent, nous allons le tisser dynamiquement sur les services et références sur lesquels il doit s'appliquer. Par exemple, faisons un glisser/déposer du composant log vers le service `tw` pour tracer les appels à ce service. Nous voyons les traces suivantes s'afficher :

```
[FRASCATI-BASIC-LOG] Before proceed
[FRASCATI-BASIC-LOG] result of proceed: Current weather in Lille:
Chance of Rain. High 18°C (64°F). Winds 10 kph NNE
[FRASCATI-BASIC-LOG] After proceed
```

Comme notre intent de journalisation est un composant SCA, il peut lui aussi être reconfiguré [Fig.7].

Si nous avons voulu positionner cet intent au démarrage de l'application, nous aurions modifié la déclaration du service `tw` en ajoutant le mot-clé `requires` :

```
<service name="tw" promote="orchestration/TwitterWeather" requires
="log-intent">
```

De cette façon, nous indiquons la liste des intents à appliquer à un service ou une référence.

Conclusion

Nous avons pu voir tout au long de cet article qu'il est possible de garder le contrôle sur des applications en cours d'exécution: observation de l'architecture, reconfiguration de celle-ci, ajout/suppression d'aspects non fonctionnels, etc. Il est donc possible de faire évoluer des applications sans devoir les redéployer, permettant ainsi de pouvoir répondre à différentes préoccupations : gestion des pannes, corrections de bugs, évolutions à chaud.

■ Christophe Demarey et Damien Fournier
Ingénieurs de Recherche - INRIA.

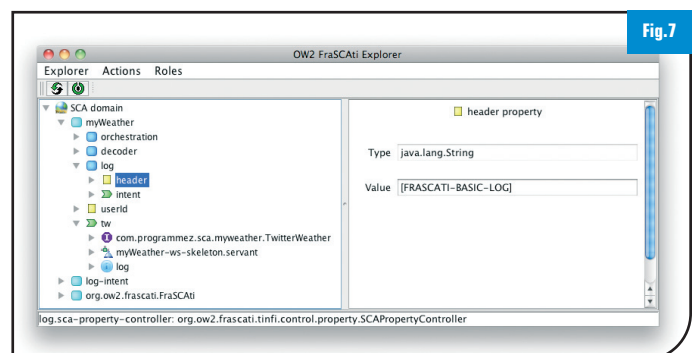
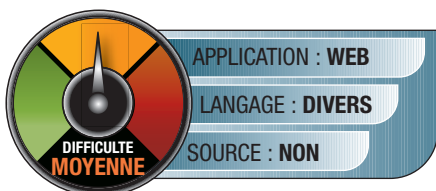


Fig.7

WebMatrix : simplifiez-vous le développement web

Microsoft a sorti cet été la première bêta de WebMatrix, un nouvel outil de développement web. Le développeur disposait déjà de Visual Studio, Visual Web Developer et Expression Web. Pourquoi un nouvel outil ?



Il restait cependant un trou dans la raquette. Les environnements de développement Microsoft sont aujourd'hui optimisés pour le développement d'applications ambitieuses : toute la richesse de .Net y est directement exposée, et cette richesse ne va pas sans une certaine complexité. Le but avéré de WebMatrix est de se focaliser sur des scénarios moins techniques et plus orientés tâches.

A cet effet, Microsoft a fait de grands efforts de simplification, pour fournir un outil de prise en main immédiate : WebMatrix.

INSTALLATION DE WEBMATRIX

WebMatrix s'installe de façon un peu particulière sur votre machine. L'installation se fait depuis la page <http://www.microsoft.com/web/webmatrix>, mais au lieu de télécharger un installateur et de lancer son exécution, vous allez simplement cliquer sur un lien de la forme suivante : <http://www.microsoft.com/web/gallery/install.aspx?appid=webmatrix>

Ce lien vous emmènera tout d'abord vers le téléchargement et l'installation de Web Platform Installer (en abrégé WebPI).

Cet outil est dédié à l'installation d'applications web, et couvre à la fois les pré-requis, les frameworks, les outils de développement web, et les applications web Open Source les plus populaires.

Une fois WebPI sur votre machine, un assistant vous guidera dans l'installation de WebMatrix et de tous les composants additionnels requis (notamment .Net 4, qui vous permet d'exploiter la nouvelle syntaxe Razor fournie dans WebMatrix et prochainement dans Visual Studio) : [Fig.1].



LE CONTENU DE WEBMATRIX

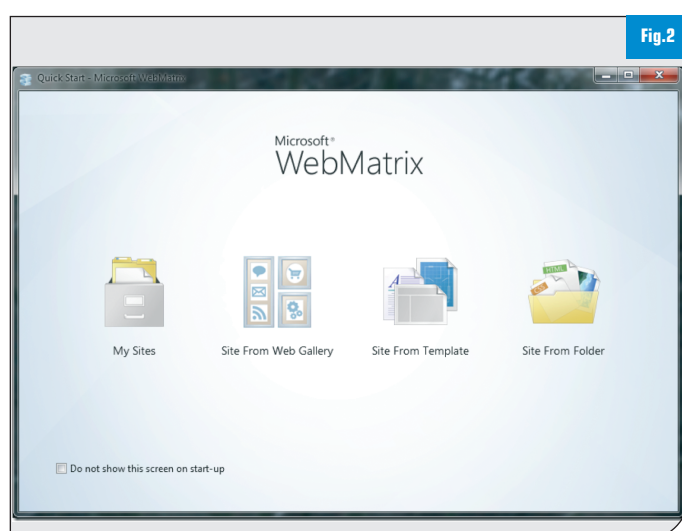
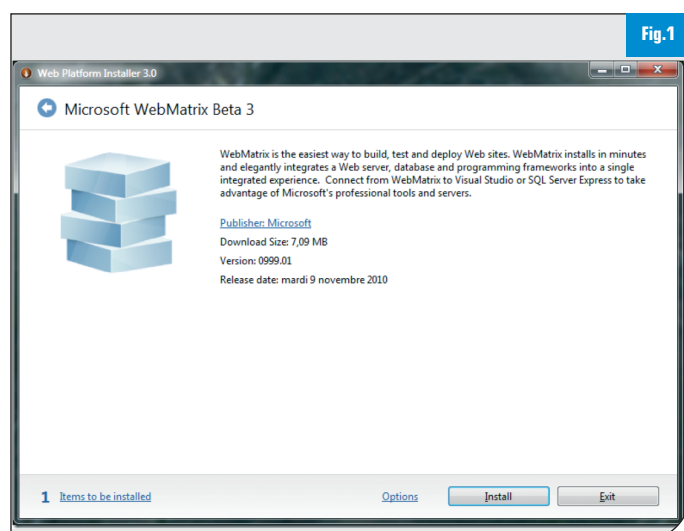
WebMatrix se veut un outil tout-en-un de développement, test et déploiement d'applications Web. Il reprend donc les composantes classiques d'une infrastructure web, tout en s'efforçant de simplifier au maximum chaque brique de l'édifice.

LE SERVEUR WEB : IIS EXPRESS

IIS Express est un serveur web directement tiré de IIS 7.5, la version fournie avec Windows Server 2008 R2. La principale différence est qu'il ne s'installe pas en tant que service (il est lancé en même temps que WebMatrix chaque fois que vous travaillez sur un site Web) et qu'il est du coup disponible sur toutes les versions de Windows depuis Windows XP Home jusqu'à Windows Server 2008 R2. Il n'a pas non plus besoin de droits administrateur pour s'exécuter, ce qui vous permet de travailler sur n'importe quelle machine.

LES FRAMEWORKS ET APPLICATIONS WEB

Beaucoup de nouveautés dans cette brique ! La première est la présence d'applications Web dans WebMatrix. Lorsque vous lancez WebMatrix pour la première fois, vous êtes accueilli par l'écran suivant : [Fig.2]. Le bouton "Site From Web Gallery" vous conduit à une liste d'applications Open Source, tant en PHP qu'en ASP.Net. Vous



serez sans doute surpris de trouver des applications PHP installables depuis un produit Microsoft, mais n'oubliez pas qu'à côté de sa casquette d'environnement de développement Web (ASP.Net est et restera chez Microsoft la principale technologie Web), Microsoft fournit un serveur web (IIS), et que ce serveur est parfaitement adapté à l'exécution d'applications PHP.

Ce bouton vous donne accès à une longue liste d'applications Web, que vous pouvez toutes installer en quelques secondes sur votre machine de développement: WebMatrix se charge de toutes les étapes d'installation, y compris l'installation et le paramétrage de PHP si nécessaire. Nous parlerons plus en détail des frameworks disponibles un peu plus tard, mais sachez que vous pourrez manipuler trois environnements ASP.Net : ASP.Net WebForms (disponible depuis ASP.Net 1), Asp.Net MVC (bientôt en version 3), et ASP.Net Web Pages (prochainement disponible dans Visual Studio).

LA BASE DE DONNÉES : SQL SERVER COMPACT

Pourquoi une nouvelle base de données ? En réalité, WebMatrix sait se connecter à trois bases de données différentes : Sql Server (2005 et ultérieures), MySQL (5 et 6), et SQL Server Compact. Cette dernière base de données a une grande importance dans les hébergements mutualisés car elle ne demande pas d'installer une base de données "complète" sur le serveur de production. C'est en effet une base de données fichier, directement embarquée dans l'arborescence du site web, que vous pouvez utiliser même si votre hébergeur n'offre pas SQL Server.

DÉCOUVERTE DE L'ENVIRONNEMENT DE DÉVELOPPEMENT

WebMatrix comporte toutes les fonctionnalités vues ci-dessus, mais c'est aussi un environnement de développement Web permettant d'écrire, de modifier et de publier vos propres applications Web [Fig.3].

Un éditeur de code

Cet éditeur de code vous propose une coloration syntaxique pour les fichiers suivants :

- **ASPX, ASCX, CS, VB**, et tous les formats couramment utilisés en ASP.Net
- **HTML, CSS**, Javascript, et tous les formats couramment utilisés pour le développement html

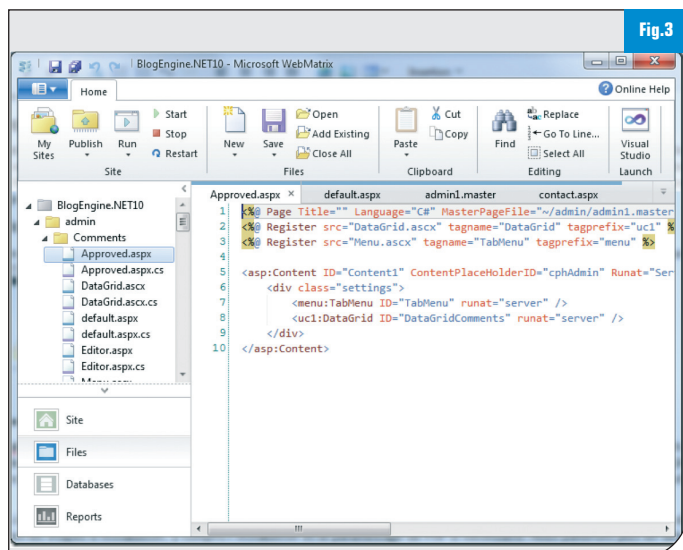


Fig.3

- **PHP, INC**, et tous les formats couramment utilisés en développement PHP
- **CSHTML et VBHTML** : les nouveaux formats de fichiers apportés par ASP.Net Web Pages

Des outils de test et de débogage

Commençons par un outil bienvenu des WebMasters, WebMatrix comporte un module d'optimisation des pages pour les moteurs de recherche. Cette fonctionnalité est tirée d'un outil appelé le SEO Toolkit (disponible sur <http://www.microsoft.com/web/seo/>) et qui inspecte votre site à la recherche de toutes les erreurs et maladroresses qui nuisent au bon référencement de votre site [Fig.4].

Les autres outils sont moins innovants mais bien pratiques, avec notamment un module qui trace toutes les requêtes parvenant à IIS Express et qui vous permet d'isoler rapidement les requêtes intéressantes (images, erreurs, et une fonction de recherche dans les requêtes) : [Fig.5].

Si vous avez l'habitude d'utiliser Visual Studio, deux différences importantes vont vous sauter aux yeux : d'une part, il n'y a pas d'intelliSense, uniquement de la coloration syntaxique (cette lacune sera probablement comblée dans la V2 au vu des demandes répétées de tous les bêta-testeurs), et il n'y a pas de notion de page de démarrage du site : WebMatrix lance directement la page que vous éditez quand vous appuyez sur le bouton Run. Ce bouton permet de lancer la page simultanément dans tous les navigateurs installés sur la machine. Un vrai plus.

Des outils de publication

WebMatrix comporte deux assistants de publication web, tous deux accessibles à partir du bouton Publish. Le premier utilise FTP, il n'y a rien de particulier à en dire, le second mode utilise Web Deploy, une technologie de déploiement automatisé introduite par IIS 7 [Fig.6].

Cette technique vous permet d'automatiser toutes les étapes du déploiement, y compris la gestion des permissions sur les répertoires et la création des bases de données. Elle dispose également d'un mode Mise à Jour qui ne transfère sur le serveur de destination que les fichiers dont le contenu a changé. Cette technique est plus riche, et est notamment à la base des applications disponibles dans WebMatrix, mais nous en parlerons dans un prochain article.

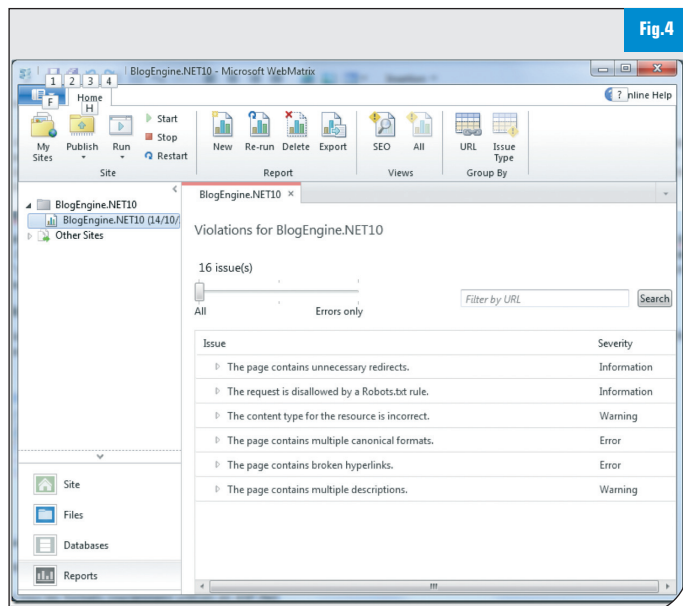


Fig.4

Razor, la nouvelle syntaxe d'ASP.Net Web Pages

Cette technologie de développement Web est une version simplifiée de ASP.Net MVC : elle repose sur l'un des moteurs de rendu d'ASP.Net MVC, appelé Razor. Les pages développées dans cette technologie utilisent les extensions de fichier cshtml et vbhtml, ce qui vous permet de mélanger les deux langages à votre gré dans un développement Web. Les caractéristiques de cette version d'ASP.Net sont les suivants : Accès simplifié aux bases de données :

```
4 var db = Database.Open("PhotoGallery");
5 var galleries = db.Query(@"SELECT Galleries.Id, Galleries.Name, COUNT(Photos.Id) AS PhotoCount
6 FROM Galleries LEFT OUTER JOIN Photos ON Galleries.Id = Photos.GalleryId
7 GROUP BY Galleries.Id, Galleries.Name").ToList();
```

Imbrication simple de code et de rendu html :

```
10 <h1>Galleries</h1>
11 @if (galleries.Count == 1) {
12     <p>Il n'y a qu'une galerie.</p>
13 } else {
14     <p>Il y a @galleries.Count galleries.</p>
15 }
16 <ul class="thumbnails gallery">
17     @foreach (var gallery in galleries) {
18         <li class="gallery">
19             <a href="@HrefAttribute("~/Gallery/View", gallery.Id)">
20                 
21                 <span class="below-image">@gallery.Name</span>
22                 <span class="image-overlay">@gallery.PhotoCount photo(s)</span>
23             </li>
24     }
25 </ul>
```

Intégration de fonctions de haut niveau en une seule ligne de code pour simplifier l'écriture des tâches courantes. Ces fonctions s'appellent des helpers et en voici une liste non exhaustive :

- **WebGrid** : affichage de grilles de données triables et paginables
- **WebMail** : envoi d'un email en une seule fonction
- **Recaptcha** : vous donne automatiquement un anti-spam graphique pour empêcher les robots de spammer votre site
- **Twitter** : affiche tous les micro-messages publiés par une personne ou sur un sujet donné.
- **WebImage** : vous permet de retravailler une image sans avoir à la stocker ou l'ouvrir.
- **Chart** : affichage de graphiques (camemberts, diagrammes en barres, etc.)
- **Analytics** : insère automatiquement le javascript de suivi Yahoo, Google, etc.
- **Bing** : ajoute la fonctionnalité de recherche à votre page
- **Json** : effectue le codage et le décodage au format JSON
- **Video** : insertion d'une video aux formats WMV, Flash ou Silverlight
- **WebCache** : gestion simplifiée du cache

Voici par exemple une page twitter IE9 avec le code source corres-

pondant : [Fig.7]. Ces fonctionnalités sont trop riches pour en faire le tour dans un article d'introduction, mais nous en reparlerons.

Pour qui ?

WebMatrix est un outil beaucoup plus simple de prise en main que Visual Studio, et ses fonctionnalités d'installation d'applications Open Source le mettent également à la portée des développeurs PHP même s'ils ne pratiquent pas ASP.Net. Cette simplicité a naturellement un prix (pas de fonctions d'intégration avec TFS, pas d'éditeur pour Entity Framework, pas de débogueur) qui nous permet de distinguer plusieurs cas d'utilisation de WebMatrix ou Visual Studio :

- Pour le développeur ASP.Net travaillant sur des projets complexes ou en équipe : Visual Studio est nettement préférable
- Pour retoucher une application Open Source : WebMatrix simplifie énormément l'installation et la prise en main des applications.
- Pour le développeur PHP : WebMatrix a un environnement d'édition bien conçu, mais ce sont surtout les fonctionnalités de publication qui sont intéressantes.
- Pour des développements très courts (un seul développeur) : la syntaxe très concise de Razor permet d'écrire plus rapidement certaines parties du site.

Pour la petite histoire, WebMatrix n'est pas le premier outil Microsoft utilisable en PHP : Expression Web a également cette fonctionnalité, même si elle est plus orientée vers l'édition HTML que vers le développement. WebMatrix est en bêta 3 depuis le 9 novembre, et devrait sortir en version finale dans les mois qui viennent. Si vous désirez creuser les fonctionnalités de Razor sans attendre le prochain article, vous pourrez trouver une série de tutoriaux en anglais (<http://www.asp.net/webmatrix>) et en français (<http://msdn.microsoft.com/fr-fr/asp.net/web-webmatrix.aspx>) sur les sites ASP.Net et MSDN.

■ Pierre Couzy

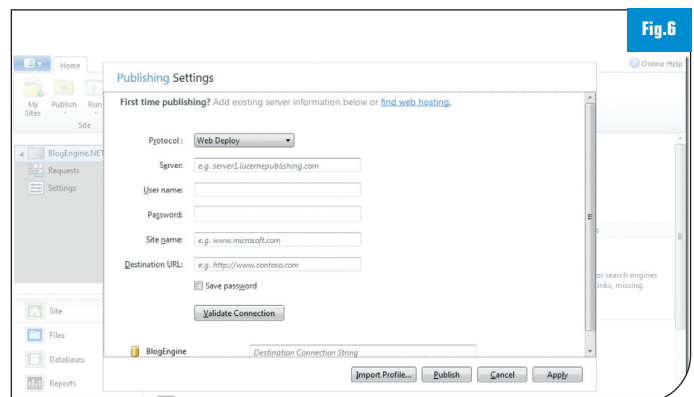


Fig.6

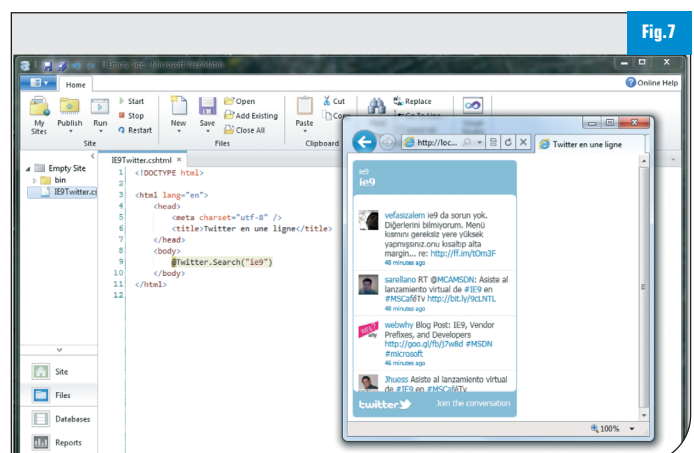


Fig.7

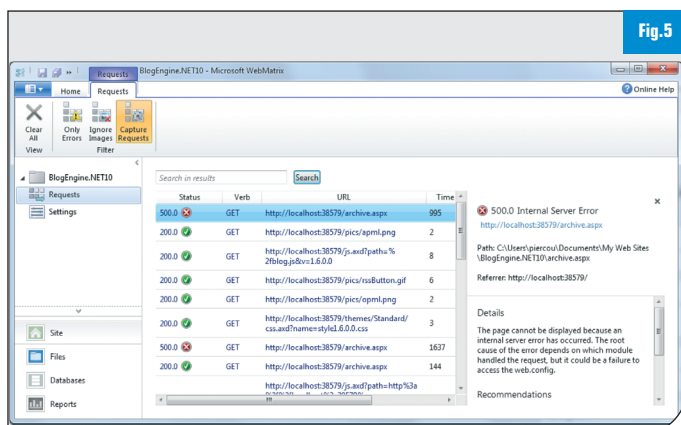
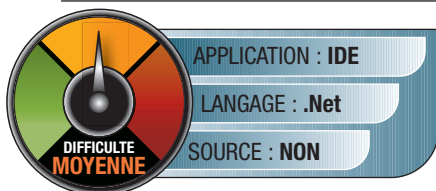


Fig.5

Visual Studio Lightswitch : IDE nouvelle génération ?

Depuis des décennies Microsoft n'a cessé d'innover dans de nouvelles technologies et de nouveaux outils de développement, afin de simplifier la tâche des développeurs. Visual Studio Lightswitch est-il une nouvelle étape ?



Pour les plus anciens, rappelez-vous dans les années 90, le développement Windows et la technologie COM n'était pas une mince affaire et ne se faisait qu'en C++ et

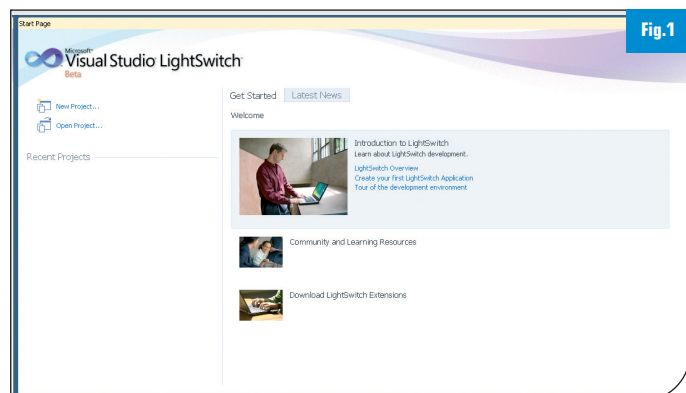
pratiquement à la main. Est arrivé Visual Basic, afin de démocratiser le développement sous Windows ou nous faisons du COM, comme monsieur Jourdain faisait de la prose. Même chose pour les bases de données, ou Microsoft a généralisé l'utilisation avec Microsoft Access. Plus proche de nous, Microsoft sort des bibliothèques de développement parallèle avec Visual Studio 2010, afin de rendre la tâche du développeur plus facile dans ce domaine.

Néanmoins, les outils et les technologies de Microsoft depuis l'arrivée de la plate-forme .NET, n'ont cessé de s'étendre et de se complexifier, au détriment de ceux qui ne souhaitent pas rentrer trop en profondeur dans les détails techniques et les architectures modernes, pour ceux pour qui une simple application présentant des données est suffisante.

Avec Microsoft Visual Studio Lightswitch, Microsoft comble un trou, en supprimant la complexité de la création d'application trois tiers en faisant des choix technologiques spécifiques pour vous. Vous n'avez plus qu'à vous concentrer sur les données et la manière de les présenter à l'écran.

ALORS QU'EST-CE QUE VISUAL STUDIO LIGHTSWITCH ?

Lightswitch est un moyen simple pour créer des applications d'entreprise de bureau et le Cloud. Lightswitch rend la création d'applications d'entreprise personnalisées, professionnelles pratiques et abordables grâce à des modèles et des outils prédéfinis dans un environnement de développement simplifié. Les applications créées avec Lightswitch peuvent facilement être déployées sur le bureau, sur un serveur web ou dans le Cloud.



En un mot Lightswitch est conçu pour simplifier et raccourcir le cycle de développement d'applications d'entreprise basées sur des données et des formulaires.

COMMENT ÇA MARCHE ?

Comme je le disais en introduction, Lightswitch supprime la complexité de développement d'application 3 tiers. Lorsque vous développez une application Lightswitch, vous développez de fait une application Internet riche 3 tiers, décomposée de la manière suivante :

- **Une couche présentation**, basée sur Silverlight 4.0 et le modèle MVVM (Modèle Vue Vue-Modèle)
- **Une couche logique** (middle tiers) basée sur les services RIA WCF, que l'on peut déployer en local pour une application de bureau, sur un serveur Web IIS, ou dans le Cloud sur Windows Azure
- **Une couche données** qui prend en charge SQL Server, SQL Express, les listes SharePoint, ou d'autres sources de données via des services WCF RIA.

Installation de Visual studio Lightswitch

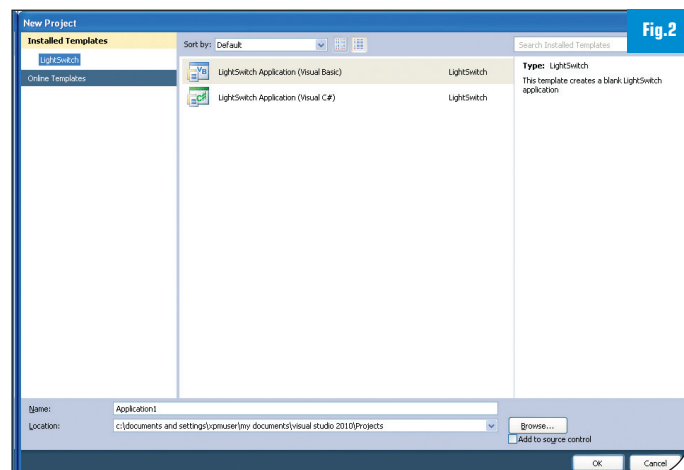
Si vous possédez déjà Visual Studio 2010, l'environnement de développement de Lightswitch sera directement disponible avec vos autres environnements. Dans le cas contraire il s'installera en autonome comme illustré sur la figure 1 et 2. [Fig.1]

Pour créer une application Lightswitch, rien de plus simple. Deux modèles sont disponibles, un modèle pour développer en Visual Basic, et un en C# [Fig.2].

Pour notre exemple, choisissons le modèle Visual Basic. L'assistant création d'une nouvelle table se lance [Fig3].

Créons notre 1re table Clients.

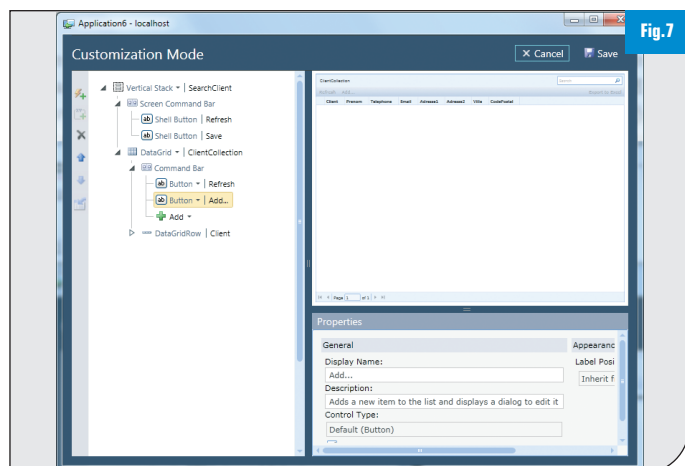
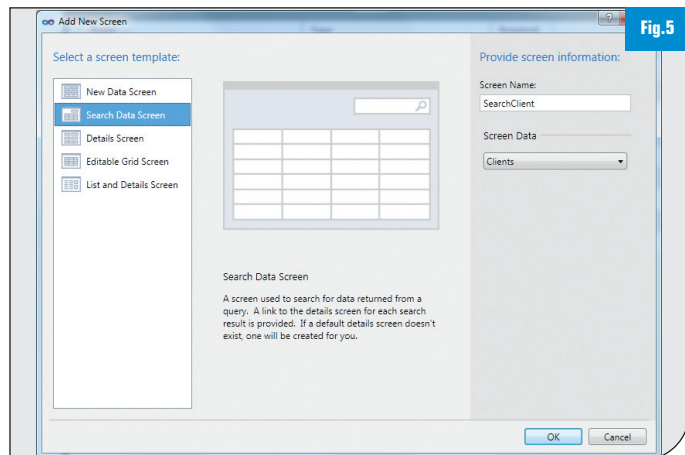
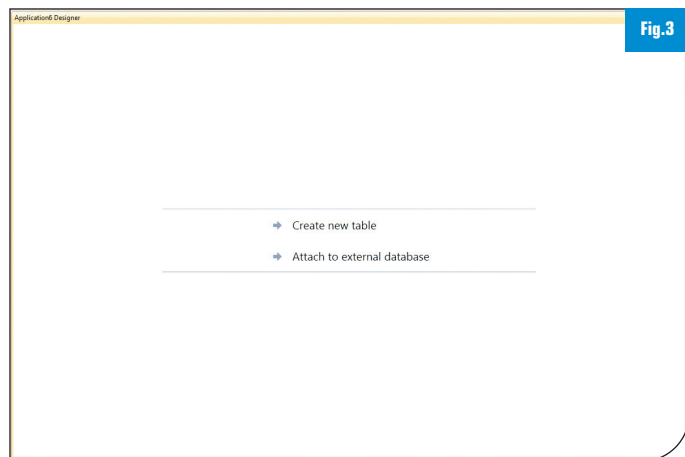
La création d'une table, se fait par l'intermédiaire d'un concepteur d'entités [Fig.4]. Notez sur la figure 4 que nous pouvons avoir des types courants String, et Int32 et autres, mais également des types



plus complexes, comme `PhoneNumber`, `EmailAddress` etc., qui pour ces derniers, peuvent proposer des formatages et de la validation de données par défaut.

Lors de la création de la table, vous créez non seulement la table physique **Clients** dans une base de données SQL Express, mais également l'entité **Client** au sens Entity Framework du terme. C'est-à-dire l'enveloppe de code qui permettra de manipuler l'entité et les données contenues dans la table physique. Toutes les API seront alors manipulables par code.

Maintenant nous allons associer un écran de recherche à notre table `Clients` [Fig.5]. Ici nous avons choisi le modèle **Search Data Screen**, mais notez qu'il en existe d'autres, fournissant d'autres fonctionnalités, comme la possibilité d'avoir des grilles éditables, ou des écrans Maître/Détails.

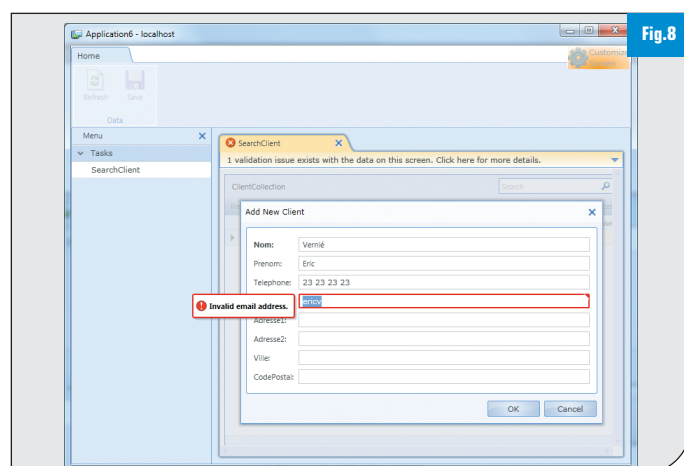
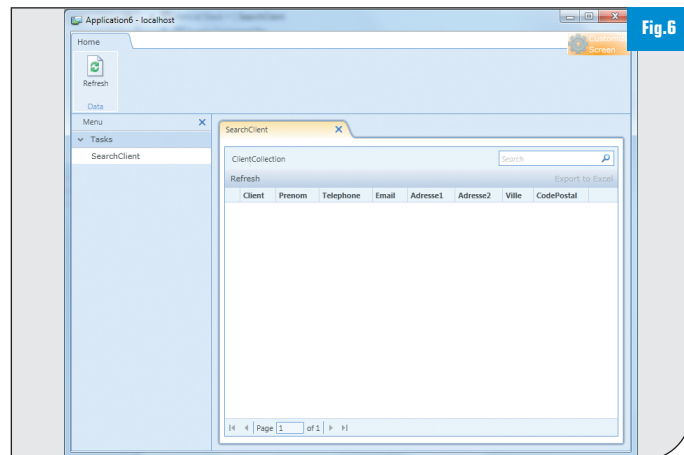
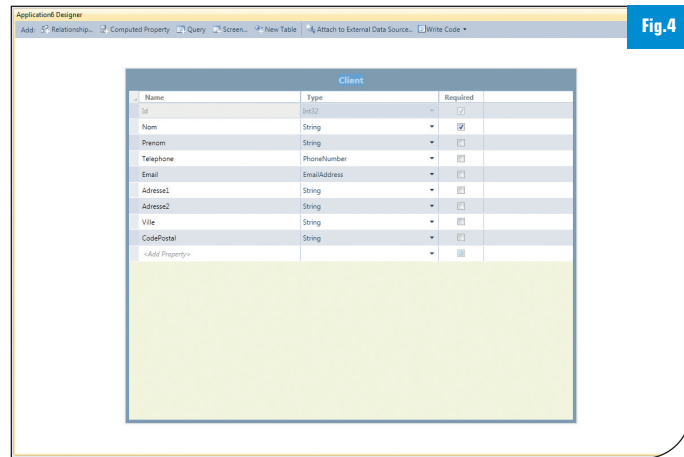


Et voilà, votre 1re application Lightswitch est prête à être exécutée. L'écran comporte une zone de recherche qui nous permet de filtrer sur les clients qui nous intéressent [Fig.6]. Le problème ici, c'est que dans cet écran de recherche, nous ne pouvons pas ajouter d'enregistrement ni les sauvegarder.

Avec Lightswitch, il est très simple d'ajouter ces deux fonctionnalités, en cliquant sur le bouton **Customize Screen**, puis en ajoutant un bouton **Save** sur le ruban (à côté du bouton `Refresh`) et un bouton **Add...**, comme illustré sur la [Fig.7].

Sur cette figure nous passons dans un mode de customisation, qui permet aux développeurs d'avoir immédiatement un rendu visuel à l'écran des contrôles qu'il peut ajouter ou positionner.

En ajoutant ces deux boutons, nous avons indiqué à Lightswitch, de nous fournir, non seulement un écran de saisie basé sur l'entité



Client [Fig.8], mais également toute la mécanique de sauvegarde des enregistrements et ceci sans écrire une seule ligne de code.

Notez que le type EmailAddress fournit une validation par défaut, et toujours sans écrire une seule ligne de code.

Mais en tant que développeur, nous aimons le code. Alors si vous souhaitez mettre les mains dedans, il est possible d'ajouter ses propres règles de validation, ou de travailler avec des propriétés calculées. Par exemple, dans le concepteur d'entité vous pouvez ajouter une nouvelle propriété calculée, qui concatène le nom et le prénom du client.

```
Private Sub NomCompleet_Compute(ByRef result As String)
    ' Set result to the desired field value
    result = Me.Nom + « , » + Me.Prenom
End Sub
```

[Fig.9]

Ou alors une règle de validation qui met en majuscule la ville du client lors de la validation

```
Private Sub Ville_Validate(ByVal results As EntityValidation
ResultsBuilder)
    ' results.AddPropertyError("<Error-Message>")
    If Not String.IsNullOrEmpty(Me.Ville) Then
        Me.Ville = Me.Ville.ToUpper()
    End If
End Sub
```

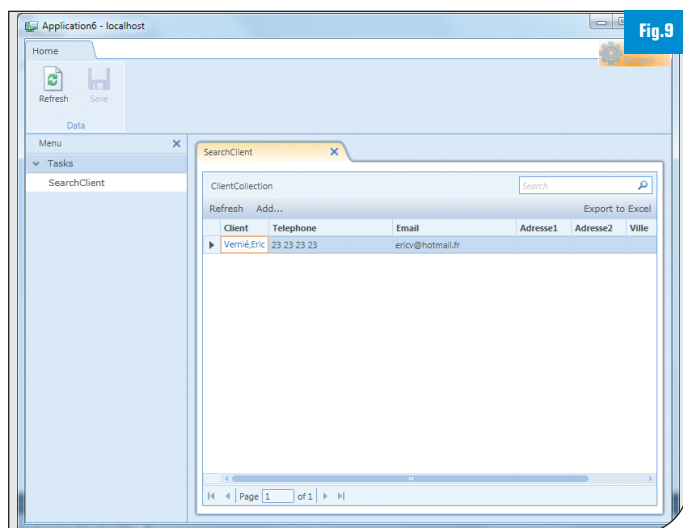


Fig.9

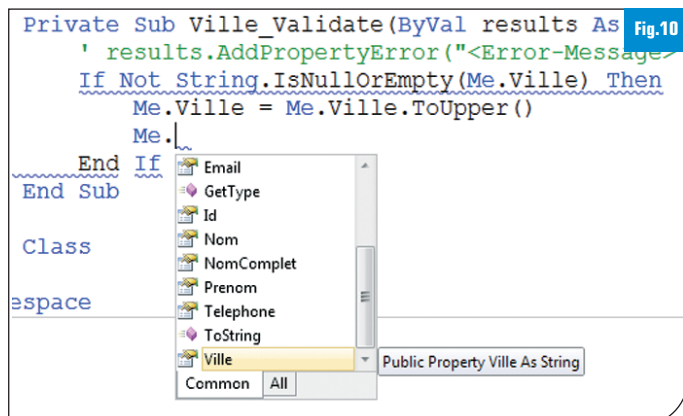


Fig.10

Comme je vous le disais plus haut, lorsqu'on crée une table, on crée également une entité, qui contiendra tout le code pour la manipuler, comme illustré sur la [Fig.10].

Lightswitch fournit également d'autres API, pour manipuler l'application, comme la possibilité de gérer des enregistrements, d'ouvrir d'autres écrans etc. Avec Lightswitch, il est également possible de construire des requêtes pour affiner la recherche, ou pour trier les enregistrements. Ces requêtes peuvent également accepter des paramètres.

Dans notre exemple, nous pouvons par exemple modifier la requête de l'écran de recherche, afin de trier les enregistrements par nom et prénom lors du chargement de l'écran [Fig.11].

ETENDRE LIGHTSWITCH

Jusqu'à présent nous avons développé une application avec Lightswitch, mais sachez qu'il est possible de développer des extensions pour Lightswitch. L'extension à Lightswitch, devient très utile lorsque l'application est largement diffusée, et que vous souhaitez rajouter des fonctionnalités qui ne sont pas prises en charge par défaut. Je ne rentre pas dans le détail dans cet article, mais sachez que vous pourrez trouver un tutorial consacré à ce sujet à cette adresse. <http://msdn.microsoft.com/fr-fr/vstudio/msdn.lightswitch.aspx>

Dans la figure 12, nous utilisons un contrôle Silverlight Bing Map (qui n'est pas dans Lightswitch) pour enrichir notre application [Fig.12]

CONCLUSION

En résumé, avec Lightswitch vous pouvez facilement développer des applications 3 tiers, basées sur des technologies telles que Silverlight 4.0 pour la partie présentation, Entity Framework pour le modèle de données, et les services RIA WCF pour la partie logique et middle tiers. Mais il est également possible de développer des extensions pour Lightswitch afin d'enrichir l'environnement de développement. Si vous souhaitez continuer l'exploration de Lightswitch, je vous encourage à faire le tutorial <http://msdn.microsoft.com/fr-fr/vstudio/msdn.lightswitch.aspx>, et à visiter le site officiel <http://blogs.msdn.com/b/lightswitch/> et le blog des développeurs <http://blogs.msdn.com/b/lightswitch/>



■ Eric Vernié
Microsoft France



Fig.11

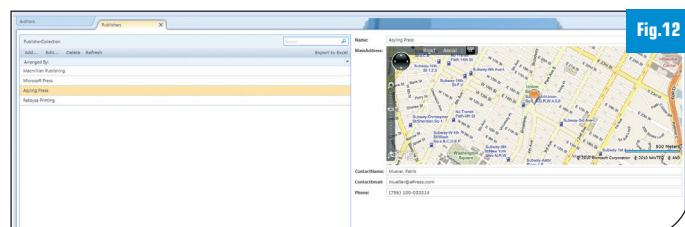
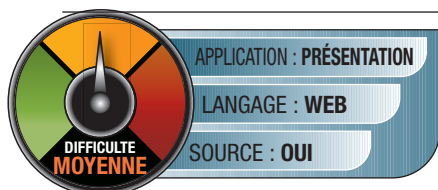


Fig.12

Silverlight Pivot : une autre manière de naviguer sur le web !

Qui n'a jamais rêvé de pouvoir interagir avec un volume de données important, d'une manière intuitive, simple et plutôt plaisante ? Quel développeur n'a jamais imaginé pouvoir fournir ce type de navigation à ses clients, pour quelques lignes de code ?



Et bien sachez que depuis quelques semaines, Microsoft a mis à disposition le **contrôle Silverlight Pivot** (<http://www.getpivot.com/>), qui

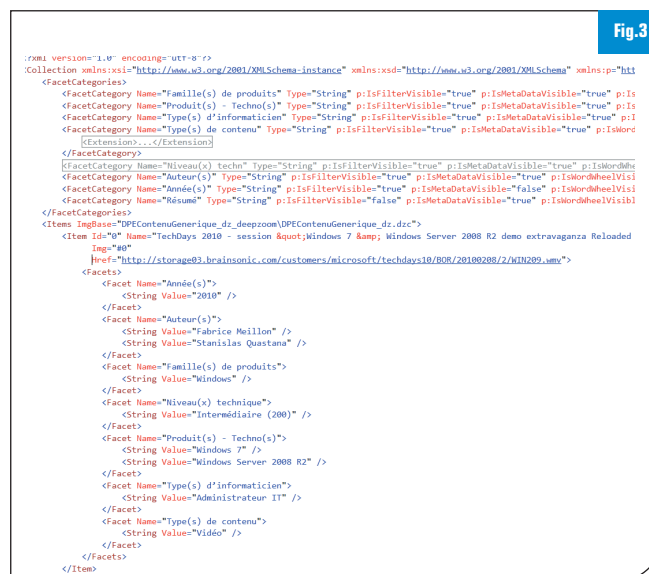
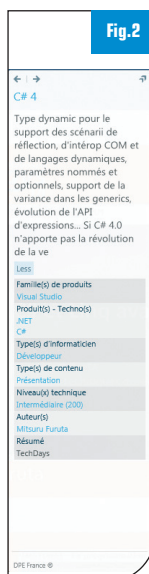
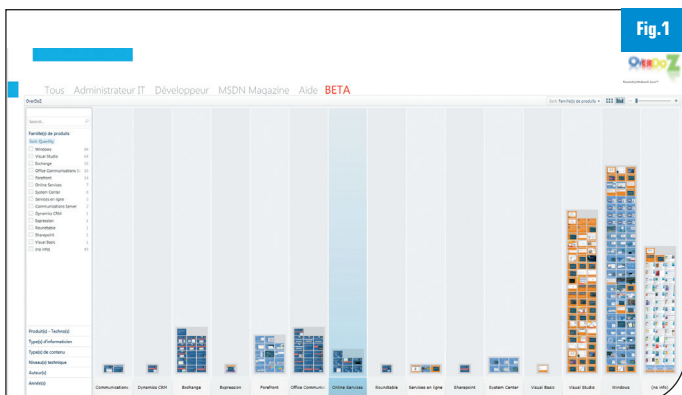
va permettre à tous les développeurs, de fournir une interface intuitive, afin de retrouver, visualiser, classer et analyser des données. Pour bien comprendre la suite de cet article, je suis parti d'un exemple, qui est une ébauche de portail sur le contenu Microsoft Francophone, (MSDN, Technet, microsoft.com/vision, etc.), et que vous pouvez tester à cette adresse <http://pico.blob.core.windows.net/bernard/Default.htm>

Une fois testé, que pouvons-nous constater ?

- Les informations sont affichées sous la forme d'une collection de vignettes, des images dans lesquelles nous pouvons naviguer en zoomant ou dé-zoomant. Il constitue l'**élément** de base de la collection.
- Il est possible de filtrer les données via des critères multiples. (Famille(s) de produits, Produit(s) Techno(s), etc.), qui sont appelés **Facets** (en anglais) dans le jargon de Silverlight Pivot.
- De faire de la recherche en saisissant un mot clé. Par exemple, tapez le mot clé *COACH* dans la boîte de recherche, cela affiche tous les coachs MSDN et Technet disponibles.
- Il est également possible d'afficher les données sous forme d'histogramme comme sur la figure suivante : [Fig.1].
- Lorsque vous sélectionnez un élément dans la liste, une fiche de résumé apparaît à droite de l'écran, qui reprend les facettes ainsi que la description associées à l'élément [Fig.2].
- En double-cliquant sur l'élément, vous sautez directement au contenu.

Pour bien comprendre le contrôle Silverlight Pivot, il y a deux étapes à appréhender :

- La création d'une collection de données.



- Feuille qui permet d'identifier la collection.
- La feuille **facet_categories** : [Fig.5]
- Feuille qui décrit les facettes que vous souhaitez comme multicritères. A noter que le champ `sort_values` permet de prédéfinir un ordre de tri.
- La feuille `items` : [Fig.6]

Dans cette feuille, il est impératif de bien reprendre tous les titres des facettes de la feuille **facet_categories**. Le champ **image**, doit contenir le chemin de l'image que vous souhaitez traiter avec les API DeepZoom.

Remarque :

Les images ici sont à créer manuellement, il peut être alors judicieux d'y accorder de l'importance quant à sa qualité (n'oubliez pas qu'il est possible de zoomer dans l'image en elle-même). Dans notre exemple, nous avons choisi un code de couleur, Bleu pour administrateur, Orange pour développeur, Blanc pour MSDN, afin que d'un seul coup d'œil la catégorie de l'élément soit accessible à l'utilisateur.

Enfin le champ **href**, comme son nom l'indique est un lien qui permettra à l'utilisateur de sauter directement sur le contenu qui l'intéresse. Une fois que le fichier Excel est rempli, et que les images sont créées, il faut utiliser l'outil PAuthor de la manière suivante.

Pauthor.exe /source excel MaCollection.xlsx /target deepzoom MaCollection_dz.cxml

Remarque :

Dans notre exemple, la collection est générée statiquement, car le nombre d'éléments ne dépasse pas les 3000. Dans le cas où ce nombre irait au-delà des 3000, il est de bon ton de générer la collection dynamiquement. Pour de plus amples informations, allez à l'adresse <http://getpivot.com/developper-info/>

Utilisation du contrôle Silverlight Pivot dans du code

L'utilisation du contrôle est assez simple pour peu que vous connaissiez déjà le développement Silverlight. Si tel n'est pas le cas, je vous encourage à suivre le coach Silverlight à l'adresse suivante <http://msdn.microsoft.com/fr-fr/silverlight/msdn.coachsilverlight.aspx>

Le contrôle est disponible à l'adresse suivante. <http://www.getpivot.com/download/>. Une fois installé, vous retrouverez dans le répertoire C:\Program Files (x86)\Microsoft SDKs\Silverlight\v4.0\PivotViewer\Jun10. Les binaires ainsi qu'un exemple de code qui vous permettra de démarrer avec le contrôle. Néanmoins, avant de pouvoir l'utiliser, un certain nombre de pré-requis sont à avoir :




Fig.4

A	B	C	D	E	F	G	H	I
name	icon	brand_image	additional_search_text	copyright_title	copyright_url			
OverDoZ				DPE France ©	http://msdn.microsoft.com/fr-fr/bb219575.aspx			

Fig.5

A	B	C	D	E	F	G	H
name	type	format	is_filter_visible	is_metadata_visible	is_vowdwheel_visible	sort_name	sort_values
Famille(s) de produits	String		true	true	true		
Produit(s) - Techno(s)	String		true	true	true		
Type(s) d'informaticien	String		true	true	true		
							Tutoriel (pas à pas)
							Article
							Présentation
							Vidéo
Type(s) de contenu	String		true	true	true		Exemple de code
							Débutant (100)
							Intermédiaire (200)
							Confirmé (300)
							Expert (400)
							Gourou (1500)
Niveau(x) technique	String		true	true	true		
Auteur(s)	String		true	true	true		
Année(s)	String		true	false	false		
Résumé	String		false	true	true		

Fig.6

A	B	C	D	E	F	G	H	I	J
name	image	description	href	related_links	Famille(s) de produits	Produit(s) - Techno(s)	Auteur(s)	Type(s) d'informaticien	Niveau(x) Technique
Gestion des traces : Partie II		Dans mon article précédent Gestion des traces avec Visual Basic 2005, je vous dans cet article, vous découvrirez comment créer un objet .NET avec Visual Basic 2005, et avec les API du Microsoft Common Language Runtime 2.0, vous pourrez étendre vos	http://msdn.microsoft.com/fr-fr/library/bb727315.aspx		Visual Studio	.NET Diagnostic	Eric Vernié	Développeur	Intermédiaire
Consommer un objet Visual Basic 2005 avec Visual Basic 6.0			http://msdn.microsoft.com/fr-fr/library/bb727303.aspx		Visual Studio	VB .NET CLR	Eric Vernié	Développeur	Intermédiaire
Les API de Hosting du Microsoft Common Language Runtime 2.0			http://msdn.microsoft.com/fr-fr/library/bb727295.aspx		Visual Studio		Eric Vernié	Développeur	Expert (400)

- Visual Studio 2010
- Web Platform Installer (<http://www.microsoft.com/web/downloads/platform.aspx>)
- Silverlight 4 Tools for Visual Studio 2010 (incluant le SDK et le Runtime de développement) <http://msdn.microsoft.com/fr-fr/silverlight/bb187452.aspx>
- Silverlight 4 Toolkit from April 2010 (<http://silverlight.codeplex.com/releases/view/43528>)

A partir de Visual Studio 2010, créez une nouvelle application Silverlight 4.0 en C# par exemple, mais cela fonctionne également avec Visual Basic. Incorporez les assemblages .NET (commençant par System.Windows.Pivot), qui se trouvent dans le répertoire C:\Program Files (x86)\Microsoft SDKs\Silverlight\v4.0\PivotViewer\Jun10\Bin

Le contrôle pivot, contient une méthode **LoadCollection**, qui prend comme paramètre l'adresse de la collection en elle-même.

Pour charger une collection, le code en est réduit à sa plus simple expression.

```
PivotViewer.LoadCollection(@"http://pico.blob.core.windows.net/bernard/DPEContenuGenerique_dz.cxml",String.Empty)
```

Remarque :

L'adresse ne sera bien évidemment pas inscrite en dur dans le code, mais sera à récupérer par le biais d'une chaîne d'initialisation à l'aide du paramètre `initparams`

Ce contrôle pivot implémente également l'événement `ItemDoubleClick` qui vous permettra de récupérer le lien de l'élément.

```
private void PivotViewer_ItemDoubleClicked(object sender, ItemEventArgs args)
{
    String linkUriString = cPivotViewer.GetItem(args.ItemId).Href;
    HtmlPage.Window.Navigate(new Uri(linkUriString, UriKind.RelativeOrAbsolute), "_blank");
}
```

Un exemple de code complet est disponible dans le répertoire C:\Program Files (x86)\Microsoft SDKs\Silverlight\v4.0\PivotViewer\Jun10\Source qui vous permettra d'aller plus loin.

CONCLUSION

Comme vous pouvez le constater il est facile grâce au contrôle Silverlight Pivot, de présenter des données de manière plus intuitive et attrayante. Il y a fort à parier que dans un avenir proche, nous verrons fleurir sur la toile des sites l'implémentant.

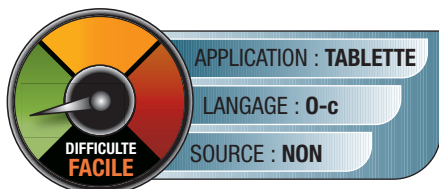
Pour de plus amples informations, n'hésitez pas à visiter le blog du team Silverlight <http://team.silverlight.net/announcement/start-preparing-for-the-silverlight-pivot-control-now/>

■ Eric Vernié

Relations Techniques Développeurs (Microsoft France)

iPad : qu'est-ce qui change pour le développeur ?

La tablette tactile d'Apple est désormais disponible partout dans le monde. Comme pour l'iPhone il y a 3 ans, l'iPad jette un pavé dans la mare d'un marché jusqu'ici balbutiant : les tablettes tactiles. On peut même y voir les prémices de l'ordinateur du futur : réduit à sa plus simple expression : un écran, si naturel qu'un enfant peut s'en servir et en même temps assez puissant pour répondre à tous les usages de la vie courante: mails, Internet, vidéos etc.



A mi-chemin entre le Mac et l'iPod au niveau de son écran et des usages, l'iPad est pourtant beaucoup plus proche de l'iPhone aussi bien au niveau matériel que

logiciel. Je vous propose de faire le point dans cet article sur les différences majeures et les impacts techniques qui en découlent.

La plateforme

Premier constat, au niveau logiciel l'iPad utilise le même système d'exploitation que l'iPhone, à savoir iOS (auparavant iPhoneOS). C'est en fait très logique : l'iPad est lui aussi basé exclusivement sur une interface multi-touch, ses ressources matérielles sont assez limitées et sont mieux exploitées par un système d'exploitation créé spécifiquement pour un usage mobile.

Gros avantage, l'iPad est donc compatible avec toutes les applications écrites pour l'iPhone et bénéficie dès sa sortie d'un gigantesque catalogue. Cela permet également à Apple de pérenniser sa relation avec les développeurs iPhone en leur offrant de nouveaux territoires à conquérir. Ces derniers seront immédiatement à l'aise avec l'iPad puisque le langage, les outils et même l'AppStore restent. La seule différence notable est finalement que l'iPad et l'iPhone utilisent deux branches différentes d'iOS (3.2 et 4), en clair, les nouveautés d'iOS 4 comme le multitâche ne sont pas encore disponibles pour l'iPad. Ces deux branches seront normalement unifiées dans une prochaine version à l'automne.

Universal Binary

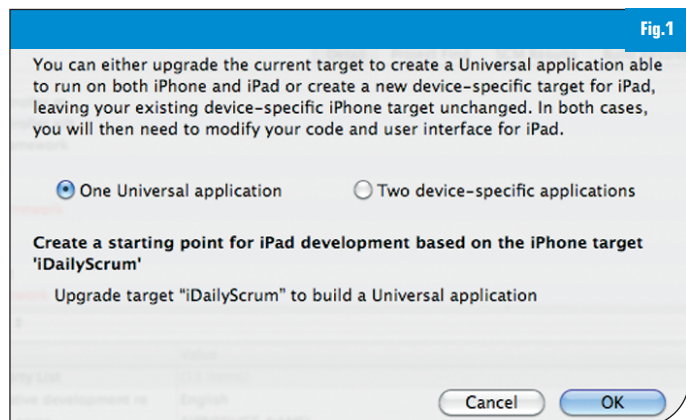
Si vous commencez demain votre nouvelle application iPad, le processus restera inchangé par rapport à l'iPhone : Xcode propose plusieurs assistants pour générer le squelette du projet.

En revanche, une question de taille se pose si vous avez déjà un existant iPhone et que vous souhaitez uniquement l'enrichir d'une version iPad, sans pour autant dupliquer toute la logique métier. Apple propose et encourage deux manières de procéder :

- Garder un seul projet Xcode mais avoir deux *targets*, une pour chaque plateforme. Cette technique pouvait déjà être utilisée pour générer plusieurs applications iPhone ayant beaucoup de code commun.
- Créer un *Universal Binary*, c'est-à-dire un seul binaire capable de s'exécuter sur chacune des plateformes.

Pour le côté mise en œuvre, il suffit d'ouvrir un projet iPhone existant, sélectionner la target et choisir dans menu *Project > Upgrade Current Target for iPad*, les deux stratégies sont alors proposées [Fig.1]. Comment choisir ? Les deux techniques se valent, chacune apportant ses avantages spécifiques. La première permet de ne dis-

Fig.1



socier que les *ViewControllers* et les Nibs en réutilisant tout l'existant (ressources, logique métier) et permet de promouvoir une version HD de l'application que les utilisateurs devront racheter. Le binaire universel est lui plus simple à distribuer, plus logique du point de vue utilisateur et permet même de mutualiser les *ViewControllers*. En revanche il demandera plus de tuyauterie dans le code : il faut pouvoir distinguer sur quel appareil on se trouve et réagir en conséquence, par exemple :

```
if (UI_USER_INTERFACE_IDIOM() == UIUserInterfaceIdiomPad) {
    // iPad
}
else {
    // iPhone, iPodTouch
}
```

Les nouveaux éléments de l'interface

Jusqu'ici nous n'avons pas réellement rencontré de différence profonde, plutôt des évolutions. C'est en revanche du côté de l'interface graphique où l'on assiste à des changements importants avec l'introduction de deux nouveaux éléments dans UIKit : *UISplitViewController* et *UIPopoverController*.

Sur iPhone, le contrôleur de navigation *UINavigationController* était la voie royale pour présenter de l'information hiérarchique en permettant de n'afficher qu'un seul niveau à la fois. L'idée était d'optimiser au mieux l'espace disponible, mais cette limitation n'a plus lieu d'être sur iPad d'où l'ajout d'un contrôleur permettant d'afficher une vue maître détail sur un seul écran.

Le *UISplitViewController* devient donc la classe par défaut pour la majeure partie des applications, il est toujours possible d'utiliser un *UINavigationController* mais plus comme contrôleur de plus haut niveau. Les applications Préférences et Mail sont deux bons exemples de l'utilisation de cette classe [Fig.2].

Qui dit grand écran dit également le retour d'éléments graphiques plus classiques de nos chères applications desktop : les palettes ou fenêtres flottantes, appelées ici «pop over». La seule différence étant que ces fenêtres disparaissent dès que l'on clique hors de leur zone pour laisser le maximum d'espace pour le contenu principal. Par défaut un *UIPopoverController* fait 320 pixels de large, ce qui permet de réutiliser rapidement un Nib d'une application iPhone.

Plus d'interactions

On sent qu'Apple a voulu faciliter les interactions

entre un utilisateur et l'iPad, par exemple il est fortement encouragé de supporter toutes les orientations. Ceci n'était pas gênant pour un iPhone car on le tient dans une seule main, mais sur un iPad, chan-

ger l'orientation à tout bout de champ peut devenir extrêmement pénible. Du côté du multi-touch une nouvelle API fait son apparition : *Gesture Recognizers* qui permet de reconnaître de façon simple différents gestes auxquels nous sommes maintenant habitués, par exemple *pinch* (zoom), *swipe* (mouvement latéral), la rotation ou encore les appuis longs.

C'est une excellente nouvelle puisqu'auparavant il pouvait être fastidieux de reconnaître ces différents mouvements à partir des coordonnées. Voici comment reconnaître très facilement un *pinch* sur une vue :

```
UIPinchGestureRecognizer *pinchGesture = [[UIPinchGestureRecognizer alloc] initWithTarget:self action:@selector(handlePinch:)];
[self.view addGestureRecognizer:pinchGesture];
[pinchGesture release];
```

Ces nouveautés visent à ce que les applications iPad puissent pleinement tirer parti du multi-touch.

Une API pour les documents

On a beaucoup reproché à Apple le côté fermé de la plateforme, et bien cela change ! Pas aussi vite que certains le voudraient, mais chaque application iPad peut maintenant partager et échanger des documents au travers d'un répertoire partagé. La grosse limitation reste cependant l'utilisation obligatoire d'iTunes lorsque l'iPad est connecté à un ordinateur pour échanger des documents.

Autre changement, une application peut se déclarer dans le système (via Info.plist) comme capable d'ouvrir certains types de fichiers. Lorsque Mail par exemple ouvre une pièce jointe, on peut dorénavant choisir de l'ouvrir directement dans l'application concernée [Fig.3].

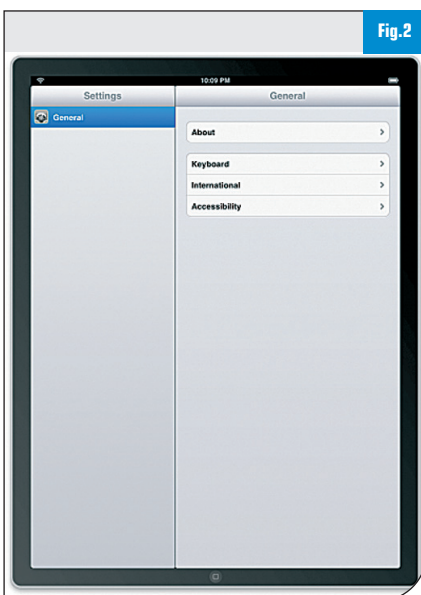
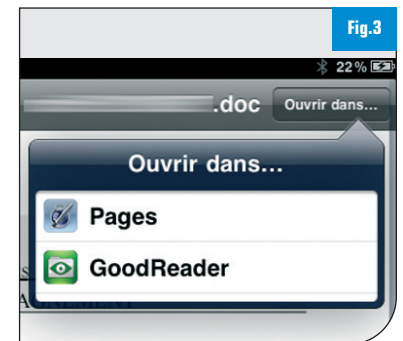
Conclusion

Voilà en quelques paragraphes les changements les plus importants introduits par l'iPad et pour la plupart portés sur le nouveau iOS 4. Vous l'aurez compris, d'un point de vue technique l'iPad diffère assez peu. En revanche les différences sont majeures dans la philosophie des applications : l'iPhone est un terminal pour la consultation, l'iPad pour la consultation ET la production.

C'est criant si l'on met bout à bout les évolutions : un large écran, les palettes (pop over), les interactions facilitées, l'export de documents etc. Il est clair qu'Apple mise sur ce nouveau terminal et sur les nouveaux usages que nous saurons en faire. Pour la petite histoire cet article aura évidemment été écrit sur un iPad !

■ Olivier Martin

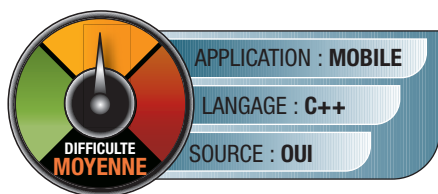
Responsable de l'offre Mobilité au sein d'OCTO Technology, cabinet d'architectes en systèmes d'information (<http://blog.octo.com>, <http://universite-du-si.com>)



Découverte de Samsung Bada

1^{re} partie

Samsung a lancé cette année son OS mobile dénommé Bada (signifiant en coréen océan) afin d'une part de remplacer le système propriétaire installé sur tous les téléphones mobiles Samsung classiques (encore appelé « feature phones ») pour les transformer en Smartphones, et d'autre part d'équiper avec leur propre OS leurs Smartphones. Voici la première partie d'un article qui va vous emmener à la découverte de ce nouvel OS pour vous mettre ensuite le pied à l'étrier.



Bada se décompose en quatre parties :

- Le noyau qui peut être un OS temps réel mais aussi Linux indépendamment du

téléphone (le S8500 utilise du code FreeBSD, NetBSD et OpenBSD)

- La couche device fournit les fonctionnalités de base comme la téléphonie, la gestion du multimédia, la sécurité, le moteur graphique
- La couche Service permet des fonctionnalités plus élaborées travaillant notamment en réseau avec le serveur Bada pour fournir des données, par exemple la cartographie, la connexion aux réseaux sociaux.
- Enfin la couche framework expose une API pour les développeurs afin de construire des applications et accéder aux fonctionnalités de plus bas niveau.

Les applications Bada peuvent être de trois types : natives, service et web. La première catégorie est celle que nous allons investiguer dans cette série d'articles. La seconde vise à fournir des applications exposant des services aux autres applications. La dernière permet d'écrire des applications en HTML et même en Flash ce qui différencie beaucoup Bada de iOS et abaisse considérablement la barrière d'entrée pour faire des applications sur la plateforme.

LES APPLICATIONS

Les applications natives sont écrites en C++ et Samsung a réalisé un IDE basé sur Eclipse et CDT et fournissant un SDK assez complet pour développer des applications comprenant un designer d'interface (plutôt basique), et permettant de :

- compiler l'application en faisant l'édition de lien sur les bibliothèques Bada,
- lancer l'application en mode simulation et la déboguer,
- installer l'application sur le matériel cible,
- créer un package pour le déploiement sur le store Samsung.

Malheureusement cet IDE ne fonctionne que sous Windows et est de plus assez gourmand en ressources, ce qui empêche quasiment de l'installer et s'en servir dans une VM de façon satisfaisante...

Mais en contrepartie, Samsung offre un générateur d'événements très bien fait qui permet de simuler un grand nombre de paramètres de l'environnement du téléphone comme le géo-positionnement du téléphone, la bande passante, le niveau de la batterie, la connexion d'un câble USB et bien d'autres choses encore afin de tester de façon assez exhaustive l'application sans téléphone.



LES POSSIBILITÉS ET PARTICULARITÉS

Voici donc un tour exhaustif de ce dont un développeur, qui souhaite réaliser une application, peut disposer pour construire son application :

- **Osp::Base** contient les types de données élémentaires (Object, String, ...) et les wrappers des types primitifs. Il contient aussi ce qu'il faut pour gérer les timers et les threads, les collections et un ensemble de classes utilitaires comme Math, StringUtils, StringTokenizer. Il supporte également un sous-ensemble de newlib et la librairie C++ STL.
- **Osp::Io** contient des API de manipulation de systèmes de fichiers mais aussi de bases de données et surtout un composant central, le registry, qui sert à stocker tous les réglages et les préférences utilisateur.
- **Osp::Text** et **Osp::Locale** contiennent les API pour manipuler l'encodage du texte et la locale.
- **Osp::System** contient les API pour obtenir des informations de la part du kernel (mémoire, uptime, niveau de la batterie, ...) et contrôler certains éléments matériels (alimentation de l'écran, le vibreur).
- **Osp::App** est fondamental pour les applications. Il permet la gestion du cycle de vie d'une application. Comme au début d'iOS, Bada n'autorise le lancement que d'une application à la fois mais autorise le lancement d'agents sans interface graphique. Il permet également à une application de consommer les opérations exportées par les autres applications comme par exemple l'accès au calendrier, au carnet d'adresses ou encore la galerie médias.
- **Osp::Graphics** regroupe toutes les opérations de dessin qu'elles soient 2D (vecteurs et bitmaps) ou 3D grâce à OpenGL ES et EGL.
- **Osp::Ui** est également très important puisqu'il regroupe tous les contrôles graphiques et autres conteneurs disponibles dans l'OS. C'est également dans ce namespace que l'on retrouve des composants comme FlashLite, les animations, les cartes, les contrôles web mais aussi la possibilité de décrire une interface en XML à la manière de ce que fait Google avec Android. A noter que les

contrôles sont indépendants de la résolution du téléphone pour pouvoir être utilisés dans différents matériels sans avoir à réécrire l'application de façon spécifique. Ce point est assez épineux et toujours plus ou moins problématique. La plupart du temps les belles promesses ne sont pas tenues...

- **Osp::Uix** permet d'accéder à un bon nombre de fonctionnalités plus évoluées comme les différents capteurs (accéléromètres, orientation, boussole, proximité, GPS...), le service météo, les « motion gestures ». Ce namespace contient en outre une fonctionnalité différente par rapport aux autres OS : la reconnaissance de visages. C'est sur ces API que se basera le prochain article qui abordera la réalisation d'une application Bada.
- **Osp::Net** regroupe toutes les fonctionnalités d'accès au réseau que ce soit à haut niveau (http, DNS, WiFi) ou à bas niveau (TCP, UDP, Bluetooth). A savoir que pour le bluetooth, l'OS est bien moins restrictif que sur iOS puisque les profils GAP, OPP et SPP sont supportés.
- **Osp::Telephony** permet d'accéder au statut d'appel mais aussi de récupérer beaucoup d'informations concernant l'opérateur, les informations de la carte SIM, ... Ceci va encore une fois plus loin que sur d'autres OS mais cela ne posera-t-il pas des problèmes de privauté ?
- **Osp::Messaging** regroupe les opérations d'envoi et réception de messages au sens large : SMS, MMS mais aussi email et messages push. A noter que si la fonctionnalité de push était prévue, elle a été retirée par Samsung récemment. Samsung rencontrait-il des problèmes de mise en échelle ou de protection des données ?
- **Osp::Media** permet la lecture et l'enregistrement de media audio et vidéo
- **Osp::Security** permet la gestion des certificats, des clés et des algorithmes de cryptage
- **Osp::Social** est un autre namespace très novateur puisqu'il constitue une couche d'abstraction entre le téléphone et les informations contenues dans les réseaux sociaux. Un serveur Bada est pour cela mis à contribution pour servir d'adaptateur entre le téléphone et la diversité des réseaux. Pour le moment Twitter, Facebook et MySpace sont supportés.
- **Osp::Content** permet la gestion et la recherche dans le contenu stocké sur le téléphone mais aussi sur le serveur Bada. Il s'agit ici essentiellement de rechercher et modifier les métadonnées associées aux contenus quels qu'ils soient (audio, vidéo, image, ...)
- **Osp::Locations** fournit les informations liées au positionnement (relevé de position et point d'intérêts mais aussi service de traçage d'itinéraire, rendu de cartes...) A noter que Samsung a décidé de s'appuyer sur deCarta pour le rendu des cartes, geocoding, et autres services d'itinéraires, contrairement à Apple et bien sûr Google. Cependant, ces services sont capricieux et demandent encore des efforts de la part de Samsung pour bien fonctionner.

• **Osp::Commerce** offre des services d'achat directement par le téléphone comme le fait Apple avec son inAppPurchase.

• **Osp::Web** permet d'embarquer une page web dans une application et d'utiliser les différents contrôles associés (historique, téléchargement) ainsi que l'évaluation de Javascript.

Il y aurait bien plus à dire sur chaque namespace mais passons aux choses sérieuses et essayons de faire notre première application !

CRÉATION D'UN HELLO WORLD

Rien de tel pour plonger dans la programmation d'applications Bada que de commencer avec un HelloWorld. Notre application comportera deux contrôles : un contrôle EditField, qui permettra à l'utilisateur de choisir un texte à saisir et un contrôle Button, qui lui permettra de lancer une MessageBox dans lequel s'affichera le texte précédemment saisi par l'utilisateur.

Pour créer un projet Bada faites : Files > New > bada Application Project. L'assistant nous permet de paramétrer notre application en six étapes.

La première étape nous propose de choisir le nom et le type de projet. Choisissez comme nom de projet HelloBada et comme type projet : Bada Frame Based Application. La seconde étape permet de choisir le LP (pour Language Pack), choisissez le LP1, il inclut les principales langues (anglais, français, allemand, ...). La troisième étape consiste à choisir le manifest de l'application (nous expliquerons plus tard le but de ce fichier). La quatrième étape : ne choisissez pas auto-scaling (par défaut). Cette fonctionnalité permettra à l'application de choisir sa résolution automatiquement. Pour notre exemple nous n'en aurons pas besoin. Pour la cinquième étape choisissez vous-même vos informations. La dernière étape consiste à choisir les différentes configurations pour le débogueur, laissez le choix par défaut.

Dans un premier temps nous allons créer le premier formulaire, pour ce faire nous utiliserons UI Builder. L'UI Builder est un WYSIWYG permettant de réaliser les interfaces en XML.

Dans Eclipse, faites : Window > Show View > Other > Bada > Resource Dans cette vue, sélectionnez la résolution 480x800. Faites un clic droit sur Forms, sélectionnez « Insert Resource ». Cliquez sur le formulaire, recherchez la vue Properties, changez l'ID par IDC_FORM_HELLOWORLD.

Sur votre droite dans la toolbox, glissez-déplacez le composant Button. Changez l'ID de celui-ci par IDC_BUTTON_HELLO. Profitez-en pour changer la propriété Text par Hello.

Comme pour l'étape précédente glissez-déplacez un composant EditField et changez l'ID en IDC_EDITFIELD_WORLD.

Votre vue est maintenant créée, nous pouvons dès à présent générer la classe correspondante : clic droit sur la vue du formulaire > Add class.

Choisissez comme nom de classe FORM_HELLOWORLD.

Cochez l'EventListener IActionEventListener. Celui-ci permettra à notre classe d'implémenter la méthode OnActionPerformed et ainsi



Fig. A

Une fois notre classe générée, dans le fichier header, nous allons pouvoir déclarer en `protected` deux attributs qui seront des pointeurs sur nos composants.

```
protected:
    Osp::Ui::Controls::Button*    __pButtonHelloWorld;
    Osp::Ui::Controls::EditField* __pEditFiedHelloWorld;
    //Action ID
    static const int ID_BUTTON_HELLOWORLD = 100;
```

Puis nous ajoutons l'évènement à notre bouton et nous définissons son `ActionId`

```
Code Result FORM_HELLOWORLD::OnInitializing(void){
    result r = E_SUCCESS;

    // Get resources form XML

    __pButtonHelloWorld = static_cast<Button *>(GetControl(L» IDC_
_BUTTON_HELLOWORD»));

    __pEditFiedHelloWorld = static_cast<EditField *>(GetControl
(L» IDC_EDITFIELD_WORLD»));

    // Add event on the button

    if (__pButtonHelloWorld != null){
        __pButtonHelloWorld->SetActionId(ID_BUTTON_HELLOWORLD);
        __pButtonHelloWorld->AddActionEventListener(*this);
    }

    return r;
}
```

Pour traiter l'action du bouton, nous implémentons l'interface `Osp::Ui::IActionEventListener` qui ne possède qu'une seule méthode (le listener) `OnActionPerformed`.

```

Void FORM_HELLOWORLD::OnActionPerformed(const Control& source,
int actionId){
    //On choisit le traitement en fonction de l'action id
    switch(actionId){
        case ID_BUTTON_HELLOWORLD:{
            MessageBox messageBox;
            messageBox.Construct(«Hello», __pEditFiedHelloWorld->Get
Text(), MSGBOX_STYLE_OK, 0);
            int messageBoxResult;
            messageBox.ShowAndWait(messageBoxResult);
        }
        break;
        default:{
            AppLog(«Impossible de déterminer l'actionId»);

```



Pour terminer, et afin de lancer, au démarrage, votre formulaire FORM_HELLOWORLD, allez dans le fichier HelloWorld.cpp et remplacez la méthode OnAppInitializing par :

```

    Bool HelloWorld::OnAppInitializing(AppRegistry& appRegistry){
        FORM_HELLOWORLD* formHelloWorld = new FORM_HELLOWORLD();
        formHelloWorld->Initialize();
        // Add the form to the frame
        Frame *pFrame = GetAppFrame()->GetFrame();
        pFrame->AddControl(*formHelloWorld);
        // Set the current form
        pFrame->SetCurrentForm(*formHelloWorld);
        // Draw and Show the form
        formHelloWorld->Draw();
        formHelloWorld->Show();
        return true;
    }
}

```

Vous pouvez dès à présent compiler et lancer votre application [Fig.A].

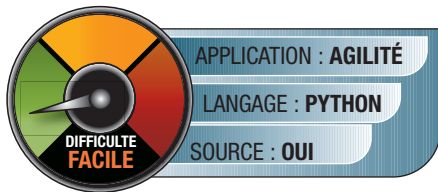
■ **Fabrice Dewasmes**, *Mobile Department Director chez Pragma Consult, Luxembourg*

■ Alexis **Détrie**, élève ingénieur, SUPINFO Metz

Pragma Consult a réalisé l'une des 90 applications sélectionnées par Samsung dans le cadre du concours mondial de développement Samsung Bada. Pragma Consult concourt dans la catégorie navigation avec l'application Doogies. Plus d'infos sur <http://developer.bada.com>

Pylons, un nouvel outil de développement Web agile

Il présente beaucoup de similitudes avec Ruby on Rails, mais il est basé sur Python. Nous découvrons aujourd'hui Pylons, un outil de développement Web agile.



Dans le monde du développement Web, la réactivité est fondamentale. Pour cette raison, nombreux sont les gestionnaires de contenu ou les frameworks de développement agile. Un des plus populaires est Ruby on Rails (RoR), basé, comme son nom l'indique, sur le langage Ruby. Programmez! N° 98 vous a présenté Django, un framework similaire à RoR, mais basé sur le langage Python. Aujourd'hui nous nous intéressons à un autre framework open source basé sur Python et qui vient d'être publié en version 1.0: Pylons. Lui aussi très similaire à RoR, Pylons présente des spécificités que le rendent très intéressant.

1 UN APERÇU DE PYLONS

Pour le programmeur, Pylons est une architecture Modèle-Vue-Contrôleur ou MVC. Ceci impose une démarche au programmeur et l'oblige à obéir à une logique bien définie. L'intention étant que, une fois cette démarche bien assimilée, le programmeur ne se concentre que sur ses buts et ne perde pas de temps quant à la façon d'y parvenir. Travailler vite et bien sans perte de temps est une philosophie que les concepteurs de Pylons ont su aussi appliquer à eux-mêmes avec bonheur. Ainsi, plutôt que de réinventer la roue, ils ont construit ce framework en s'appuyant sur les meilleures bibliothèques existantes: WSGI, un middleware positionné entre une application Web et le serveur, Mako, une bibliothèque de modèles (templates), SQLAlchemy, un mapper Objet-Relationnel, Routes, une bibliothèque de routage de requêtes http, etc. Bien que les composants soient nombreux, l'ensemble donne une bonne impression d'homogénéité. En outre, Pylons est plutôt bien documenté. Mieux que la majorité des logiciels open source, avec toutefois quelques petits problèmes de mise à jour de la documentation au rythme des évolutions de Pylons. Voyons tout cela par la pratique en prenant l'outil en main.

2 HELLO WORLD

Tradition oblige :-). Commençons par faire un Hello World similaire à celui proposé par la documentation, puis nous l'enrichirons afin de mettre en évidence certaines particularités insuffisamment signalées par la documentation, à mon humble avis. Comme nous l'avons dit Pylons fait très grandement appel à des modules tiers. Pour éviter des conflits avec des modules installés avec votre Python usuel, Pylons vous propose de créer un environnement Python virtuel, une sorte de Sandbox, dans lequel seront installés Pylons et ses nombreux modules. Cette idée d'environnement Python virtuel est excellente et d'ailleurs rien ne s'oppose à la reprendre pour d'autres propos. Pour mettre en place cet environnement virtuel, on utilise le script Python go-pylons.py en lui passant un nom de répertoire qui

contiendra cet environnement virtuel. Par exemple :

```
python go-pylons.py c:\mypythondev
```

Il est bien évident que Python doit au préalable être installé sur votre système. Une version 2.4 ou supérieure est requise. Le script go-pylons est disponible sur le site officiel ou sur notre site, conjointement aux exemples de cet article. Une fois l'exécution du script terminée, il suffit de se positionner dans le sous-répertoire Scripts du répertoire précédemment défini et d'activer l'environnement :

```
source ./activate (Linux) ou .\activate.bat (Windows)
```

Tout est prêt. Mieux, le script a installé les setuptools, ce qui vous permettra d'ajouter à l'environnement toute bibliothèque Python qui vient sous la forme d'un Egg via la commande easy_install, disponible dans le même répertoire Scripts. Nous pouvons maintenant créer le squelette de notre application avec *paster*, l'outil couteau-suisse de Pylons

```
paster create -t pylons helloworld
```

Cette opération va créer toute une arborescence. Les noms des répertoires parlent d'eux-mêmes. Le répertoire config mérite toute votre attention. Pour lancer l'application, il suffit de positionner dans le répertoire racine de celle-ci. Ensuite un petit coup de *paster* fait l'affaire :

```
cd helloworld
paster serve --reload development.ini
```

Si tout va bien, vous obtenez un message indiquant que Pylons écoute le port. Sous réserve que votre pare-feu laisse le port 5000 ouvert, l'URL :

```
http://localhost:5000/
```

vous amène sur la page d'accueil de votre application qui est une page statique.

3 TRAVAILLER AVEC L'ARCHITECTURE MVC

En programmation MVC, le contrôleur se positionne entre les données et la vue qui les représente. Il reçoit les événements de l'utilisateur et effectue les actions correspondantes. Avec Pylons, les événements sont à la base des urls saisies dans le navigateur client et routés vers le(s) contrôleur(s) via le mécanisme de routage. Avec Pylons le code des contrôleurs doit être déposé dans le répertoire dédié \$AppName/controllers, c'est-à-dire pour notre exemple, dans le répertoire helloworld/controllers qui le trouvera dans le répertoire racine helloworld. Voici le code du contrôleur le plus rudimentaire possible (fichier hello.py) :

```
import logging

from pylons import request, response, session, tmpl_context as c, url
```



```
from pylons.controllers.util import abort, redirect

from helloworld.lib.base import BaseController, render

log = logging.getLogger(__name__)

class HelloController(BaseController):
    def index(self):
        return 'Programmez!'
```

Ce contrôleur entre en action quand l'utilisateur donne cette URL dans le navigateur :

```
http://localhost:5000/hello/index
```

La méthode `index` du contrôleur est invoquée grâce au mécanisme de routage par défaut qui a été mis en place par le `paste` lors de la création de l'application. Pour que tout fonctionne bien, il est important de respecter une convention de nommage. La classe d'un contrôleur auquel on accède par le terme `hello` doit avoir comme nom `HelloController`. C'est logique. Cependant la documentation ne précise pas que des majuscules sont malvenues dans le nom de classe. Ainsi voici une erreur qui peut arriver facilement lorsqu'on débute en Pylons et qui peut énerver : on veut travailler par exemple avec une url `/helloworld/index` et on dénomme la classe `HelloWorldController`. Ceci ne fonctionnera pas, et sans émettre de messages d'erreurs. On prendra donc bien soin, dans un tel cas, de nommer correctement la classe : `HelloworldController`. Ce premier exemple qui n'affiche qu'un mot dans la page Web générée est bien pauvre. Enrichissons-le.

4 LES TEMPLATES MAKO

Au lieu de retourner un simple mot sans format, notre contrôleur pourrait retourner du HTML. Par exemple :

```
return '<h1>Programmez!</h1>'
```

Mais c'est précisément ce qu'il faut éviter dans une architecture MVC. Le rendu des données doit être effectué dans la partie 'Vue' de l'architecture. Pylons utilise une librairie de templates: Mako. Le mot template qui se traduit par modèle en français, ne doit pas induire en erreur, nous nous situons bien dans la partie 'Vue' de l'architecture MVC. Pour utiliser les templates Mako, nous commençons par modifier notre contrôleur comme ceci :

```
from pylons import request, response, session, tmpl_context as c, url

# etc.

class HelloController(BaseController):
    def index(self):
        c.mag_favori = "Programmez!"
        return render('/hello.mako')
```

Tout simplement, au lieu de retourner le mot 'Programmez!' nous retournons le rendu du template `hello.mako`, qui n'existe pas encore. Restons encore un instant sur le contrôleur. Nous remarquons que nous ajoutons un membre à un objet nommé `c` de type `tmpl_context`, pour contexte de template. Par ce moyen, nous pouvons passer des paramètres au template. Côté template, le runtime

de Pylons se charge de rendre ce contexte accessible, sous le nom `c` également. Abordons maintenant Mako qui est une librairie de templates à part entière. Cette librairie emprunte, d'après ses concepteurs, le meilleur des templates de Django, de Cheetah, Mighty et Genshi. Mako fournit une syntaxe non XML et à l'exécution les templates sont transformés en modules Python puis compilés en byte-code Python, pour une performance maximum. En plus de tout ceci, Mako propose un mécanisme d'héritage multi-zones, la possibilité d'intégrer des blocks de code Python, etc. Il est maintenant temps de créer notre fichier `hello.mako`, fichier qui doit être déposé dans le sous-répertoire `helloworld/templates`. Voici le code :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=
ISO-8859-1" />
<title>Insert title here</title>
</head>
<body>
```

Votre magazine favori est : \${c.mag_favori}
 Abonnez-vous ! :)

```
<%
from datetime import date
today = date.today()
%>

<p>Date d'aujourd'hui: ${today}</p>

</body>
</html>
```

C'est tout à fait simple. Cet exemple commence par illustrer comment accéder à une variable globale Python, en l'occurrence le contexte `c` dont nous avons parlé plus haut. Ensuite nous avons inséré un petit bloc de code Python qui se charge de définir une variable contenant la date du jour. Enfin nous accédons à cette variable pour l'afficher dans la page générée [Fig.3].

5 LE MÉCANISME DE ROUTAGE

Là encore, au lieu de réinventer la roue, Pylons utilise une librairie très complète, `Routes`, dont le seul propos est de mettre en correspondance URL et actions de l'application. Les possibilités offertes sont nombreuses: simple mise en correspondance d'une URL et d'une action, le support de sous-domaines, le filtrage d'URL avec des expressions régulières, etc. La table de routage (`map`) d'une application Pylons est définie dans le fichier `config/routing.py`. Ce fichier contient un code d'initialisation puis à la fin se trouve une zone dans laquelle vous pouvez définir vos propres routes. Lorsque les modifications sont enregistrées dans le fichier, Pylons a l'intelligence de redémarrer automatiquement. Plus généralement Pylons surveille tous les fichiers sensibles et redémarre dès qu'une modification y est effectuée. Le fichier `routing.py` contient deux routes définies par défaut :

```
# CUSTOM ROUTES HERE

map.connect('/{controller}/{action}')
map.connect('/{controller}/{action}/{id}')

return map
```

Il est maintenant aisé de comprendre pourquoi notre URL `/hello/index` aboutit à l'exécution de la méthode `index` du contrôleur `Hello`. Ce mapping est générique une URL `/autre/` doit provoquerait automatiquement l'exécution de la méthode `doit` d'une classe `Autre-Controller`. Tout cela est très bien et peut être enrichi par exemple par un filtrage d'expressions régulières. Posons-nous maintenant le problème de mettre "n" place une route particulière pour une page très particulière de notre site: la page d'accueil. L'URL `/hello/index` est contre-intuitive et nous voulons simplement `/`. Nous ajoutons donc une ligne à `routing.py`

```
# CUSTOM ROUTES HERE

map.connect(None, '/', controller='helloworld', action='show')
map.connect('/{controller}/{action}')
map.connect('/{controller}/{action}/{id}')
```

Cette ligne particulière doit être placée avant les lignes génériques, car les correspondances sont interprétées du haut vers le bas. Notre ligne de code ne nomme pas notre route (`None`). Nous aurions pu donner, par exemple, le nom `'home'`. Ensuite vient l'URL (`/`) puis le contrôleur et l'action. Pour l'occasion nous avons défini un nouveau contrôleur :

```
class HelloworldController(BaseController):
    def show(self):
        c.mag_favori = "Programmez!"
        return render('/helloworld.mako')
```

qui demande le rendu d'un autre template. Dans ce nouveau template nous avons simplement ajouté le logo de `Programmez`. Nous essayons donc fièrement notre nouvelle URL <http://localhost:5000/> et ... ça ne fonctionne pas. La page statique par défaut que nous avons affichée au tout début de cet article s'affiche dans le navigateur avec obstination. L'explication de ce phénomène se trouve dans un coin de la

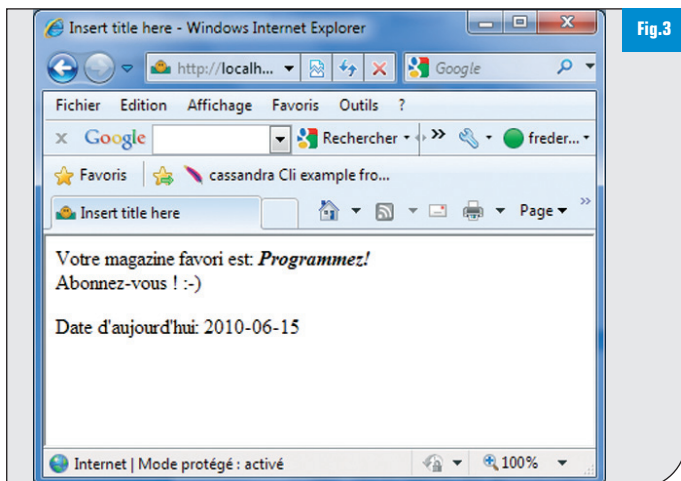


Fig.3

Notre première page dynamique réalisée avec Pylons.

documentation: Pylons sert toujours en priorité les pages statiques. Pour l'URL `/` il regarde dans le répertoire `helloworld/public` avant de regarder la table de routage, et dans ce répertoire, il trouve le fichier `index.html` créé par le `pastor` lors de la génération de l'application. Il suffit d'effacer ce fichier pour obtenir le résultat attendu [Fig.4].

6 TRAVAILLER AVEC UN MODÈLE DE DONNÉES

Pour ce travail avec le dernier aspect de l'architecture MVC, Pylons à la bonne idée de s'appuyer sur le mapper objet/relationnel `SQLAlchemy` qui vous a été présenté dans *Programmez! 1.1.1*. Vous devez d'abord installer `SQLAlchemy` dans votre environnement Python virtuel, puis un adaptateur pour la base de données utilisée, pour nous `SQLite`, donc :

```
easy_install SQLAlchemy
easy_install pysqlite
```

Créez ensuite une nouvelle application, par exemple du nom de `programmez`, et lors de la création, répondez `True` lorsqu'on vous demande si vous voulez utiliser `SQLAlchemy`. On doit ensuite renseigner le fichier `development.ini` afin que `SQLAlchemy` sache comment accéder à la base de données. Dans la section `[app:main]` on donnera par exemple :

```
sqlalchemy.url = sqlite:///C:\mypyhondev\Scripts\programmez\pylons.sqlite
```

Pour cet exemple, nous voulons développer une application qui affiche des données sur les membres de l'équipe de `Programmez`, et nous voulons avoir la possibilité d'ajouter un enregistrement. Le point fort du mapping objet/relationnel, c'est que cela dispense de travailler avec SQL et de créer manuellement les tables de données. A la place, on définit une classe dont les membres seront mappés avec les colonnes de la table. Soit une classe `Personne`, déclarée dans le fichier `personne.py` déposé dans `programmez/model` :

```
from sqlalchemy import Column
from sqlalchemy.types import Integer, String

from programmez.model.meta import Base

class Personne(Base):
    __tablename__ = "personne"
```

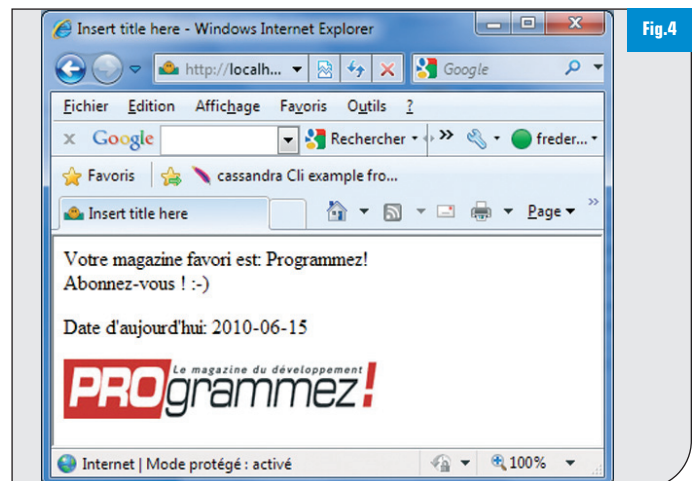


Fig.4

Un hello world routé :-)

```

id = Column(Integer, primary_key=True)
nom = Column(String(100))
prenom = Column(String(100))
fonction = Column(String(100))

def __init__(self, nom='', prenom='', fonction=''):
    self.nom = nom
    self.prenom = prenom
    self.fonction = fonction

def __repr__(self):
    return "Personne: ('%s')" % self.nom

```

Bien remarquer la présence du membre `__tablename__` qui est indispensable pour que le système sache avec quelle table travailler. Ensuite on doit préparer la création des tables en ajoutant quelques lignes au fichier `websetup.py`

```

def setup_app(command, conf, vars):
    """Place any commands to setup program here"""
    # Don't reload the app if it was loaded under the testing environment
    if not pylons.test.pylonsapp:
        load_environment(conf.global_conf, conf.local_conf)
        from programmez.model.meta import Base, Session
        from programmez.model.personne import Personne
        log.info("Creation des tables")
        Base.metadata.drop_all(checkfirst=True, bind=Session.bind)
        Base.metadata.create_all(bind=Session.bind)
        log.info("Fin de l'initialisation")

```

Bien remarquer l'import de la classe `Personne` qui conditionne la création de la table `personne`. Sans cette ligne rien ne se passe lorsqu'on donne la commande :

```
paster setup-app development.ini
```

A l'issue de cette commande la (ou les) tables nécessaires sont créées en base de données. Nous pouvons alors facilement écrire un contrôleur qui ajoute un enregistrement dans la table (URL `/addone/` doit) :

```

class AddoneController(BaseController):
    def doit(self):
        personne = Personne(nom='Mazue',
                             prenom='Frederic', fonction='Auteur')
        Session.add(personne)
        Session.commit()
        return 'addone done'

```

Mais bien évidemment, maintenant que nous sommes capables d'ajouter un enregistrement, il faut offrir à l'utilisateur la possibilité de le faire par un formulaire. Pour cela on commence par écrire un nouveau contrôleur :

```

class AddController(BaseController):
    def doit(self):
        #return render('add-form-with-webhelpers.mako')
        return render('add-form.mako')

```

```

def process(self):
    personne = Personne(nom=request.params['nom'],
                        prenom=request.params['prenom'],
                        fonction=request.params['fonction'])
    Session.add(personne)
    Session.commit()
    return redirect(url(controller='team', action='show'))

```

Le principe de fonctionnement est le suivant, on commence par rendre un template qui affiche un formulaire ainsi défini :

```

<form name="add-record" method="POST" action="/add/process">
Prenom: <input type="text" name="prenom" /><br />
Nom: <input type="text" name="nom" /><br />
Fonction: <input type="text" name="fonction" />
<p><input type="submit" name="submit" value="Submit" /></p>
</form>

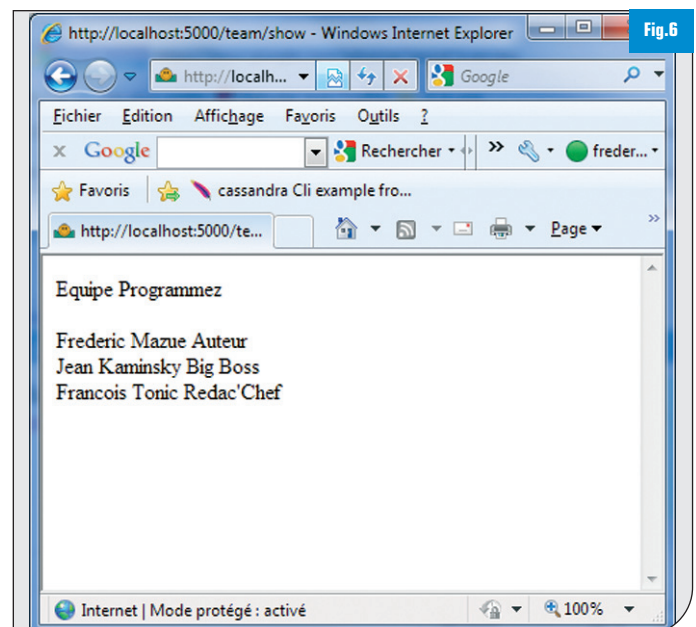
```

Puis, tout simplement, l'attribut `action` du formulaire se charge d'invoquer la méthode `process` du contrôleur. Là, nous récupérons les paramètres et ajoutons l'enregistrement. Ceci fait, nous redirigeons vers un autre contrôleur (code disponible sur notre site) qui se chargera du rendu du contenu de la table [Fig.6]. Bien remarquer l'invocation de `redirect` et `non redirect_to` comme dit par erreur dans la documentation. Grand réutilisateur de composants, Pylons propose la librairie `WebHelpers` pour créer des formulaires. Nous invitons le lecteur à jeter un oeil au template `add-form-with-webhelpers.mako` ainsi qu'au code ajouté au fichier `lib/heloers.py`

7 CONCLUSION

Construit sur d'excellentes librairies, Pylons est un remarquable framework de développement agile, utilisable facilement même par les débutants, et dont le modèle MVC impose un code bien structuré. Souhaitons-lui de gagner rapidement une grande popularité.

■ Frédéric Mazué - fmazue@programmez.com



Toutes les données de notre table affichées dans une page

DÉVELOPPEZ VOTRE SAVOIR-FAIRE

Economisez jusqu'à 50%



Programmez ! est le magazine du développement Langage et code, développement web, carrières et métier : Programmez !, c'est votre outil de veille technologique.

Pour votre développement personnel et professionnel, abonnez-vous à Programmez ! www.programmez.com

1 -25%

Abonnement 1 an

49€ au lieu de 65,45 € tarif au numéro - Tarif France métropolitaine

2 +0,8€ par mois

Abonnement Intégral : + archives

1 an au magazine + archives sur Internet et PDF
59€ Tarif France métropolitaine

3 jusqu'à -50%

Abonnement 2 ans + 1 livre numérique ENI

• 79€ au lieu de 130,90 (valeur de 22 numéros) Tarif France métropolitaine + un livre d'une valeur de 23,9 € à 31,9 €, soit un total de 154,8 € à 162,8 €

• 89€ 2 ans au magazine + archives sur Internet et PDF + 1 livre numérique ENI



OUI, je m'abonne

Vous pouvez vous abonner en ligne et trouver tous les tarifs www.programmez.com

☐ Abonnement 1 an au magazine : 49 € (au lieu de 65,45 € tarif au numéro) Tarif France métropolitaine

☐ Abonnement Intégral : 1 an au magazine + archives : 59 € Tarif France métropolitaine

☐ Abonnement 2 ans au magazine + livre numérique ENI : 79 € Tarif France métropolitaine

☐ Abonnement 2 ans au magazine + livre numérique ENI + archives : 89 € Tarif France métropolitaine

Livres à Choisir : ☐ Visual Studio 2010 ☐ PHP5.3 ☐ Bing Maps ☐ MySQL 5, Administration et optimisation

☐ Java et Spring, Concevoir, construire et développer une application Java/J2EE avec Spring. Détails sur www.programmez.com/abonnement.php

☐ M. ☐ Mme ☐ Mlle Entreprise : _____ Fonction : _____

Prénom : _____ Nom : _____

Adresse : _____

Code postal : _____ Ville : _____

Tél : _____ (Attention, e-mail indispensable)

E-mail : _____ @ _____

☐ Je joins mon règlement par chèque à l'ordre de Programmez ! ☐ Je souhaite régler à réception de facture

A remplir et retourner sous enveloppe affranchie à : Programmez ! - Service Abonnements - 22 rue René Boulanger - 75472 Paris Cedex 10. abonnements.programmez@groupe-gli.com

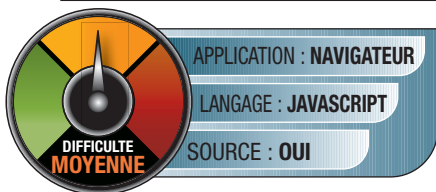
Offre limitée, valable jusqu'au 31 décembre 2010

Le renvoi du présent bulletin implique pour le souscripteur l'acceptation pleine et entière de toutes les conditions de vente de cette offre. Conformément à la loi Informatique et Libertés du 05/01/78, vous disposez d'un droit d'accès et de rectification aux données vous concernant. Par notre intermédiaire, vous pouvez être amené à recevoir des propositions d'autres sociétés ou associations. Si vous ne le souhaitez pas, il vous suffit de nous écrire en nous précisant toutes vos coordonnées.

La magazine du développement
PROgrammez !

Ecrire des extensions pour le navigateur Google Chrome

Google Chrome est un navigateur qui gagne en popularité, surtout depuis qu'il permet d'écrire des extensions. Nous voyons aujourd'hui comment procéder.



Nous vivons une période au cours de laquelle les navigateurs Internet se livrent à une âpre concurrence. Les développeurs des principaux navigateurs, Internet Explorer,

Firefox, Opera, Safari et Google Chrome s'efforcent notamment d'améliorer les performances de leurs moteurs Javascript et le respect des standards que sont CSS 3 et HTML 5. Pour son navigateur Chrome, Google avait commencé par mettre en avant sa légèreté, sa rapidité et sa simplicité. Et malgré cela Chrome ne gagnait pas en popularité. Ce n'est que quand il est devenu possible de programmer des extensions que Chrome a gagné des parts de marché, ce qui semble indiquer clairement qu'un navigateur non extensible n'a pas ou peu d'avenir. Apple l'a compris aussi en dotant récemment son navigateur Safari 5 de cette possibilité. Le défi était alors pour Google de conserver la légèreté du navigateur tout en le rendant extensible. Cela semble plutôt réussi, l'interface de programmation est simple et claire ainsi que nous le découvrons maintenant.

1 ANATOMIE D'UNE EXTENSION À GOOGLE CHROME

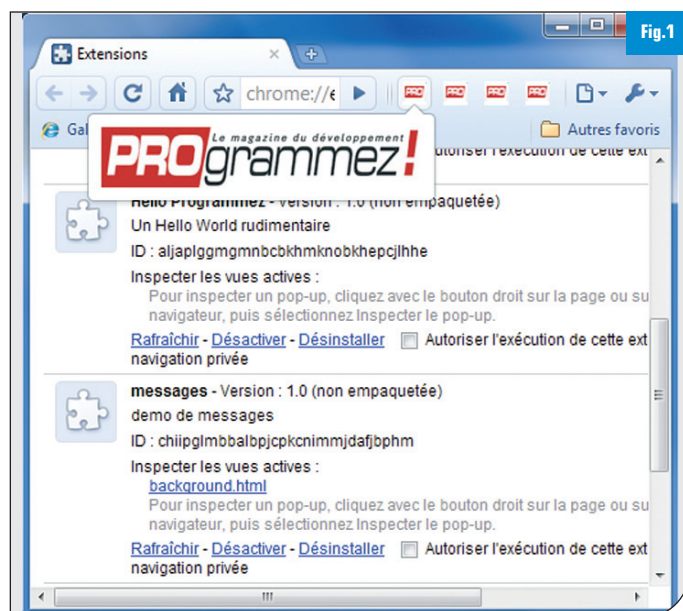
Tout développeur Web connaissant HTML, Javascript et CSS peut développer une extension Chrome, car ce sont les seuls outils nécessaires. Contrairement à d'autres navigateurs, et en raison de la simplicité de son interface utilisateur, on ne devra pas travailler avec des fichiers à la XML pour décrire l'interface de l'extension proprement dite. Les points d'intégration sont peu nombreux. Une icône cliquable dans une barre d'outils à droite de la zone de texte de l'URL courante, et une icône cliquable dans la zone de texte sont les seules possibilités en plus des popup ou des pages HTML. Rappelons que Chrome n'a pas de barre de statut ni de barre de menus, donc il n'est pas possible de s'y brancher. Enfin, il n'est pas possible, du moins au moment de la rédaction de cet article, d'ajouter une entrée dans les menus déroulants de Chrome. Par contre il est possible d'agir sur la liste des favoris et aussi de faire apparaître des notifications dans la barre des tâches du système d'exploitation hôte. Le point central d'une extension Chrome est un fichier manifeste écrit au format JSON, qui décrit brièvement l'extension, énumère tout ou partie de ses fichiers constitutifs, et, surtout, octroie des permissions. JSON est d'ailleurs omniprésent dans le travail avec l'API. Ensuite une extension s'articule soit autour d'une page HTML rudimentaire dite popup qui apparaît lorsque l'utilisateur clique sur l'icône de l'extension, soit autour d'une page HTML dite background qui reste toujours invisible. L'expérience montre qu'il est possible d'utiliser à la fois un background et un popup, à condition d'être attentif au fait que dans ce cas, un clic sur l'icône de l'extension fait surgir le popup et n'est pas transmis au background. En

plus du background ou du popup il est possible d'ouvrir des onglets et d'y charger des pages HTML. Enfin une extension peut être constituée de fichiers Javascript normaux, dits de contenu, et destinés à être injectés dans les pages visitées. Bien entendu, il est possible d'utiliser toutes les possibilités de Javascript classiques et notamment la fonction XMLHttpRequest() pour obtenir des données de diverses sources.

2 HELLO WORLD!

Commençons par nous faire la main avec une première extension rudimentaire qui ouvre un simple popup qui affiche le logo de Programmez! [Fig.1]. Comme nous l'avons dit, une extension Chrome est constituée par un ensemble de fichiers. Ceux-ci doivent être déposés dans un répertoire quelconque. Avant tout, nous devons définir le contenu du fichier manifest.json :

```
{
  «name»: «Hello Programmez»,
  «version»: «1.0»,
  «description»: «Un Hello World rudimentaire»,
  «browser_action»: {
    «default_icon»: «prog_icon.png»,
    «default_title»: «Programmez!»,
    «popup»: «popup.html»
  }
}
```



Notre premier exemple se contente d'ouvrir un popup.

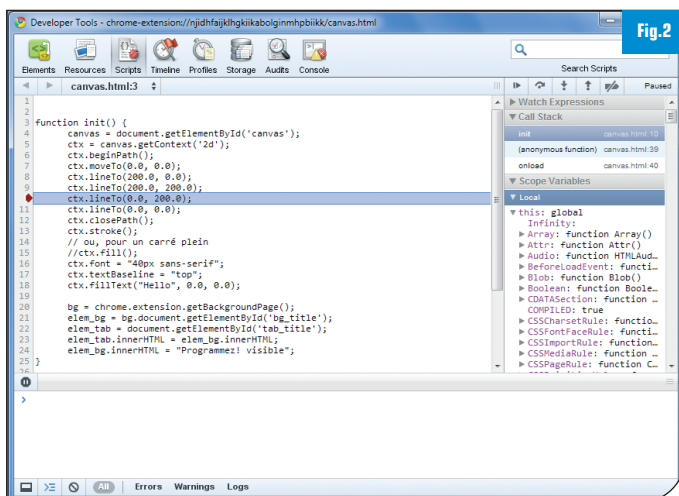
L'icône de l'application (default_icon) doit être d'une taille de 19*19 pixel. Elle apparaît à droite de la zone d'URL. Le titre (default_title) est le texte d'une bulle d'aide qui apparaît lorsque le pointeur de la souris s'attarde sur l'icône. Le popup est un contenu HTML :

```
<script>
alert(«Abonnez-vous»);
</script>
<img title=» src=»logo_programmez.jpg»>
```

La présence de la définition du popup dans le fichier manifest.json suffit à en assurer l'apparition lorsque l'utilisateur clique sur l'icône. Le popup se dimensionne automatiquement autour de son contenu qui, en plus du HTML, peut être du code Javascript. Le fichier de l'image est déposé dans le répertoire de l'extension et on donne simplement son chemin relatif pour qu'elle puisse être chargée. Tout est prêt. Pour installer l'extension, on se rend dans le menu de Chrome Réglages/Extensions, ou simplement, on saisit cette URL dans le navigateur: chrome://extensions/. Une fois dans la page de gestion des extensions, on clique sur le signe plus à côté de mode développeur, pour ouvrir ce mode, puis on clique sur le bouton 'Charger l'extension non emballée'. Après quoi l'on donne le répertoire contenant les fichiers, et l'extension s'installe, à condition que le fichier manifest.json soit correct. Si au cours du développement, on modifie un des fichiers, on demandera à Chrome d'en tenir compte en cliquant sur le lien 'Rafraîchir'. On notera la présence du bouton 'Empaqueter l'extension qui permet de créer une archive distribuable, y compris sur la galerie de Google. Enfin il est possible de déboguer une extension. Dans le cas d'un popup, on clique avec le bouton droit de la souris sur l'icône de l'extension, puis on choisit 'Inspecter le popup' dans le menu contextuel qui apparaît pour se retrouver dans le débogueur. Dans le cas d'un background ou de n'importe laquelle de nos pages actives, il suffit de cliquer sur le lien background ou le lien de la page dans la page de gestion des extensions pour faire apparaître le débogueur. Dans tous les cas, il suffit de cliquer sur le numéro d'une ligne de code Javascript pour définir un point d'arrêt. Lorsque l'exécution arrive sur le point d'arrêt, le débogueur devient actif. Sinon il est possible de recharger la page en cours en saisissant :

```
location.reload()
```

Dans la console du débogueur. **[Fig.2]**



Débogage de code Javascript dans le débogueur de Chrome

3 BACKGROUND ET CANVAS HTML5

Nous allons maintenant écrire un exemple un petit peu plus élaboré, car centré autour d'un background. Ce qui va nous montrer que cette page invisible est l'endroit adapté au stockage des informations concernant l'état de l'application qu'est une extension Chrome. On stockera les données soit comme des éléments du DOM, soit dans des variables Javascript. Nous verrons aussi que c'est l'endroit pour placer du code Javascript susceptible d'être accédé à tout moment et de n'importe où. Vient d'abord le fichier manifest.json

```
{
  «background_page»: «background.html»,
  «browser_action»: {
    «default_title»: «»,
    «default_icon»: «prog_icon.png»
  },

  «description»: «background-canvas-description»,

  «name»: «background-canvas»,

  «permissions»: [
    «tabs»
  ],

  «version»: «1.0»
}
```

Nous voyons que la déclaration background_page remplace la déclaration popup de l'exemple précédent. En outre, nous découvrons un autre élément important du manifeste : les permissions. Ici nous donnons à notre extension la permission d'ouvrir des onglets (tabs) dans le navigateur. Voici maintenant le code de la page background :

```
<html>
<head>
<script>

function modifieElement(texte) {
  elem_bg = document.getElementById('bg_title');
  elem_bg.innerHTML = texte;
}

chrome.browserAction.onClicked.addListener(function() {
  chrome.browserAction.setBadgeText({«text» : «Prog»});
  chrome.tabs.create({«url» : «canvas.html»});
});

</script>
</head>
<body>
<p id=»bg_title»Programmez! invisible :-)</p>
</body>
</html>
```

Cette page contient une donnée: le contenu du paragraphe d'id «bg_title». La valeur est «Programmez! invisible» car comme nous

l'avons dit ce document HTML n'apparaît pas dans le navigateur. Le code Javascript est simple. D'abord nous définissons une fonction capable de modifier l'élément de DOM qu'est le paragraphe mentionné plus haut. Ensuite le code Javascript définit un écouteur à une action [clic] de l'utilisateur sur l'icône de l'extension. Notre écouteur réagit au clic, d'abord en affublant l'icône de l'extension d'un badge, qui est une petite décoration, puis en ouvrant un onglet. Puisque nous invoquons ici pour la première fois les API Chrome, remarquons bien l'utilisation d'un espace de nom à chaque fois, et surtout, notons que les arguments passés aux API sont toujours au format JSON. Voici maintenant le code de la page qui s'affiche dans l'onglet ouvert :

```
<html>
<head>
<script>

function init() {
    canvas = document.getElementById('canvas');
    ctx = canvas.getContext('2d');
    ctx.beginPath();
    ctx.moveTo(0.0, 0.0);
    ctx.lineTo(200.0, 0.0);
    ctx.lineTo(200.0, 200.0);
    ctx.lineTo(0.0, 200.0);
    ctx.lineTo(0.0, 0.0);
    ctx.closePath();
    ctx.stroke();
    // ou, pour un carré plein
    //ctx.fill();
    ctx.font = «40px sans-serif»;
    ctx.textBaseline = «top»;
    ctx.fillText(«Hello», 0.0, 0.0);

    bg = chrome.extension.getBackgroundPage();
    elem_bg = bg.document.getElementById('bg_title');
    elem_tab = document.getElementById('tab_title');
    elem_tab.innerHTML = elem_bg.innerHTML;
    elem_bg.innerHTML = «Programmez! visible»;
}

function clic() {
    texte = 'Abonnez-vous! :-)'
    elem_tab = document.getElementById('tab_title');
    elem_tab.innerHTML = texte;
    bg = chrome.extension.getBackgroundPage();
    bg.modifieElement(texte);
}

</script>
</head>
<body onload=»init()»>
<h1 id=»tab_title»>Titre</h1>

<p onclick=»clic()»>Cliquez ici</p>
<canvas id=»canvas» width=»200» height=»200»></canvas>

</body>
</html>
```

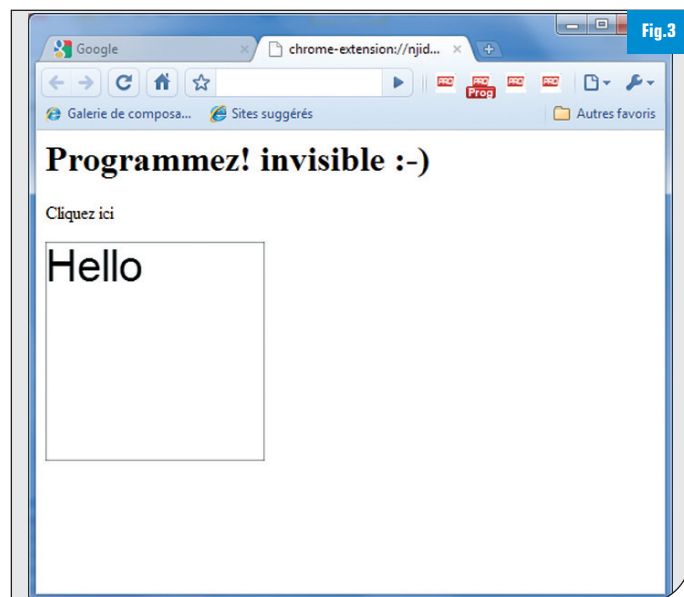
Ce code nous fait faire connaissance avec 'canvas', une des nouvelles balises HTML5. Notre fonction init réalise un tracé dans ce canvas [Fig.3]. Le code du tracé nécessite peu de commentaires. Bien remarquer l'initialisation du contexte, puis bien remarquer la présence d'appels à des API de nom strokeXXX ou fillXXX. Ce sont ces appels qui rendent les tracés effectifs dans le canvas. Sans eux rien ne s'affiche. A la fin de la fonction init, on remarque que l'on accède à la page background avec l'API chrome.extension.getBackgroundPage. Cette API est toujours accessible, donc le background est toujours accessible par ce moyen. Ensuite nous récupérons le contenu du paragraphe du DOM du background, puis nous affectons ce contenu au titre (balise h1) de notre page. Puis nous modifions le contenu du paragraphe dans le DOM du background. A ce stade, si l'utilisateur clique à nouveau sur l'icône, un nouvel onglet s'affiche, qui contient le nouveau titre. Enfin la fonction 'clic' montre que l'on peut accéder à une fonction définie dans le code Javascript du background, en l'occurrence la fonction modifieElement que nous y avons définie.

4 COMMUNICATIONS PAR PASSAGE DE MESSAGES

Communiquer en invoquant les fonctions définies dans d'autres pages est bien, mais tôt ou tard on aura besoin d'un mécanisme plus évolué. Ce mécanisme existe: la communication par messages. Cette communication peut s'effectuer selon deux modes: un mode «one shot» dans lequel on envoie un message et on attend la réponse, ou une communication dans laquelle les messages sont placés dans une file. Notre nouvel exemple ('messages' sur notre site) illustre le premier mode. En outre, comme nous utilisons aussi les notifications sur le bureau de l'utilisateur, notre fichier manifeste contient une nouvelle permission :

```
«permissions»: [
    «tabs»,
    «notifications»
],
```

Voici maintenant le code du background



```

<script>

var AllTabIds = new Array();

function storeTab(tab) {
  AllTabIds.push(tab.id);
}

function closeAll() {
  for(var i = 0; i < AllTabIds.length; i++)
    chrome.tabs.remove(AllTabIds[i]);
}

function handler(request, sender, sendResponse) {
  sendResponse(«msg reçu « + request.msg);
  setTimeout(closeAll, 4000);
}

chrome.extension.onRequest.addListener(handler);

chrome.browserAction.onClicked.addListener(function() {
  chrome.browserAction.setBadgeText({«text» : «Msg»});
  chrome.tabs.create({«url» : «tab1.html»}, storeTab);
  chrome.tabs.create({«url» : «tab2.html»}, storeTab);
});

</script>

```

Aucune balise HTML dans cette page, car il apparaît que ce n'est pas nécessaire. Tout d'abord le mécanisme de passage de messages est initialisé ainsi: `chrome.extension.onRequest.addListener(handler)`. Dans le gestionnaire du message (handler), on répond à l'envoyeur (on doit toujours répondre).

Si l'on a rien à dire, on retournera `{}` qui est un onglet, puis, après un délai de 4 secondes, on ferme les deux onglets qui sont ouverts lorsque l'utilisateur clique sur l'icône. Bien remarquer à cette occasion comment les références sur les onglets sont conservées au moment de leur création.

Les API Chrome ont ceci de particulier (et de détestable) qu'elles ne retournent pas de valeur. Au lieu de cela, elles demandent à recevoir une fonction de rappel en argument et passent à ce callback, ce qui serait normalement une valeur de retour.

Ce procédé peut avoir son intérêt, mais si les API retournaient une valeur lorsqu'on ne leur passe pas de callback, ça ne serait sans doute pas plus mal. Les deux onglets ont un code identique que l'on trouvera sur notre site.

Ce code, lorsqu'on clique sur le texte, émet un message à destination du background, et lorsqu'il celui-ci répond, cela provoque l'invocation du callback `backhandler`. Celui-ci démontre comment ouvrir une notification sur le bureau. Aucune difficulté. Mais une remarque importante s'impose. D'après la documentation, une notification est fermée lorsque l'onglet qui l'a ouverte est fermé. Dans la pratique ça ne fonctionne pas (au moins sous Windows) ce qui explique que

nous fermons les notifications après un délai de 2 secondes et donc avant la fermeture des onglets qui se produit après 4 secondes.

5 UNE «VRAIE» EXTENSION

Une extension qui n'entre en action que lorsque l'utilisateur clique sur son icône offre un intérêt pratique très limité. Il est mieux qu'elle réagisse lorsque l'utilisateur arrive sur une page déterminée, ce que fait notre dernier exemple. Si l'utilisateur surfe sur le site de *Programmez!* et seulement sur ce site, il reçoit une notification l'informant qu'il n'est pas abonné. Puis un script contenu est injecté dans la page courante pour modifier la barre de titre dans le navigateur, ceci tant que l'utilisateur ne se rend pas dans la page abonnement. Une fois abonné, si l'utilisateur quitte *Programmez!* puis y revient, il sera notifié qu'il est abonné. Voyons d'abord le fichier `manifest.json`, enrichi d'une nouvelle permission :

```

«permissions»: [
  «tabs»,
  «notifications»,
  «http://www.programmez.com/*»
],

```

Le code du background est disponible sur notre site.

On y remarque l'installation d'un écouteur sur l'événement `onUpdated` qui est déclenché chaque fois que le contenu d'un onglet est modifié. Lorsque le site *Programmez* est rencontré, on injecte le script dans la page courante (`chrome.tabs.executeScript`). Voici le code de ce script :

```

if(document.URL == «http://www.programmez.com/abonnement.php»)
{
  chrome.extension.sendRequest({msg: «set», value: 1});
}

function backhandler(response) {
  if(parseInt(response.value) == 0)
    document.title = «Abonnez-vous!»;
}

chrome.extension.sendRequest({msg: «get»}, backhandler);

```

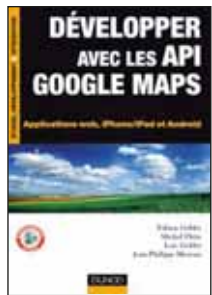
Notre script injecté ne paie pas de mine. Mais c'est pourtant quelque chose de très puissant puisqu'il accède au DOM du document courant et peut le modifier totalement. Ainsi nous pourrions interroger *Google Analytics* au moyen de `XMLHttpRequest` au sujet de la page visitée et selon le procédé expliqué dans *Programmez! 131*, puis décorer chacun des liens avec le nombre de clics effectués dessus au cours de la journée par exemple. Enfin l'état «abonnement» du visiteur est conservé dans une variable Javascript et est donc volatile d'une session à l'autre. Nous aurions pu conserver cet état dans un `localStorage`, une fonctionnalité HTML5 que Chrome implémente. Concrètement il s'agit d'un dictionnaire sous la forme d'un objet Javascript persistant et affecté au nom de domaine visité.

■ Frédéric Mazué - fmazue@programmez.com

Tous les jours : l'actu et le téléchargement
www.programmez.com

GÉOLOC

Développer avec les API Google Maps



Difficulté : ***
 Editeur : Dunod
 Auteur : divers
 Prix : 32 €

Google Maps possède une grande richesse pour le développeur aussi bien sur desktop que

Smartphone. La version 3 des API Google Maps permet aux développeurs de sites web d'enrichir plus facilement leurs pages de cartes numériques interactives et offre un meilleur support des périphériques mobiles (iPhone, iPad, Android, etc.). Cet ouvrage décrit l'essentiel des fonctionnalités de l'API, ainsi que les méthodes et outils de développement qui vous permettront d'intégrer facilement des données géolocalisées sur un fond de carte Google Maps et de tirer le meilleur parti des outils de webmapping, visualisation de données géoréférencées, etc. La deuxième partie se focalise sur les nouveaux usages rendus possibles par le succès des smartphones. Enfin, l'API open source OpenLayers est également présentée, ainsi que les API Google Maps statiques.

MOBILE

Programmez pour iPhone, iPod touch, iPad



Difficulté : ***
 Editeur : Pearson
 Prix : 19 €

Écrit par l'équipe iPup, cet ouvrage sort du lot. Tout d'abord par sa présentation et par le fait qu'il est constitué de 37 fiches. Ces fiches doivent

vous aider à mieux comprendre iOS, le système des terminaux mobiles d'Apple : langage, outils, structure d'une application iOS, etc. Tous les aspects sont abordés : traduction des applications, animations, interface, multitouche, le UIKit. Sur le contenu, rien à redire. Cependant, nous ne trouvons pas le mode d'impression (il reprend le principe d'un bloc note avec reliure en haut) très pratique à l'usage.

LIVRE DU MOIS

Développement d'applications professionnelles avec Android2

Difficulté : *** - Editeur : Pearson
 Auteur : Reto Meier - Prix : 45 €

Un grand nombre d'ouvrages de programmation sur Android oublient un élément important : comment et pourquoi développer une application professionnelle ? L'auteur se propose ici de combler cette lacune avec un ouvrage très complet. On revient aux fondamentaux du développement Android, à la maîtrise de l'interface graphique, la géolocalisation, gestion de fichiers, réseau, etc. Le modèle de programmation ne change pas, seule change l'approche que l'on doit avoir, la qualité de l'application et du code. L'auteur fournit de très nombreux codes, des conseils, des bonnes pratiques. Une excellente base pour débiter ou se perfectionner !



DESIGN

Design et interface



Difficulté : **
 Editeur : Eyrolles
 Auteur : Amélie Boucher
 Prix : 32 €

Comment réussir une bonne interface, avoir une bonne ergonomie ? Quelles sont les tendances actuelles ? De quelle manière organiser les pages, l'information, le contenu ? Beaucoup de sites se « plantent » à cause de ce travail tellement important. 60 sites ont été pris en exemple puis expliqués à la loupe pour comprendre à chaque fois ce qui les caractérise et comment se structurent les pages, le contenu. Mais les auteurs montrent aussi les lacunes, les faiblesses des interfaces, de l'ergonomie. Un livre que nous recommandons à tout développeur web et designer web.

modifications apportées au langage. L'ouvrage est une invitation au développement C# notamment en Windows Forms. .Net 4 est bien entendu présenté ainsi que Click Once, Windows Installer, LINQ, ADO.Net. De nombreux codes sont présentés et expliqués.

SÉCURITÉ

Outils d'analyse forensique sous Windows

Difficulté : ****



Editeur : Pearson
 Auteur : Harlan Carvey
 Prix : 59 €

Pour la première fois en français, découvrez l'art des outils d'analyses forensiques. Écrit

par l'un des meilleurs experts mondiaux, cet ouvrage apporte une boîte à outils très complète pour analyser en profondeur les systèmes Windows, qui, du fait de leur popularité, forment une cible de choix pour les intrusions, les infections par des logiciels malveillants et la cybercriminalité. L'auteur fait découvrir les entrailles du système, les failles potentielles, comment les découvrir, les exploiter et faire des contre mesures. Cet ouvrage traite des aspects techniques de la collecte et de l'analyse des données au cours des investigations sur des systèmes en fonctionnement ou ayant rendu l'âme. Le livre aborde les problèmes de registres, les rootkit, faire une analyse avec un budget réduit, etc. La qualité de l'ouvrage justifie amplement l'investissement de 59 euros. Indispensable !

LANGAGE

C# 4, les fondamentaux du langage



Difficulté : ***
 Editeur : Eni
 Auteur : Thierry Grossard
 Prix : 29,90 €

L'auteur revient rapidement sur Visual Studio et les grands principes de l'orienté objet dans C# ainsi que sur les nouveautés et

Depuis la sortie de VS 2010, C# est en version 4.

RÉUSSISSEZ VOS CERTIFICATIONS AVEC **Microsoft®** *Press*



9782100521975 ■ 800 pages



9782100521968 ■ 640 pages

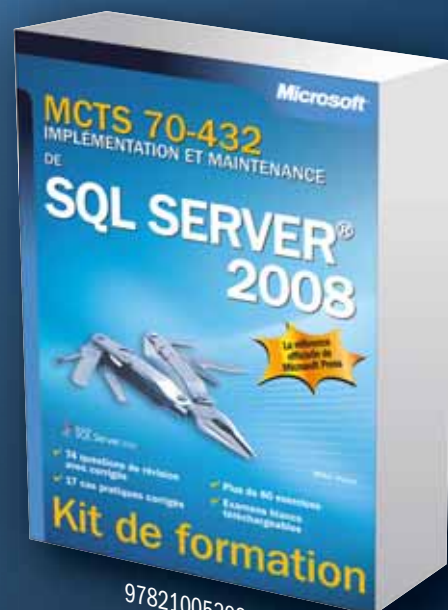
39 €
seulement
à partir du 1^{er} novembre 2010



9782100532711 ■ 624 pages



9782100532728 ■ 688 pages



9782100530366 ■ 672 pages

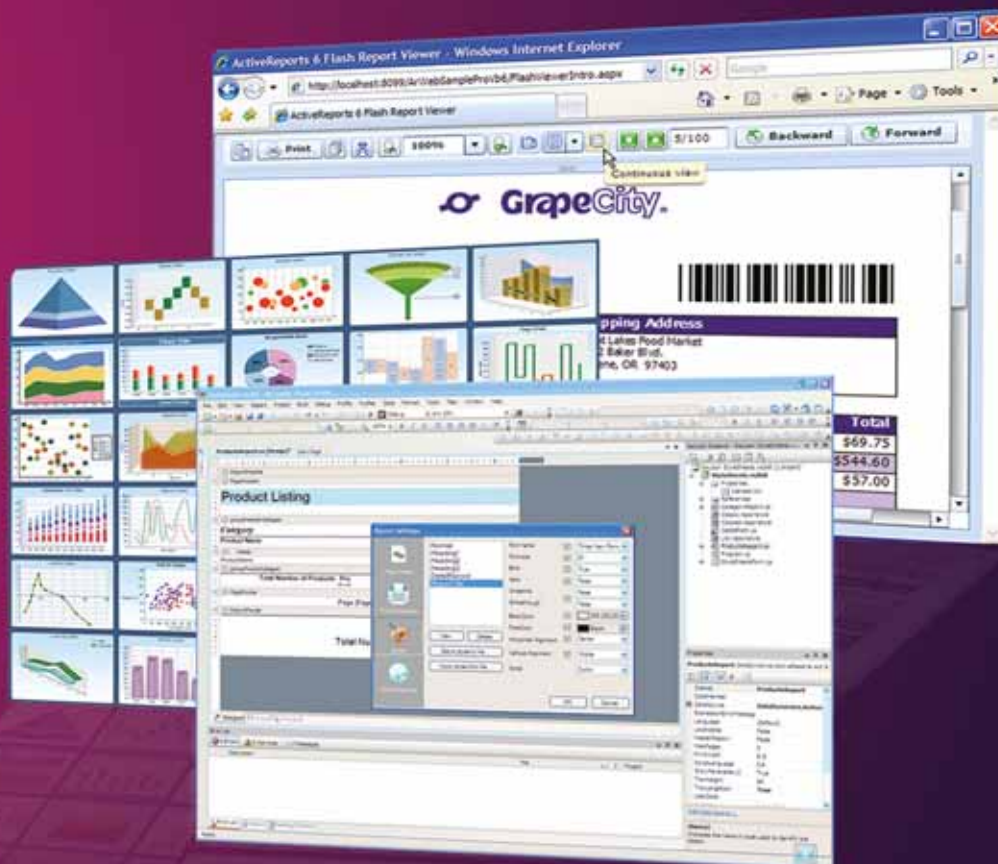
Les **Kits de formation Microsoft Press** sont les ouvrages de référence, exhaustifs et pratiques certifiés par Microsoft. Ils contiennent toute l'expertise nécessaire pour vous accompagner dans l'évolution de votre métier et pour réussir les examens Microsoft.

Tous nos ouvrages sont disponibles en librairie ou sur commande auprès de votre libraire.

MICROSOFT PRESS : LA RÉFÉRENCE OFFICIELLE

WE ARE
REPORTING

ACTIVE REPORTS 6



*Cet outil de reporting
pour Microsoft Visual Studio.NET
est LE standard.
Il balaie la concurrence, un point c'est tout.*



GCPowerTools.com



WE ARE
GrapeCity
Report & Analyze & Excel