PROGRAMMEZ !
Le magazine du développement
PROgrammez !
www.programmez.com
mensuel n°121 - juillet-Aout 2009

Le futur est là !

Cahier central
Programmation Multicœur

Windows 7

© Aldebaran

Enquête exclusive

Etes-vous SATISFAIT **de votre** JOB ?

Outils

- **Tout savoir sur** Visual Studio 2010
- **Maîtriser Adobe** Dreamweaver

JavaOne 2009

Java FX 1.2 débarque

Java va-t-il devenir PAYANT ?

Windows 7, Java 7, Surface, Android, iPhone, robotique, Interface Cerveau-Ordinateur...
Etes-vous prêt ?

Cloud
Migrer un site ASP.Net dans Azure

Java
Découvrez la puissance de Clojure

Outils
DSS : un environnement surpuissant !

C++
Créer son compilateur

M 04319 - 121 - F: 5,95 € - RD

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

WINDEV WEBDEV WINDEV Mobile

Développez 10 fois plus vite

Nouvelle version 14

PLATEFORME PROFESSIONNELLE DE DÉVELOPPEMENT (AGL)
Windows, .Net, Java
Windows 98, 2000, NT, 2003, XP, Vista, 2008

WINDEV®

DÉVELOPPEZ 10 FOIS PLUS VITE AVEC L'OUTIL DE DÉVELOPPEMENT ÉLU LE PLUS PRODUCTIF*

WINDEV 14 est l'environnement de développement totalement intégré (IDE, ALM), intégralement en français, réputé pour sa **puissance** et sa **facilité** d'utilisation.

WINDEV 14 est livré complet: éditeur d'analyses (UML,...), RAD, patterns, lien avec toutes les bases de données (ODBC, OLE DB), Oracle, SQL Server, AS/400, Informix, DB2..., lien natif MySQL, PostgreSQL, base de données Client/Serveur HyperFileSQL gratuite incluse, Générateur d'états PDF, Codesbarres, Accès natif SAP R/3, Lotus Notes, Gestion de planning, Gestion des Exigences, L5G, SNMP, Bluetooth, TAPI, OPC, FTP, HTTP, Socket, Twain, API, DLL, domotique, liaisons série et USB, débogage à distance, profiler, refactoring, génère d'applications JAVA, multilangue automatique, Gestionnaire de versions, Installateur 1-clic, etc...

Les applications créées fonctionnent avec toutes les versions de Windows: 98, 2000, NT, 2003, XP, Vista, sous TSE et Citrix, sur eeePC...

WINDEV 14 gère le cycle complet de développement, pour des équipes de 1 à 100 développeurs. Le support technique est gratuit*.

VERSION EXPRESS GRATUITE
Téléchargez-la !

Vous aussi, développez 10 fois plus vite... avec WINDEV 14.

VOTRE CODE EST MULTI-PLATEFORMES:
Windows, .Net, Java, PHP, J2EE, XML, Internet, Ajax, Pocket PC, SmartPhone, Client riche ...

Logiciel professionnel

DEMANDEZ LE DOSSIER GRATUIT
252 pages + DVD + **Version Express incluse** + 112 Témoignages.
Tél: **04.67.032.032** ou **01.48.01.48.88**
info@pcsoft.fr

Fournisseur Officiel de la Préparation Olympique

www.pcsoft.fr

PC SOFT 25 ANNEE

Logiciel professionnel. Documentation contractuel. **Support technique gratuit**: 15 requêtes sur la version en cours de commercialisation.
*: WINDEV a été élu «Langage le plus productif du marché» par les lecteurs de la revue «Programmez!», octobre 2008

Donnez votre avis sur ce numéro : www.programmez.com/magazine_satisfaction.php

PROgrammez! le magazine du développement

numéro 121 - juillet/août 2009

sommaire

\\ actus

L'actualité en bref 6
Agenda 10

\\ événements

Community One / Java One 2009 12

\\ outils

Visual Studio 2010 : visite guidée 14
Maîtriser Dreamweaver : site dynamique et MySQL (1er partie) 20

\\ dossier : **Le futur est là**

Robotique : le prochain défi ! 25
Les interfaces cerveau-ordinateurs : le futur de l'informatique 28
Préparez-vous à Windows 7 dès aujourd'hui 30
iPhone : le téléphone devient enfin intelligent et connecté 51
Développez sur mobile avec Android de A à Z 55
Java 7 : la dernière ligne droite 59
Microsoft Surface : quand la table devient interactive ! 63

\\ Cahier central :

Spécial développement parallèle 35

Pourquoi passer à la programmation parallèle (2e partie) II
Simplifier le développement parallèle VI
La chasse aux bogues parallèles VIII
Plus de cœurs, plus rapide, plus de montée en charge X
Vous ne reconnaîtrez plus votre C++ XII

\\ carrière

Etes-vous satisfait de votre job ? 66

\\ code

Découvrir l'environnement DSS et son langage de programmation visuel ..68
Clojure, un Lisp pour la JVM 72
Réalisation d'un compilateur de Pascal simplifié en C++ 76
Migration d'une application asp.net vers Windows Azure 80

\\ temps libre

Les livres du mois 82

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.

LEONARDi est une marque du groupe W4-LYRIA. Les autres marques citées dans ce document sont des marques déposées par leurs propriétaires respectifs

LEONARDi

W4-LYRIA : la modélisation au coeur des applications métier

L'ihm

EN TOUTE SIMPLICITÉ !

Pour implémenter des applications riches, accédant à des données hétérogènes (applicatifs métier, BD, et depuis la V4.1, Hibernate, GED et silos documentaires), déployées en Swing, plugin Eclipse ou DHTML/Ajax et s'insérant naturellement dans les processus d'entreprise...

Adoptez la puissance et l'agilité de l'approche Model-Driven

Concentrez-vous sur votre métier et dotez votre entreprise d'avantages compétitifs durables: amélioration du cycle de vie du logiciel, démarche itérative par prototypage pour coller aux besoins, découplage technologie/métier, évolutivité, agilité et évolutivité, le tout sans expertise technique pointue !

Rendez-vous le 26 Novembre

<MDDAY/> 2009

Inscriptions sur www.mdday.fr

LYRIA

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

éditoll

Le mouvement perpétuel

Le développeur peut-il suivre la course parfois effrénée des versions, des nouveaux outils sans perdre en productivité ou introduire des bugs dans ses codes ? La question doit sérieusement se poser. Même dans les équipes de Symfony, le framework PHP, le temps est arrivé de la maturité, de la stabilité car des utilisateurs se plaignaient de la trop rapide succession des versions. Les éditeurs devraient sérieusement se poser la question sur ces cycles de mise à jour, car les utilisateurs, les développeurs sont loin d'adopter immédiatement les dernières versions. Entre les versions majeures, les patchs, les versions mineures, parfois on en perd son binaire !

> *« Parfois on en perd son binaire ! »*

Mais comme nous adorons les chiffres magiques, ce mois-ci nous en révélons deux : Windows 7 et Java 7. Si pour le premier nous connaissons désormais sa date de disponibilité (fin octobre), pour le second, Sun se prendrait-il pour le nouveau Bizet ? Quoi qu'il en soit, cette version tarde à arriver. Pour nous faire patienter, ou enrager, qui sait ? Les laboratoires pondent une merveille Java 6 Update 14.

Argh... Difficile de croire que cette version s'attendait à une notoriété aussi grande. Car au détour d'un paragraphe, un tollé a secoué si fortement la communauté Java que Sun a d'abord tenté de minimiser l'affaire, puis de se justifier et enfin de faire disparaître la phrase compromettante. Durant quelques jours, une angoissante question se fit jour : Java allait-il devenir payant ? Immédiatement cela m'a fait penser à la double licence (open source / commerciale) ou au modèle économique de certains éditeurs ouverts : édition communautaire, édition professionnelle... Nous sommes peut être à une charnière informatique qui ne veut pas dire son nom. Avec la crise, des projets open source souffrent à trouver des contributeurs et les éditeurs réduisent leurs engagements financiers et humains. Et si l'open source tire certains bénéfices de la crise, cela ne résout pas forcément l'équation économique.

Cette équation a un nom : MySQL. Racheté par Sun, lui-même racheté par Oracle. Ou presque, reste à valider cette proposition de rachat. Cette opération a marqué les esprits. Et nous attendons avec impatience sa confirmation et surtout le devenir des briques technologiques et logicielles de Sun, notamment Java et MySQL. La situation prend une tournure étonnante avec l'annonce par le fondateur de MySQL de la création d'un fork de la base de données : MariaDB. Cette annonce peut s'analyser de deux manières : une positive montrant le dynamisme de l'open source, une négative montrant une segmentation de l'offre et une incompréhension potentielle pour l'utilisateur...

Nous nous attendons à de nouvelles surprises durant l'été ou à la rentrée, notamment du côté de Dell, Apple, Microsoft, pourquoi pas IBM et certainement du côté éditeurs / intégrateurs open source. Cette « concentration » n'avait d'ailleurs jamais réellement arrêté mais se faisait plus discrète. Bref, le mouvement perpétuel de l'évolution du monde informatique.

En vous souhaitant un été binaire ensoleillé

François Tonic
Rédacteur en chef

PROgrammez ! Le magazine du développement

Rédaction : redaction@programmez.com
Directeur de la Rédaction : Jean Kaminsky
Rédacteur en Chef :
François Tonic - ftonic@programmez.com
Ont collaboré : F. Mazué, G. Vidal.
Experts : C. Villeneuve, G. Renard, J.P. Valls, O. Martin, B. Boucard, J. Chable, T. Lebrun, S. Saurel, G. André, A. Berardino, A. Weinbach, P. Guermonprez , R. de Wargny.
Publicité : Régie publicitaire, K-Now sarl
Pour la publicité uniquement :
Tél. : 01 41 77 16 03 - diff@programmez.com
Editeur : Go-02 sarl, 21 rue de Fécamp 75012 Paris - diff@programmez.com
Dépôt légal : à parution - Commission paritaire : 0712K78366 ISSN : 1627-0908
Imprimeur : ETC - 76198 Yvetot
Directeur de la publication : J-C Vaudecrane

Abonnement : Programmez 22, rue René Boulanger, 75472 Paris Cedex 10 - Tél. : 01 55 56 70 55
mail : 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

L'INFO PERMANENTE
WWW.PROGRAMMEZ.COM

PROCHAIN NUMÉRO
N°122
septembre 2009,
parution 1er septembre

✔ **Dossier WEB**
Choisir son **hébergeur.**
Les Techniques de **référencement.**
Monétiser son site.

✔ **.Net 4.0**
Plongée au cœur de la CLR4

■ **Adobe** fait évoluer son offre acrobat.com, en ajoutant une fonction attendue : un tableur en ligne, Tables. Surtout, acrobat.com passe en modèle payant par souscription, avec de nouveaux services professionnels incluant dans l'avenir des fonctions de partage d'espace de travail. On pourra aussi organiser des conférences en ligne. Pour 2010, acrobat.com bénéficiera de nombreuses fonctions : collaboration documentaire, accès mobile, intégration avec les outils Microsoft, accès aux fonctions en ligne pour les développeurs.

■ **L'Afnic** a dévoilé son dernier rapport sur les attaques de DNS (Domain Name System). Le rapport rappelle ce qu'est le DNS, puis énumère les attaques directes ou indirectes dont il est l'objet. Les risques sont réels comme en témoigne le nombre d'attaques contre les serveurs DNS. L'Afnic rappelle aussi les méthodes de sécurisation. Site : http://www.afnic.fr/data/divers/public/afnic-dossier-dns-attaques-securite-2009-06.pdf

■ **Adobe et Nvidia** travaillent ensemble pour faire bénéficier Flash des technologies GPU et donc d'une accélération matérielle du player Flash. Ce sera particulièrement utile dans les vidéos haute définition. Par ailleurs, Nvidia travaille activement à l'initiative Open Screen d'Adobe.

■ **IBM** débloque plus d'un milliard d'euros pour les infrastructures intelligentes en Europe. Ces financements sont destinés à soutenir les efforts des entreprises dans leurs investissements informatiques cette année, en attendant les aides de l'Etat, et à développer l'infrastructure technologique et environnementale du XXIe siècle.

Conférence

Apple booste MacOS X et dope iPhone 3 avec 1000 API !

MacOS X va subir en septembre prochain une mise à jour, la 10.6, que l'on dit souvent mineure. La majeure partie de MacOS X (90%) a été modifiée, optimisée, réécrite, allégée. L'installation a subi une énorme mise à jour : 45 % plus rapide et surtout le système maigrit car l'installation nécessiterait environ 6 Go. La nouveauté majeure est Grand Central qui est une gare de triage entre le système, les applications et les processeurs multi-coeurs. Mais comment faciliter la programmation et aider les applications à dispatcher les threads, à répartir la charge ? Cela passe par des extensions du langage de développement, un moteur multicore, de nouveaux frameworks et API, et des outils adaptés. Cela permettrait de mieux traiter les fonctions des applications. Reste aux développeurs à maîtriser cette technologie... OpenCL est l'autre nouveauté. Il s'agit de tirer parti des cartes graphiques et donc de la fameuse GPU. QuickTime subit lui aussi une profonde évolution : nouveau player, streaming via le protocole http qui permet une grande souplesse dans l'adaptation des flux selon le réseau disponible. Apple prépare activement la sortie de la version 3 du système et d'un nouveau mobile : iPhone 3G S. Le SDK supporte ce modèle. Pour le graphisme, notons aussi l'apparition d'un nouveau processeur interne et surtout la présence d'OpenGL ES 2.0 ! Ce qui va booster le développement 3D ! Pour encore améliorer la maîtrise du matériel, du système, iPhone 3 propose plus de 1000 nouvelles API, de quoi occuper les développeurs.

GPU

Le développement 3D stéréoscopique

Aujourd'hui, le vrai 3D Relief ou plutôt la 3D stéréoscopique, commence tout juste à apparaître dans le grand public et tout particulièrement avec la sortie du kit matériel NVidia : GeForce 3D Vision. Mais pour le développeur qu'est-ce ce que cela change ? Tout d'abord, il faut un matériel spécifique et en particulier au niveau de la carte graphique qui doit posséder une puissance de calcul élevée, et des lunettes spéciales. Dans une approche stéréoscopique, on dispose de deux images : une pour l'œil gauche, une pour l'œil droit. Deux techniques sont utilisées par ce genre de 3D :

- par des lunettes (dites actives)
- par des lunettes passives et des filtres spéciaux sur le moniteur

Actuellement, NVidia supporte le mode passif (via les pilotes), les lunettes actives sont en cours de développement. Donc vous aurez un écran avec filtres polarisés, c'est obligatoire. Ce n'est donc pas demain que tous les ordinateurs pourront faire de la 3D Stéréo... Chez NVidia, les seuls pilotes officiels sont ceux pour Vista. Ce type de 3D nécessite une autre approche de la 3D et de l'interface. Un des problèmes est de placer les objets d'interfaces (puis de faire un rendu) selon le point de vue de la camera, de l'utilisateur. Un peu comme en 3D quand le créateur organise la lumière, les angles de rendus, etc. Et d'autre part, il faut aussi considérer la place et le rendu des objets 2D tels que les menus, les boutons, les textes, etc. L'une des difficultés est de bien régler la place des objets selon la profondeur de l'écran.

Site : http://www.nvidia.fr/object/GeForce_3D_Vision_Main_fr.html

ComponentSource
The Definitive Source of Software Components
www.componentsource.com

FarPoint Spread for ASP.NET | à partir de € 642

FarPoint

Composant de feuille de calcul ASP.NET haute performance personnalisable.

- Nouvelles fonctions : extensions AJAX, impression vers PDF, éditeur de modèle de ligne, assistant de démarrage rapide, nouveaux types de cellules, etc.
- Modes liés et non liés (aucun ensemble de données nécessaire), AJAX, import/export Microsoft Excel natif, édition en cellule, redimensionnement client, etc.
- Plus de 300 fonctions de calcul intégrées

Janus WinForms Controls Suite | à partir de € 562

Janus systems

Ajoutez les interfaces de style Outlook à vos applications WinForms.

- Janus GridEX for .NET (grille de style Outlook)
- Janus Schedule for .NET et Timeline pour .NET (calendrier et journal de style Outlook)
- Janus ButtonBar for .NET et ExplorerBar pour .NET (barres de raccourcis de style Outlook)
- Janus UI Bars & Ribbon Control (menus, barres d'outils, commandes ruban Office 2007)
- Avec style visuel Office 2007 de toutes les commandes

Aspose.Total for .NET | à partir de € 1 387

Tous les composants Aspose .NET et toutes les nouvelles versions dans l'abonnement annuel.

- Gestion par programmation des formats de fichier: Word, Excel, PowerPoint, Project, PDF, Flash
- Composants graphiques, messagerie, correction ortho., création codes barre, requêtes ad hoc, modèles de récurrence et flux de travail
- Ajoutez facilement un éditeur texte complet des contrôles de grille à votre application
- Ne nécessite pas Microsoft Office

DXperience Enterprise | à partir de € 902

DevExpress

Tous les outils DevExpress ASP.NET, WinForms et IDE Productivity en un.

- Abonnement de 12 mois pour tous les produits et mises à jour Developer Express et accès aux versions bêta en développement actif
- Composants et outils : grilles, entrée de données, outils d'écriture de code, analyse de données, graphiques, navigation/disposition, planification, solutions reporting, bibliothèques d'impression, outils de remaniement, bibliothèques ORM

© 1996-2009 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

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

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

■ Le **WC3** mise totalement sur le web sémantique qui constitue le futur immédiat de notre web. Le consortium a apporté son soutien durant la conférence SemTech 2009 (mi juin à San José). Ont été abordés : la représentation des connaissances sur le web, l'organisation de ces connaissances ou encore comment décrire le contenu pour faciliter la diffusion. Le W3C recommande l'adoption des standards SKOQ et OWL, en évolution régulière.

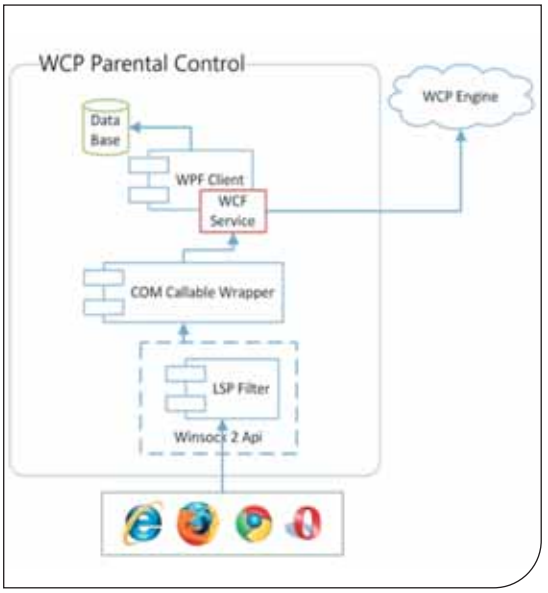
■ **Red Hat** annonce de nouveaux outils Java Application Platform autour de JBoss Open Choice. La stratégie JBoss Open Choice est la réponse imaginée par Red Hat au paysage en perpétuelle évolution des applications Java pour l'entreprise, qui regorge de toujours plus de modèles de programmation et de déploiement. La stratégie JBoss Open Choice va permettre aux entreprises de se doter dès aujourd'hui des derniers développements de la communauté Java avec la garantie d'investir durablement puisqu'elles sont certaines de pouvoir profiter des prochaines évolutions Java pour l'entreprise. De nouveaux serveurs d'applications seront proposés pour rendre encore plus souple la plate-forme et seront compatibles avec Seam, Spring et GWT.

■ **RIM** sort le nouveau serveur BlackBerry Enterprise Server 5. Cette version renforce considérablement l'administration, la gestion des autorisations, facilite et centralise la distribution des applications et de leurs mises à jour. D'autre part, il inclut de nouvelles fonctions de surveillance de l'état du système et de la plateforme, et une meilleure définition des politiques d'entreprise. Un module spécifique a été créé, *Transporter,* pour faciliter la mise à niveau de son serveur RIM. Cette édition conforte le positionnement entreprise du Blackberry.

ImagineCup 2009

Les étudiants savent innover

En mai dernier, la finale française d'Imagine Cup 2009 (dont la finale mondiale aura lieu au Caire) a une fois de plus démontré toute l'imagination de nos étudiants en informatique. Et surtout leur capacité à intégrer très rapidement les nouvelles technologies comme le cloud computing. L'équipe WikiChildProtect de l'école SUPINFO - Toulouse dirigée par Olivier Courtois et Julien Dollon a été qualifiée parmi les 5 équipes mondiales (catégorie software design, special award) pour le Caire. Elle a intégré dans son modèle applicatif les technologies Azure Services Platform. Le projet propose un nouveau concept de protection parentale sur Internet. L'un des points fort de ce projet est son moteur communautaire appelé WCP Engine. Ce moteur est capable de stocker les avis de la communauté de parents engagés, mais aussi d'interroger les contrôles parentaux déjà existants afin d'obtenir la meilleure sécurité pour vos enfants. Il fallait pouvoir adapter ce système à une montée en charge conséquente, il a donc fallu s'appuyer sur Windows Azure, .NET Services, SQL Services et SharePoint online. Le contrôle parental réalisé en WPF s'appuyait sur un filtre LSP et interrogeait le service WCF sur Azure afin de savoir si le site était à la fois approuvé par la communauté de parents, mais aussi par les autres contrôles parentaux du marché. Malgré la jeunesse de la technologie,



et l'impression d'être peu nombreux à avoir porté les applications sur le cloud, c'est un réel plaisir et une révolution d'exploiter ces services. D'un point de vue scolaire, Azure touche un marché d'étudiants très demandeurs : aujourd'hui, faire de la veille technologique sur .NET ou tout simplement publier un blog « BlogEngine » ou « Community Server » c'est avoir sa machine dédiée avec Windows Server. L'étudiant développeur, se voit obligé d'administrer une machine, qui plus est, parfois coûteuse.

Outils

Appcelerator s'occupe du desktop et du mobile

Comment créer rapidement et simplement des applications Web (html, javascript, css, Python, Ruby) sans se soucier, du terminal cible ? Appcelerator relève le défi pour desktop et mobile. Avec Titanium Desktop, vous créez des applications pour Windows, Mac, Linux avec un seul code. L'environnement utilise les standards du web, tout en permettant d'appeler les fonctions natives mais sans avoir besoin de les apprendre. La création des packages se fait simplement par configuration. Titanium Mobile est la déclinaison pour les applications Android et iPhone. Appcelerator Entourage permet d'intégrer de la SOA et des services web directement dans ses applications. C'est une approche assez originale. La version finale devrait intervenir dans les prochaines semaines. Reste à voir le succès que l'environnement peut obtenir…

Site : http://www.appcelerator.com

WINDEV WEBDEV WINDEV Mobile
Développez 10 fois plus vite
Nouvelle version 14
PLATEFORME PROFESSIONNELLE DE DÉVELOPPEMENT (AGL)
Internet, Intranet, RAD, WEB 2.0, PHP
Lien à toutes les bases de données
WEBDEV®
Ajax en 1 clic
WEBDEV 14 est l'environnement de développement professionnel qui permet de développer jusqu'à 10 fois plus vite tous les types de sites et d'applications reliés aux données de votre entreprise.
Le WEB 2.0 est facile: l'activation d'AJAX dans vos sites s'effectue naturellement; un simple clic dans l'éditeur de code indique que le code à exécuter est de type «Ajax». WEBDEV 14 est certainement le seul environnement au monde à proposer autant de souplesse et de puissance.
Un clic, et Ajax est actif
Sous l'éditeur de code de WEBDEV 14: un clic et le traitement programmé devient «Ajax»
WEBDEV 14 gère le cycle complet de développement et d'administration: langage L5G, générateur de code PHP, débogueur, Webservices, gestionnaire de sources, installateur, base de données SQL intégrée et lien avec toutes les bases du marché, composants, éditeur d'états PDF et code-barres, règles métier, dossier, outils de déploiement et d'administration...: tout est inclus, en français.
Vous aussi, réalisez vos sites WEB 2.0 10 fois plus vite... avec WEBDEV 14.
UN OCEAN DE WEB 2.0
VOTRE CODE EST MULTI-PLATEFORMES:
Windows, .Net, Java, PHP, J2EE, XML, Internet, Ajax, Pocket PC, SmartPhone, Client riche ...
DEMANDEZ LE DOSSIER GRATUIT
252 pages + DVD + 112 Témoignages.
Tél: 04.67.032.032 ou 01.48.01.48.88
info@pcsoft.fr
Logiciel professionnel
Fournisseur Officiel de la Préparation Olympique
www.pcsoft.fr
PC SOFT 25e ANNÉE
Logiciel professionnel Document non contractuel. Support technique gratuit 15 requêtes sur la version en cours de commercialisation. WINDEV, WEBDEV et WINDEV Mobile sont compatibles.

Communication

Google réinvente la collaboration avec Wave

Google Wave est le futur outil de collaboration en ligne de l'éditeur. Si les premiers pas peuvent être déconcertants, à l'usage, l'outil se révèle intéressant et d'une grande souplesse. Il s'agit d'un environnement de collaboration unifiée mêlant aussi bien le mail, la messagerie instantanée que les documents. La fenêtre principale se découpe en trois grandes zones : la boîte aux lettres avec les messages, les fils de discussions à droite, la liste des personnes connectées et les outils à gauche. Le principe repose sur la « wave ». Voici comment Google présente son nouvel outil : « *L'objectif de Google Wave est de concevoir, à partir de zéro, une plate-forme de communication unique permettant une interaction plus efficace entre les internautes. Avec Google Wave, plusieurs internautes pourront échanger en temps réel dialogues, photos, vidéos, cartes, documents et d'autres informations au sein d'un espace de communication unique et communautaire, appelé « wave »*. Tous les internautes invités à participer à une « wave » pourront insérer une réponse ou directement modifier la « wave », et voir instantanément ce que tapent leurs collaborateurs et même faire apparaître une « wave » sur un blog ou un site web, dont le contenu sera instantanément mis à jour au fur et à mesure des modifications sur la plateforme. Le but est de permettre aux internautes de communiquer et de collaborer de façon plus riche, plus instantanée et plus centralisée. »

On peut reprendre à la volée un fil de conversation vieux d'une heure et reprendre la conversation à cet endroit. On peut y ajouter des cartes, etc. Le but est réellement d'ajouter une interactivité la plus forte, la plus totale dans la collaboration. On voit à l'écran qui écrit, qui modifie un contenu, une discussion. On peut partager en quelques clics des images, des cartes et suivre là aussi les modifications apportées. C'est véritablement une petite révolution dans la manière de collaborer. Actuellement, Wave est en développement interne, mais l'ambition de l'équipe est de sortir une version utilisable dès cette année !

Des API complètes

Côté développeur, on disposera d'API pour développer des extensions pour Wave. On dispose de deux éléments : Extensions pour créer des extensions « robots » pour automatiser les tâches, construire des gadgets pour les utilisateurs Wave et Embed pour rendre son site « wave compatible ». Pour le développeur, une wave est un fil (thread dans les documents) de conversion à une ou plusieurs personnes avec un contenu dynamique et un historique. La wavelet, un « composant » d'une wave est une sorte de conteneur ayant des messages (ou blips dans l'appellation Google). Le wavelet est une unité basique d'accès aux données dans une wave. Côté langage, on retrouve les classiques Google : xml (le cœur même de wave), javascript, Python. Quid de Java et .Net ? Cela est possible via les robots wave. Les robots communiquent avec Wave en utilisant des mécanismes html, xml, Json, etc. Dans ce cas, peu importe le langage d'origine. Dans la première version de l'API Wave, les robots, pour permettre l'intégration doivent être en App Engine et codés en Java ou Python. Mais cette limitation devrait sauter dans le futur. Autre limitation, l'interface que l'on ne peut pas personnaliser via des skin. Mais cette possibilité est espérée dans une future version (dixit Lars Rasmussen), co-fondateur de Wave avec son frère Jens. L'ouverture ne s'arrêtera pas là car Wave devrait aussi permettre à des éditeurs, des développeurs tiers de s'intégrer à d'autres outils comme des annuaires d'entreprises (via une API). L'équipe Wave travaille aussi à la version mobile qui est déjà fonctionnelle (pour Android et iPhone). D'autre part, Wave s'appuie sur un protocole ; le Google Wave Federation Protocol. Il s'agit d'un projet open source. Les principes sont les suivants :

- Wave est un réseau ouvert, dans lequel chacun peut devenir un fournisseur de wave et interopérer avec le réseau public
- Wave est un modèle de réseau distribué : le trafic étant basé sur du P2P et non via un serveur central

Quelques adresses utiles pour aller plus loin :
http://googlewavedev.blogspot.com/
http://www.waveprotocol.org/Home
http://wave.google.com/

agenda \\

JUILLET

• Du 06 au 07 juillet 2009, Grand amphithéâtre de l'EPITA, les **PHP Days "Industrialisez votre PHP!"** vous permettront d'acquérir les compétences nécessaires à une utilisation professionnelle de PHP.Organisé par Alterway et Anaska.
www.phpdays.com

• Le 6 juillet 2009 le Lorraine JUG (Groupe des Utilisateurs Java de Lorraine) et le YaJUG (Groupe des Utilisateurs Java de Luxembourg) organisent une soirée de conférences « **JavaFX - Build your own Javabased Rich Internet Applications** », sur la péniche théâtre « La Lucarne » au plan d'eau de Metz.
http://www.jugevents.org/jugevents/event/16966

• Le 07 Juillet 2009
CULTURE SOA: Relevez tous les défis avec agilité ! IBM Forum,Tour Descartes - 2 avenue Gambetta, Paris la Défense. 4e rendez-vous annuel des solutions et meilleures pratiques SOA, marqué par l'intégration complète de ILOG, est celui de l'industrialisation (réservé aux clients et prospects IBM) http://www-01.ibm.com/software/fr/culture_soa/agenda.html

Symfony Live : PHP fait son show développeur !

La première édition de « Symfony live » s'est déroulée les 11 et 12 juin 2009 à Paris, organisée par Sensio Labs, en partenariat avec l'AFUP (Association Française des Utilisateurs de PHP). Environ 200 personnes étaient présentes. Dans une ambiance très chaleureuse se sont croisés les développeurs, développeuses, responsables de sociétés et les équipes de Sensio Labs. Pendant ces 2 jours, de nombreuses conférences étaient organisées touchant les thèmes techniques, des retours d'expériences et aussi la possibilité de parler directement à la Core Team qui réalise le framework Symfony. Les sujets, très intéressants couvraient par exemple PHPEdit, Eclipse PDT, les réseaux Sociaux, Dailymotion, YDN de Yahoo, l'Express, les migrations, etc.

Et de nombreuses nouveautés ont été dévoilées en même temps comme :

- Sympal CMS réalisé par le Lead developer de l'ORM Doctrine, le premier CMS officiel à partir du framework
- Symfony 1.3 : prévue pour l'automne prochain

- Symfony 2 : la nouvelle mouture verra le jour en 2010 et sera 7 fois plus rapide par rapport au framework actuel
- Une évolution importante à retenir pour Symfony 2, concernera une configuration améliorée de ses composants.
- La dernière version de la branche 1 (avant lSymfony 2.0) sera maintenue pendant 3 ans.

Durant l'événement, des conférences techniques ont été aussi mises en avant, par exemple l'utilisation des bonnes pratiques du framework Symfony et du javascript, les tests unitaires et fonctionnels… Toutes les présentations sont disponibles sur le site de l'événement (http://symfony-live.com).

Les orientations pour 2009 du framework Symfony, sont axées vers la consolidation de la communauté et aussi la documentation. Sur ce point, plusieurs ouvrages, édités par Sensio Labs sont disponibles. Eyrolles a sorti un ouvrage en français sur ce framework.

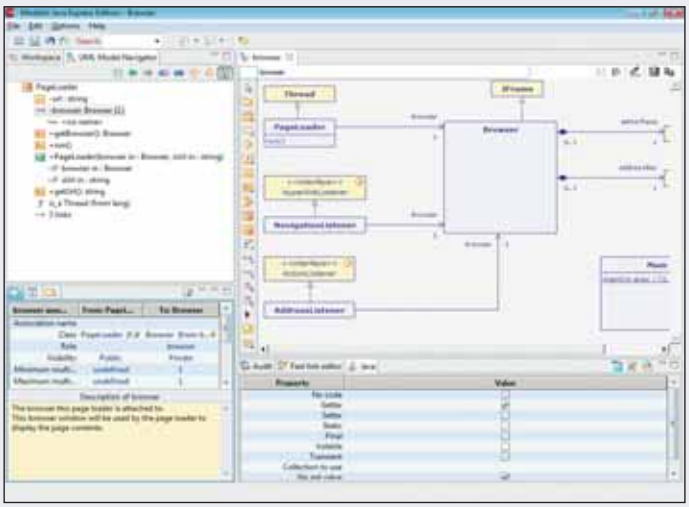
■ Christophe **Villeneuve**
Auteur du livre " PHP & MySQL-MySQLi-PDO, Construisez votre application *", aux Editions ENI.*
Ingénieur chez Ingeniweb du groupe AlterWay.

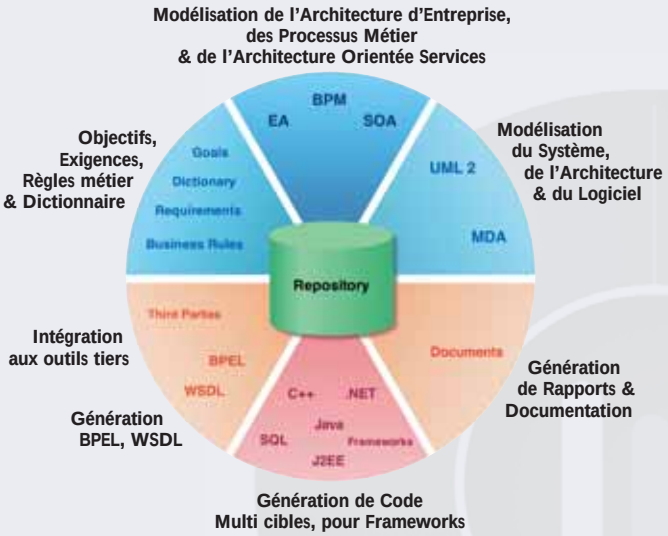
Modelio : Nouvel outil de modélisation

Modéliser sur toute la portée de l'entreprise pour un alignement SI/Métier

Modéliser n'a jamais été aussi abordable et productif

- Ergonomie simple et productive
- Support intégré de UML, BPMN, BPM, l'Architecture d'Entreprise, L'analyse des besoins, … dans un seul référentiel
- Travail de groupe distribué, intégré à SVN/Subversion
- Des générations sur étagère adaptables pour toutes les cibles
- Support MDA : transformation, extensibilité et adaptabilité





Modelio est disponible en trois éditions

- **Free :** Un outil complet de modélisation gratuit !
- **Express Java :** Un outil de développement Java guidé par le modèle de haute performance 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

MODELIOSOFT

www.modeliosoft.com

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

Community One / JavaOne 2009 : du cloud, du JavaFX, une polémique, pas de Java 7

Comme chaque année, le monde Java se réunit à San Francisco sous l'égide de Sun. Mais avec la crise ambiante et le rachat de Sun par Oracle, bien des questions restaient sans réponses.

Dès l'arrivée au Moscone Center, début juin, ceux qui prévoyaient un vide relatif dû à la crise en ont été pour leurs frais : foule dense et studieuse, remplissant les auditoriums et les salles de conférences (300 sessions, 600 intervenants), files interminables à l'entrée des labos. Bien sûr l'Open Source reste au cœur de la philosophie de la communauté Java, mais il prend une dimension nouvelle : on parle désormais de l' « Ingénieur Citoyen ». Adieu science sans conscience, bienvenue à la responsabilité sociale (et écologique) du développement technique. Avec les Universités Open Source, les projets étudiants dépassent les frontières pour organiser des conférences et compétitions amicales permettant de resserrer les liens. Au début de la semaine Sun, trois absents faisaient sentir leur présence : Adobe pour Flash, Apple pour l'AppStore et Oracle. Si pour Oracle, la communauté repartira mi-figue mi-raisin, pour Adobe et Apple, nous avons ressenti leur poids sur la stratégie et l'orientation. Pour les contrer, Sun ressort JavaFX et tente la riposte avec JavaStore !

Java en chiffres

Comme toujours dans les grand' messes, la communauté adore les chiffres, la preuve :

- 1 milliard de plates-formes Java
- Applications EBay gérant 60 milliards de dollars de ventes en 2008
- 300 000 transactions par seconde au Chicago Exchange, 3,5 milliards de transactions / jour
- Installation de Java sur les contrôleurs de télévision, permettant l'intégration des programmes et des flux de données internet

En bref, Java est partout.

Peu d'annonces fortes, mais prometteuses

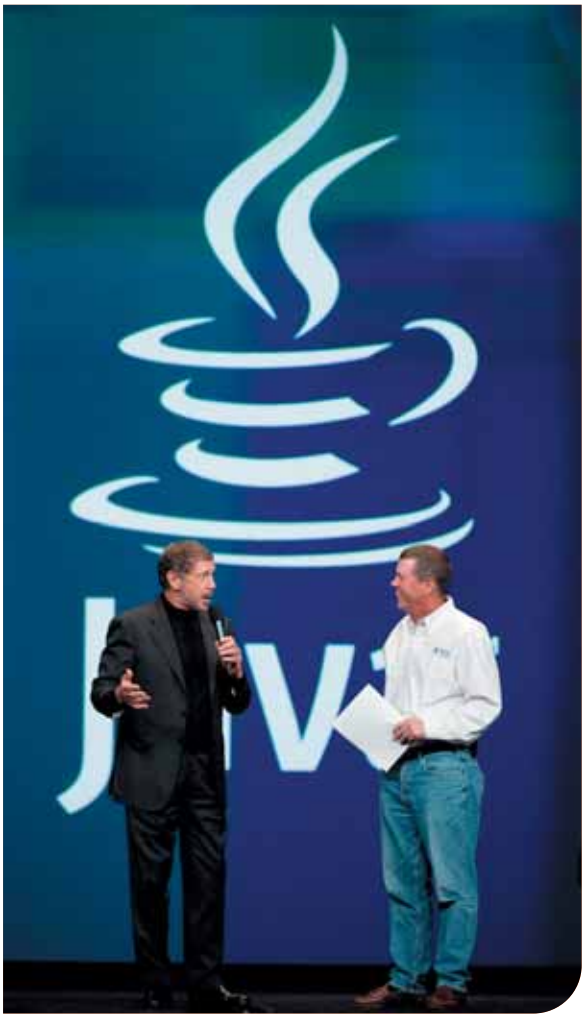
La grosse surprise, ou presque, est l'ouverture du portail de la boutique en ligne pour les applications Java. Apple et son App Store font décidément école ! Java Store se destine aux applications Java et Java FX. Il s'agit encore d'une pré-version et les applications payantes ne sont pas encore possibles. Au-delà de Java Store, plusieurs composants sont intéressants à remarquer :

• Java WareHouse : repository central pour y déposer les applications et les proposer pour la distribution, avec un processus de soumission.

• Project Vector : projet crucial car il s'agit d'un socle pour les marchés en ligne, les boutiques en ligne pour les applications Java / JavaFX

D'autre part, pour le moment, le Java Store est en bêta privée uniquement pour les Américains. Le reste du monde devra attendre 2010 et nous ne connaissons pas le mode de rémunération qui sera pratiqué...

Java FX évolue avec la version 1.2 qui se dote d'une interface utilisateur multi- plate-forme refondue avec de nouveaux objets, d'un support du streaming, d'une amélioration des performances, de nouveaux outils comme un outil de création. Cette 1.2 se concentre aussi sur l'outillage de développement qui était un des points faibles (disponibilité fin 2009). L'intégration de code Java dans les applications JavaFX se voit facilitée,



Larry Ellison et Scott Macnealy pendant la keynote.

et on commence à voir les premières applications avec JavaFX TV et une application de discussion en ligne (DarkChat) développée spécialement pour JavaOne. D'autre part, Glassfish Portfolio propose une pré-version de Java EE 6 et de Glashfish 3 (qui devrait arriver en version finale cet été). Si Java 7 a brillé par son absence, Java SE 6 Update14 a suscité une vive polémique à cause du nouveau ramasse-miettes, alias G1. Dans la note de version, il était clairement spécifié que la mise en production d'une application utilisant G1 était possible uniquement en ayant un support Java SE for Business. Trois raisons ont été avancées pour cette « clause » indiquée noir sur blanc sur

la note de version officielle de la mise à jour Java :

- contrôler le passage en production de G1 pour éviter les problèmes, vu que les premières applications sont stratégiques (ex. : Chicago exchange)
- permettre de gagner de l'argent et financer les développements de G1
- enfin, Java6E14 est toujours en développement.

Durant la JavaOne, les conditions d'utilisation furent tout bonnement modifiées en supprimant les références à ce support obligatoire. Reste que la polémique a enflammé la communauté. Cette Java 6 Update 14 propose aussi : le support 64 bits et les améliorations de GlassFish intégrant une nouvelle interface utilisateur et un portail moderne offrant un VPN SSL.

Le cloud à l'honneur

Pourtant dans les sessions et les allées de la conférence, trois sujets majeurs se distinguaient : Cloud Computing, OpenSolaris et la virtualisation. Cet été verra la naissance officielle de SUN Cloud. L'architecture permettra de définir graphiquement les éléments nécessaires à l'environnement souhaité pour son application, et reposera sur les techniques de virtualisation. Une palette d'outils permettra d'analyser en temps réel les performances du système, et l'on pourra utiliser le service SUN Cloud pour sauvegarder un PC voire un centre de données complet.

L'évolution des discours sur le Cloud Computing s'associe à un glissement sémantique. La définition devient à la fois plus floue – recherche-t-on seulement une mutualisation des coûts, une économie d'échelle ou simplement une élasticité du système facilitant l'expansion – et plus précise avec les différenciations entre :

- Software as a Service
- Platform as a Service
- Tools as a Service
- Infrastructure as a Service

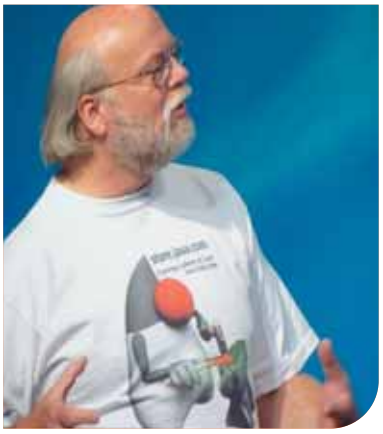
Avec pour chacun de ces points une offre adaptée. En termes d'application on trouvera :

- Base de Données Cloud
- Collaboration Cloud
- Application Cloud
- SharePoint Cloud

Les restrictions augmentant avec le niveau d'abstraction. Tout cela n'est possible qu'avec une architecture solide de virtualisation. Celle de SUN (VDI 3) s'appuie sur Virtual Box disponible dans tous les environnements, ainsi que sur les terminaux « légers » SUN Ray. VDI se décompose en 3 niveaux :

- Identification de l'utilisateur
- Gestion virtuelle de l'environnement utilisateur
- Gestion des données (via iSCSI)

L'identification et la communication avec le terminal client tolèrent des réseaux lents en optimisant le rafraîchissement de l'écran (on m'a parlé de géométrie fractale pour l'optimisation de l'affichage). Virtual Box intègre nativement RDP (Remote desktop protocol). Les développements annoncés concernent le travail déconnecté par transfert de la machine virtuelle sur le poste de travail, le support de la 3D, MAC OS X et l'installation par « boîte » clés en main. Tout cela devrait être disponible dans la mise à jour prévue au quatrième trimestre et celle de 2010.



James Gosling durant la keynote

Le nouveau Java FX

Le centre de conventions

Le patron d'Oracle sur scène

La keynote laissera un souvenir fort au monde Java. Pour la première fois, le patron d'Oracle, Larry Ellison monta sur scène aux côtés des patrons actuels de Sun qui ne devraient pas le rester bien longtemps. Et la mine triste et la petite forme de Scott McNealy, figure emblématique de Sun, laissaient présager de son sort : son départ pur et simple. Fidèle à son image, Ellison a pris à bras le corps l'ensemble des « sujets qui fâchent », a renouvelé son support aux équipes de SUN, et réitéré l'intérêt d'Oracle pour Java en faisant remarquer plusieurs points :

- A part la base de données, l'ensemble des produits Oracle fonctionne en environnement Java
- Le budget combiné en R&D de Sun et Oracle s'élève à près de 5 milliards de $, soit dans les 10 premiers budgets R&D mondiaux.
- L'arrivée de Google Android représente une opportunité, Sun et Oracle vont aussi se lancer dans les objets java.

Mais reste à Oracle à finaliser le rachat de Sun et surtout à rassurer la communauté. Car nous ne savons pas ce qui se passera à l'avenir. La polémique autour de G1 a suscité beaucoup de craintes que Sun a eu bien du mal à apaiser.

The show can go on!

■ Gilbert **Vidal**

Visual Studio 2010 : visite guidée

Microsoft Visual Studio

Après une première version lourde et lente, Microsoft propose depuis quelques semaines une version bêta de Visual Studio 2010. Celle-ci est exploitable, plus légère que la précédente même si elle demeure incomplète au niveau fonctionnel. Faisons le tour du propriétaire.

Visual Studio 2010 est toujours moins dissociable des langages qu'il intègre. Il revêt une multitude de nouveautés que nous pourrons d'ores et déjà classifier en différentes catégories :

• Nouveautés de l'IDE – Environnement de développement
• Nouveautés des langages : VB, C#, C++ et le petit dernier F#
• Nouveautés du .NET Framework 4.0 et de la CLR 4.0
• Nouveautés des outils et compléments
• ...

Avant d'aborder les nouveautés, nous allons déjà voir comment se procurer cette version. **Conseil :** préférez une installation dans une machine virtuelle pour des raisons de stabilité et éviter tout problème de corruption système.

Vous procurer Visual Studio 2010

Rendez-vous pour cela à l'une des deux adresses suivantes :

- http://msdn.microsoft.com/fr-fr/vstudio/dd582936(en-us).aspx
- http://www.microsoft.com/visualstudio/en-us/products/2010/default.mspx [Fig.1]

Vous aurez ainsi le loisir de télécharger l'édition professionnelle, la Team Foundation Server ou encore la Team System, vous retrouverez l'ensemble des pré-requis depuis ce même site internet. Vous aurez également la possibilité de télécharger le .NET Framework 4.0 uniquement si vous le souhaitez. Pour le moment, .Net 4 n'est pas inclus par défaut.

Note : Si vous souhaitez acquérir Visual Studio 2008, pensez au service Software Assurance proposé par Microsoft vous permettant de bénéficier tout de suite de la version 2008 de Visual Studio tout en vous donnant la garantie de passer à la version 2010 sans coût supplémentaire en achat de licence. Retrouvez plus d'informations relatives au software assurance depuis l'adresse suivante : http://www.microsoft.com/France/acheter/sa/default.mspx

Visual Studio étant maintenant en votre possession, abordons pas à pas quelques nouveautés de la version 2010.

Fig.1

Nouveautés de l'environnement de développement

Le téléchargement étant maintenant en cours, vous aurez largement le temps de poursuivre la lecture de ce dossier dédié à Visual Studio 2010. Abordons maintenant quelques nouveautés spécifiques de l'IDE (environnement de développement). Comme vous le constaterez, il y a beaucoup d'évolutions par rapport à Visual Studio 2008. [Fig.2]

Fig.2

Fig.3

webperformance load tester

Testez facilement les performances de vos applications Web et Déployez en toute confiance

Développement automatisé de Cas-Tests et Test de Charge

Rapports d'Analyse des Tests de Performance et de Charge

Analyse des Serveurs et de l'impact sur les Performances

Interface Utilisateur et Rapports en Français

Licence à partir de 1.995 € H.T.

Evaluation gratuite sur http://www.kapitec.com/Pub/WP?id=121

Bénéficiez de notre assistance technique pendant les 15 jours d'évaluation.

kapitec Software

Power Your Web Projects

Tél.: 05 34 27 90 03
sales@kapitec.com

Web Performance Load Tester est une marque déposée de Web Performance, Inc.

Dès l'ouverture de Visual Studio 2010, les changements majeurs se font sentir. Effectivement, l'intégration de la technologie WPF (Windows Presentation Foundation) est bien présente et le mode dit « Dog food », qui consiste dans le cas de Microsoft à consommer sa propre technologie, ici WPF, pour construire ses logiciels (Visual Studio, Surface, Windows7, ...), prend tout son sens ! Observez au travers de la capture de la [Fig.3] les modifications impactées au sein de la page d'accueil de Visual Studio 2010.
Cette page d'accueil, exploitant la technologie WPF, très différente de celle que l'on connaissait au sein de Visual Studio 2008, est complètement paramétrable et administrable. Vous pourrez ainsi créer aisément votre propre page de démarrage en y ajoutant les informations relatives à votre équipe de développement : liens vers les ressources internes, blogs internes, consommation de flux RSS divers et variés, ...
Cette page de démarrage devient ainsi la vôtre et vous permet de faire plus encore de Visual Studio votre outil unique, bien au-delà du fait de vous permettre de programmer !
Avant d'attaquer la modification de notre page de démarrage, étudions rapidement ses nouveautés comme la nouvelle méthode d'administration de vos projets en page de démarrage. Voyez dans la capture suivante la nouvelle capacité d'administrer comme vous le souhaitez vos projets affichés : [Fig.4]. Vous pouvez constater la présence de deux nouvelles icônes vous permettant respectivement :

- **Croix Rouge :** de supprimer un projet de votre liste des projets affichés
- **Punaise :** de figer le projet associé dans la liste des projets affichés.

Quant au processus de modification de votre page de démarrage, celui-ci sera excessivement simple puisqu'il vous suffira de modifier le code XAML de votre page. Retrouvez l'ensemble des explications de modification de votre page d'accueil de Visual Studio 2010 depuis le blog de la Team Visual Studio 2010 à la page suivante : http://blogs.msdn.com/vsxteam/archive/2009/05/20/visual-studio-2010-beta-1-start-page-customization.aspx
N'oubliez cependant pas d'autoriser la personnalisation de celle-ci au sein des options de Visual Studio : [Fig.5]
Un autre effet direct du mode Dog Food de la technologie WPF dans votre IDE favori est l'intégration intuitive du mode vectoriel et la capacité de gérer le Zoom sur votre code et la partie design ! L'exploitation de la molette de votre souris combinée à la touche CTRL de votre clavier dans le contexte d'écran de code au sein de Visual Studio 2010 vous permettra de profiter au mieux de cette nouvelle fonctionnalité apportée par la technologie WPF. Pratique et plutôt plaisante à l'utilisation. Un des autres avantages de l'exploitation de la technologie WPF au sein de cette nouvelle version de Visual Studio, au-delà de la capacité de personnalisation de la page de démarrage, est de vous permettre de personnaliser le rendu de vos écrans (onglets) d'affichage de votre code-source : [Fig.6]
Cette opération peut se faire par modification des Templates WPF (XAML) au sein de l'IDE.

Le multi-écran

D'autres fonctionnalités, comme la navigation simplifiée au sein de votre code, ou encore la mise en surbrillance automatique de l'ensemble des itérations d'un même symbole au sein d'une page de code devrait plus que vous ravir. [Fig.7]
C'est une autre nouvelle approche, plus intuitive, de l'environnement de développement, et effectivement, Microsoft se devait d'améliorer, de modifier l'éditeur pour gagner en confort et en visibilité du code, du projet, bref améliorer le quotidien du développeur ! Mais ce n'est pas tout, les bonnes nouvelles ne venant jamais seules, l'équipe en charge du développement de l'interface a fortement travaillé sur la lisibilité du code au sein de l'IDE pour le développeur.
Observez la capture suivante présentant par exemple la capacité pour Visual Studio 2010 de proposer un affichage intelligent en double écran permettant d'avoir d'un côté votre IDE et l'ensemble des outils associés et de l'autre votre fenêtre de code en plein écran : [Fig.8]
Cette nouveautés, bien que non orientées code, risquent de nous apporter un certain degré de confort de lecture dans notre travail. A essayer de toute urgence !
Et ce n'est pas tout, abordons maintenant deux nouveautés propres à l'IDE mais plus orientées code que sont le Quick Search (recherche rapide) ou encore le Hierarchy Call (l'appel hiérarchique).

- ***Quick Search :*** extension des fonctionnalités de recherche existant au sein de Visual Studio 2010, permet d'effectuer une recherche globale

Fig.4

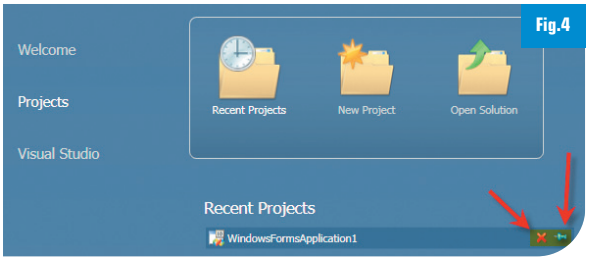


Fig.5

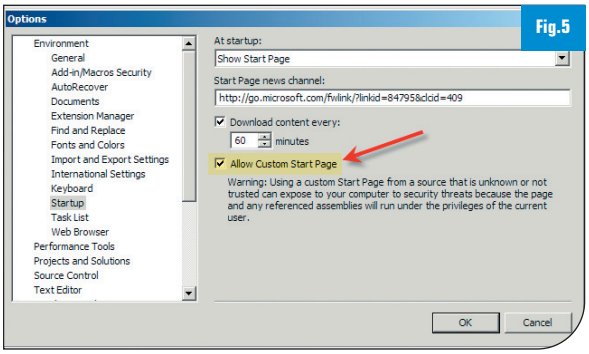


Fig.6

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;

namespace SimpleDataStructures.Tests
{
    public class SimpleStack
    {
        ArrayList _list = new ArrayList();

        public bool IsEmpty
        {
            get { return _list.Count == 0; }
        }

        public void Push(int p)
        {
            _list.Add(p);
        }
    }
}
```

ou par capitalisation au sein de l'ensemble des symbols de votre projet ou solutions Visual Studio 2010. [Fig.9]

- ***Hierarchy Call :*** Cette nouvelle fonctionnalité au sein de Visual Studio 2010 vous permettra de retrouver rapidement l'ensemble des « Call To » ou « Call From » d'une méthode précise. [Fig.10]

Dans la lignée des optimisations code de l'IDE, un des points auquel il nous est de plus en plus difficile d'échapper, et ce n'est pas un mal, est la mise en œuvre de projets de tests unitaires !

Visual Studio 2010 incorpore ainsi quelques nouvelles fonctionnalités assez intéressantes relatives aux tests unitaires, et pour cause, vous aurez la capacité de travailler en rétro-engineering depuis vos projets de tests afin de créer automatiquement vos classes ou vos méthodes au sein de votre projet à tester, avec la nouvelle fonctionnalité de génération incorporée au sein de Visual Studio 2010 comme présenté dans la capture suivante : [Fig.11]

Vous pourrez ainsi, au travers de cette nouvelle interface, définir la portée, le type ou encore le nom de votre type ainsi que le projet cible de génération. Après avoir abordé plus précisément quelques nouveautés majeures de l'IDE Visual Studio 2010, abordons maintenant, une synthèse générale des nouveautés.

Les autres nouveautés

Effectivement, Visual Studio 2010 apporte son lot de nouveautés en plus ou en parallèle de celles propres au Framework 4.0 et à la CLR 4.0. Balayons-les en catégories respectives.

Amélioration de l'expérience utilisateur

Comme vous avez pu le constater dans la première partie du dossier, la version 2010 de Visual Studio apporte au développeur une interface moderne et un enrichissement de l'expérience utilisateur permettant une meilleur compréhension du contexte courant :

- Meilleure organisation de l'interface utilisateur
- Réduction de l'encombrement et de la complexité
- Amélioration de l'éditeur
- Meilleur support des documents et fenêtres flottantes
- Amélioration de l'accès à vos ressources [Fig.12]

Programmation parallèle

La gestion de la programmation parallèle est simplifiée au sein de Visual Studio 2010, ainsi les développeurs d'applications en code natif ou managé peuvent construire aisément des applications exploitant ces notions grâce aux nouveautés suivantes :

- Support de la programmation parallèle au sein de l'IDE
- Les extensions parallèles du .NET Framework permettent un support des données impératives et tâches parallèles, du parallélisme déclaratif de données et plus encore
- Gestion du parallèle debugging

Le parallélisme est une des grosses nouveautés de la CLR 4.0 que nous aborderons en détail dans **le prochain numéro.**

Amélioration de la gestion du cycle de vie des applications

Visual Studio Team System 2010 propose une série de nouvelles fonctionnalités dans le cadre de gestion du cycle de vie de vos applications. Ces fonctionnalités sont autant destinées aux développeurs, architectes, chefs de projets ou encore testeurs fonctionnels avec la solution complémentaire portant le nom de code Camano et aujourd'hui découpée en deux outils distincts :

- Microsoft Visual Studio Lab Management 2010 Bêta 1 : ce nouvel outil permettra de générer et gérer des environnements virtuels de tests?

http://www.microsoft.com/downloads/details.aspx?FamilyID=9df71545-4524-451b-8a17-9925b64010d2&displaylang=en

- Visual Studio Team System 2010 Test Load Agent Bêta 1 : qui ouvre une nouvelle catégorie d'outils dans le cycle de développement, dédiée aux testeurs fonctionnels.

http://www.microsoft.com/downloads/details.aspx?familyid=1967F42A-1E8B-4C70-9329-8478B68097D9&displaylang=en

On peut rajouter comme nouveauté dans VSTS 2010 les éléments suivants :

Architecture :

- ***Architecture Explorer:*** vous permettant de naviguer au sein de l'architecture technique de votre solution ou projet.
- ***Use case designer :*** vous permettant de créer votre UML au sein de

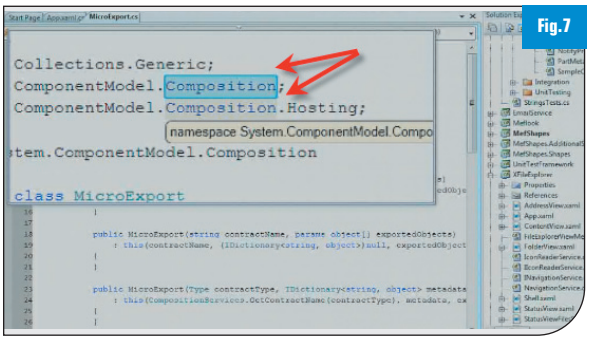


Fig.7

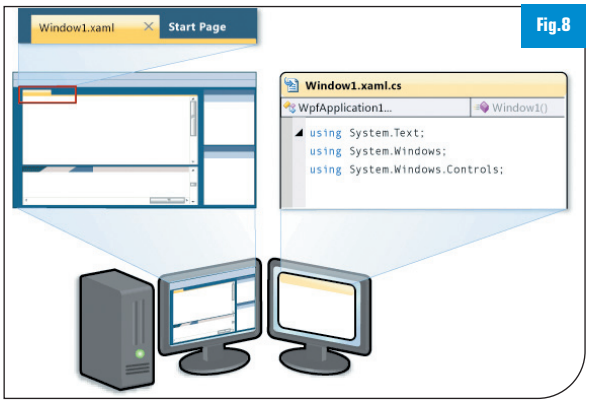


Fig.8

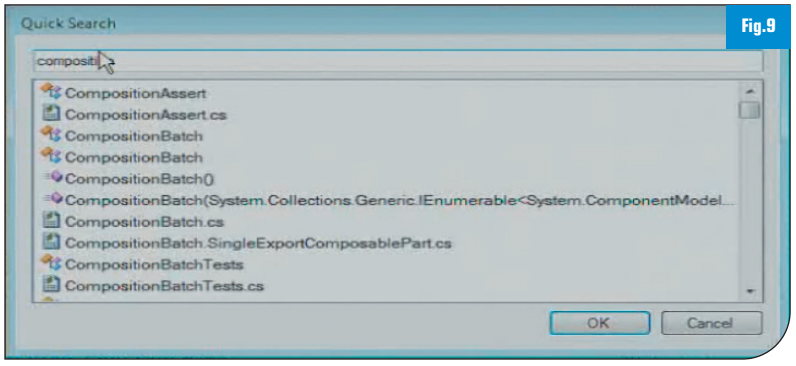


Fig.9

Visual Studio avec une synchronisation bidirectionnelle.

- ***Layer diagram:*** Cette nouveauté vous permet de définir un diagramme présentant les couches techniques de vos solutions, de définir vos règles de validation d'architecture afin de guider vos développeurs.

Mais il en existe encore bien d'autres : Sequence Diagram, Modeling Project System ; UML Model Explorer, Component diagram, … Et oui, UML fait son apparition au cœur de Team System, même si tous les diagrammes ne sont pas présents.

Développement et base de données

- ***Historical debugging :*** débugging par navigation au sein de la pile
- ***Test impact analysis :*** permettant de déterminer les tests unitaires impactés par les dernières modifications de code.
- ***Database extensibility :*** incorporation de nouveaux providers

Quelques liens :

- Download the Microsoft Visual Studio® 2010 Beta : http://www.microsoft.com/visualstudio/en-us/products/2010/default.mspx
- Visual Studio 2010 and .NET Framework 4 Beta 1 Walkthroughs : http://msdn.microsoft.com/fr-fr/vstudio/dd441784(en-us).aspx
- Synthèse nouveautés de VS 2010 : http://msdn.microsoft.com/en-us/library/bb386063(VS.100).aspx
- Nouveautés du .NET Fx 4.0 : http://msdn.microsoft.com/en-us/library/dd409230(VS.100).aspx
- Focus sur les nouveautés de VS2010 : http://msdn.microsoft.com/en-us/library/dd547188(VS.100).aspx
- Synthèse des nouveautés de VSTS : http://blogs.codes-sources.com/azra/archive/2009/04/09/visual-studio-team-system-2010-quoi-de-neuf.aspx

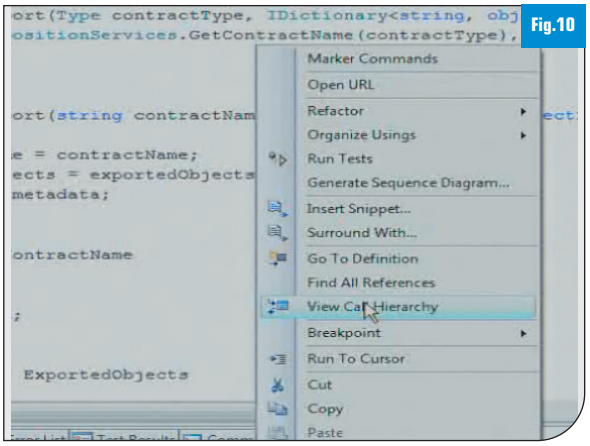


Fig.10

CONCLUSION

En conclusion, je ne pourrais que vous conseiller de regarder de très près cette nouvelle version de Visual Studio tant celle-ci incorpore de nombreuses nouveautés diverses et variées. Tout profil y retrouvera ses marques et avantages dans la pratique de son activité au quotidien. Y compris une toute nouvelle catégorie d'utilisateurs à ne jamais négliger : « Nos CLIENTS » avec des solutions comme CAMANO permettant la gestion automatisée de scripts et saisies de tests fonctionnels.

Mes coups de cœurs :

- Quick Search
- Hierarchy Call
- Architecture Explorer
- Nouveautés de VSTS 2010 en général.

Vous pourriez également me demander quels sont les points moins positifs de cette nouvelle version? Je dois avouer en toute franchise ne pas encore en avoir réellement découverts, tant la liste des nouveautés est longue et vient combler les principales lacunes de la version 2008 de Visual Studio.

Rendez-vous dans quelques mois pour une analyse plus critique après une gestion en production plus soutenue de cette version.

Quoi qu'il en soit, n'oubliez pas de profiter de la « software assurance » afin de réaliser une migration à moindre frais de 2008 à 2010.

Grégory **Renard**
CTO - R&I Manager Wygwam
Microsoft Regional Director - Microsoft MVP (Most Valuable Professional)

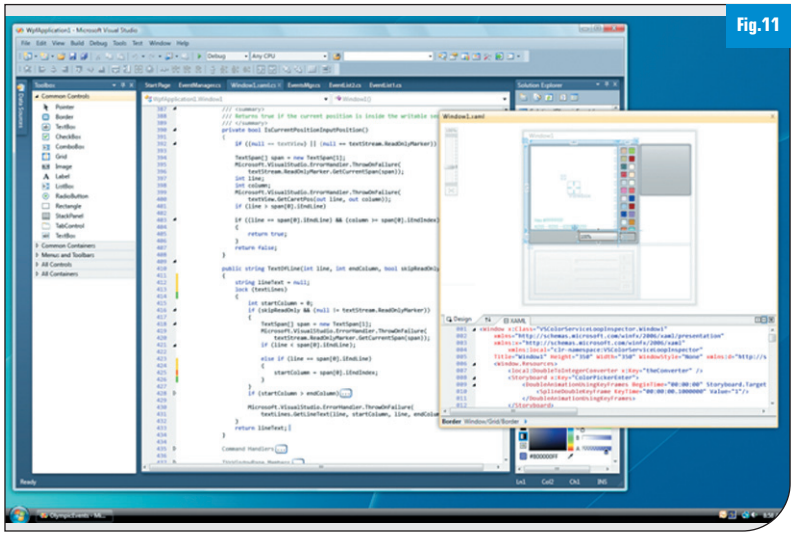


Fig.11

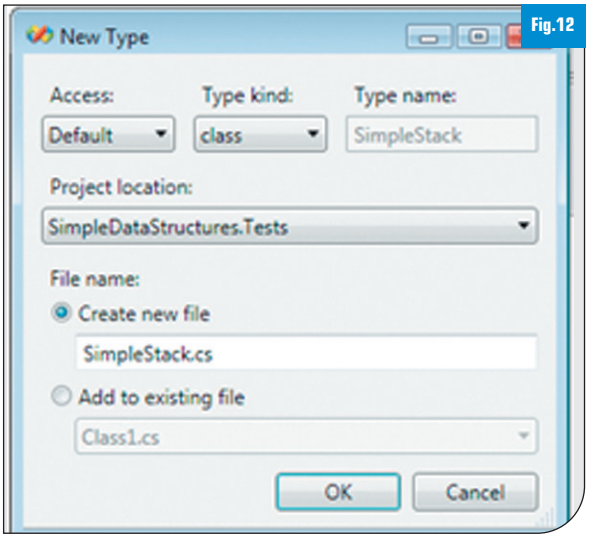


Fig.12

Tous les jours : l'actu et le téléchargement
www.programmez.com

WINDEV WEBDEV WINDEV Mobile

Développez 10 fois plus vite

Nouvelle version 14

PLATEFORME PROFESSIONNELLE DE DÉVELOPPEMENT (AGL)
Pocket PC, Smartphone, Terminal
Windows CE, Windows Mobile

WINDEV® Mobile

DÉVELOPPER VOS APPLICATIONS POUR POCKET PC, SMARTPHONE & TERMINAL MOBILE : FACILE !

WINDEV Mobile 14 est l'environnement de développement professionnel qui permet de développer jusqu'à **10 fois plus vite** les applications sur mobile dont votre entreprise et vos clients ont besoin: gestion de stock, force commerciale, géolocalisation, saisies médicales, expertises, relevés de terrain, prise de commande temps réel, réglage de chaîne de production, etc...

La **puissance** et la **facilité** de développement de WINDEV Mobile 14 permettent un développement complet en quelques journées.

L'environnement est livré complet, le déploiement des applications réalisées est **gratuit** sans redevances (base de données incluse).

Toutes les fonctionnalités d'un AGL professionnel sont offertes. Tous les aspects de la mobilité sont gérés: base de données, accès direct, réplication, WiFi, Bluetooth, 3G, Internet, socket, ActiveSync, réseau, J2EE, SMS, RFID, lien avec votre S.I., **codes-barres automatiques**...

Un tableau de bord en temps réel sur son téléphone? Facile !

VERSION EXPRESS GRATUITE
Téléchargez-la !

Vous aussi réalisez vos applications mobiles 10 fois plus vite... avec WINDEV Mobile 14.

*(Logiciel **professionnel,** Support Technique gratuit*)*

VOTRE CODE EST MULTI-PLATEFORMES:
Windows, .Net, Java, PHP, J2EE, XML, Internet, Ajax, Pocket PC, SmartPhone, Client riche ...

DEMANDEZ LE DOSSIER GRATUIT
252 pages + DVD + **Version Express incluse**
+ 112 Témoignages.
Tél: **04.67.032.032** ou **01.48.01.48.88**
info@pcsoft.fr

*Logiciel **professionnel***

Fournisseur Officiel de la Préparation Olympique

www.pcsoft.fr

PC SOFT 25 ANNÉE

Logiciel professionnel. Document non contractuel. **Support technique gratuit**: 15 requêtes sur la version en cours de commercialisation. - WINDEV, WEBDEV et WINDEV Mobile sont compatibles.

Maîtriser Dreamweaver : Site dynamique et MySQL

1re partie

Vous venez d'installer sur votre machine Dreamweaver CS4 et Wamp 2 ? Nous vous proposons aujourd'hui de connaître et de maîtriser les bases de Dreamweaver qui peuvent parfois être complexes.

Nous verrons comment dans cette série d'articles :

1. **Ajouter un site**
2. **Connecter votre base de données**
3. **Le rôle du jeu d'enregistrements**
4. **La manipulation basique des données, ajouter, modifier, supprimer.**
5. **Les objets de formulaires.**

1 DÉFINITION DU SITE EN MODE ASSISTÉ

Avant de définir le site, ajouter deux dossiers identiques, l'un côté développement sur votre disque sous **Mes Documents,** l'autre sous la racine **www de Wamp,** c'est ici que votre site pourra s'afficher dans votre serveur local. Dans le menu Site, sélectionnez Nouveau site puis onglet élémentaire. [Fig.1]

Sur la page suivante de l'assistant, il faut sélectionner « oui je veux utiliser une technologie de serveur », puis dans le menu déroulant, **PHP MySQL** (pour notre exemple). Dans la page suivante, on renseigne le dossier côté développement. Je vous conseille cette méthode, car cela évite de perdre des heures de travail, on gardera côté serveur local la dernière version du fichier, puis on renseigne la partie réseau. Si vous cochez la case : **Actualiser** …, le site sera mis à jour automatiquement dès que vos fichiers modifiés seront enregistrés dans votre dossier local. Cette option, je vous la conseille quand vous serez plus à l'aise avec **DW Cs4.** Si vous souhaitez mettre en ligne, il faudra choisir « **FTP** » et renseigner tous les paramètres de votre hébergeur. Lorsque tout est paramétré, le site apparaîtra dans le panneau fichier. [Fig.2]

Pourquoi ajouter un site ?

La gestion des fichiers entre le dossier qui sera sur votre disque dur et votre ressource distante, qui peut être votre hébergeur, ou bien un serveur local, sera gérée d'une façon automatique, et réduira un grand nombre d'erreurs. La méthode que je vais décrire ici est celle que j'utilise à chaque fois que je commence un développement, libre à vous de l'adapter à vos besoins. Avant d'aller plus loin, il est nécessaire d'installer Wamp 2.0 sur votre machine, vous aurez à votre disposition, un serveur Apache + Mysql + PHP, une boîte à outil complète pour bien commencer votre premier site dynamique. Vous trouverez sur le web, des didacticiels qui expliquent le fonctionnement du logiciel.

2 CONNECTEZ VOTRE BASE DE DONNÉES

Il nous faut maintenant créer la base de données de notre site web comme ci-dessus : une base **Test** contenant une table CLIENTS, pour continuer notre progression. Cette action est réalisée grâce à Wamp 2 qui intègre PhpMyAdmin [Fig.3]

Retour à DW CS4

Nous ajoutons tout d'abord une nouvelle page PHP : menu Fichier -> Nouveau -> sélectionnez Page Vierge PHP puis cliquez sur Créer. Le nom de la nouvelle page sera : index.php. Si tout est bon, dans le

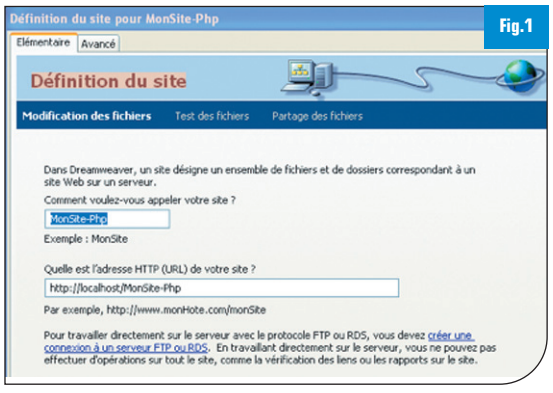


Fig.1

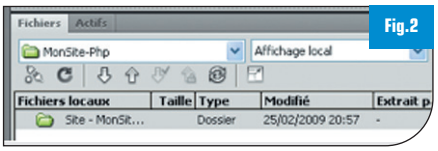


Fig.2

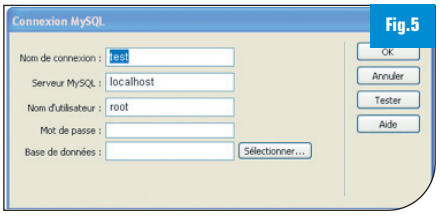


Fig.5

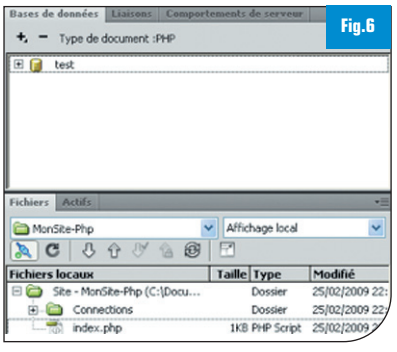
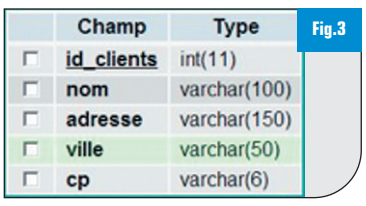


Fig.6



Champ	Type
id_clients	int(11)
nom	varchar(100)
adresse	varchar(150)
ville	varchar(50)
cp	varchar(6)

Fig.3

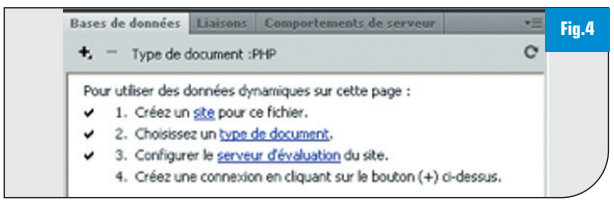


Fig.4

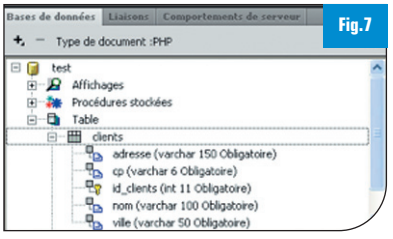


Fig.7

panneau **base de données,** cliquez sur Plus puis Connexion MySQL, l'assistant de création se lance.[Fig.4]
Plusieurs informations sont nécessaires pour créer et connecter une base de données :
- **Nom de la connexion** > donnez lui un nom, un fichier du même nom sera ajouté à votre site.
- **Serveur MySQL** > le vôtre souvent, **localhost (en local)**
- **Nom d'utilisateur : root** ou celui que vous vous avez mis pour la gestion de vos bases de données.
- **Mot de passe :** le renseigner si il y en a un, sinon laisser le champ vierge. Un message de la part de **DW Cs4,** vous avertira que ce n'est pas le meilleur choix. [Fig.5]
Pour sélectionner une base existante, il suffit de cliquer sur Sélectionner. Pour mémoire, nous avons pris Test comme base de données. On contrôle ensuite la présence de la base : [Fig.6]
Vous remarquerez qu'un dossier *Connections* est ajouté à votre site, il contient le fichier de connexion :

```
<?php
# FileName="Connection_php_mysql.htm"
# Type="MYSQL"
# HTTP="true"
$hostname_test = "localhost";
$database_test = "test";
$username_test = "root";
$password_test = "";
$test = mysql_pconnect($hostname_test, $username_test, $password
_test) or trigger_error(mysql_error(),E_USER_ERROR);
?>
```

3 LE JEU D'ENREGISTREMENTS

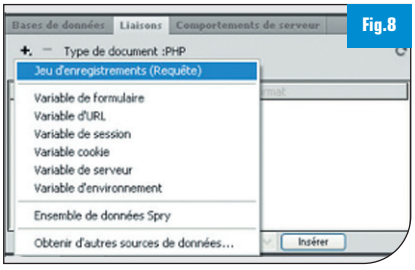
Nous allons voir en détail le jeu d'enregistrements, véritable trait d'union entre une base de données MySQL et votre page web. Pour comprendre toutes les subtilités du jeu d'enregistrements, nous garderons la même table, et si besoin en ajouterons d'autres.
A retenir : recordset = jeu d'enregistrements
Allez au panneau **Application > Onglet base de données,** et dérouler le menu sous le nom de votre connexion. [Fig.7]
On ajoute une page à notre site. La page : **index.php** étant ouverte, allez dans le panneau :

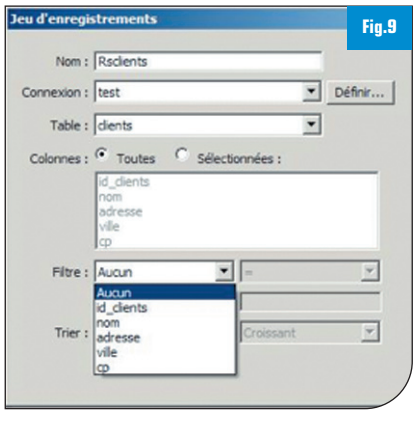
Application > Onglet Liaisons > Cliquez sur + [Fig.8]

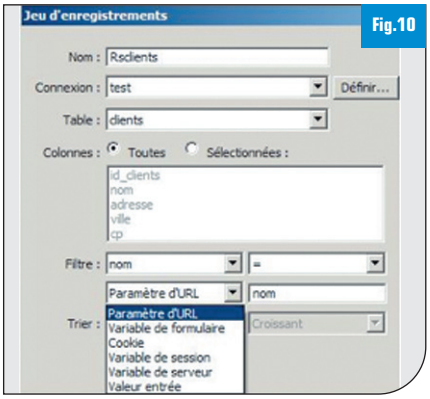
Choisir « Jeu d'enregistrements (Requête) », pour lancer l'assistant. Le principal intérêt du stockage des données dans une base c'est de pouvoir les manipuler, chose que nous verrons dans un prochain article. Gardez à l'esprit qu'à chaque fois que vous avez besoin de ces données il vous faut ajouter autant de jeu d'enregistrements, en fonction de ce que vous souhaitez obtenir comme résultat. Chaque jeu d'enregistrements correspond en fait à une commande **SELECT,** cette commande est très utilisée pour la sélection des données à manipuler dans une requête. [Fig.9]

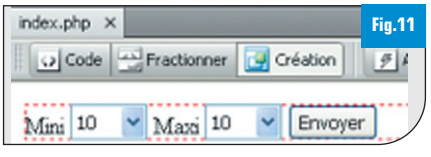
En mode simple nous avons les informations suivantes :
- **Nom** > le nom de votre recordset, donner un nom qui sera en relation avec la requête SQL ou votre table.
- **Connexion** > la connexion mise en place après la définition du site.
- **Table** > la table que vous souhaitez utiliser
- **Colonnes** > ici vous avez la possibilité de choisir tous les champs ou bien de sélectionner certains champs.
- **Filtre** > Liste les champs sur lesquels nous pouvons agir pour obtenir certaines données, il sera combiné avec le sélecteur : égal, plus grand que, plus petit que, etc.
Remarquez aussi que le sélecteur qui se situe juste dessous celui du « Filtre », est devenu actif.

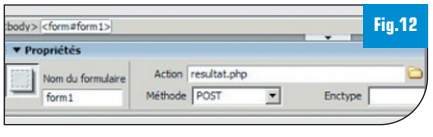
Vous avez à votre disposition tous les cas de figure. C'est avec ces sélecteurs et en les combinant que l'on ajoute à notre requête les paramètres de la clause WHERE. [Fig.10]

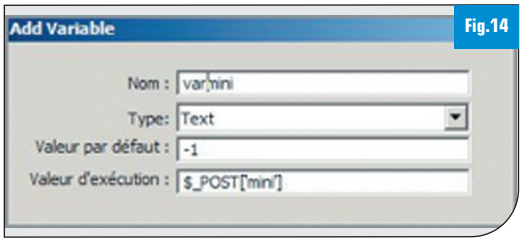


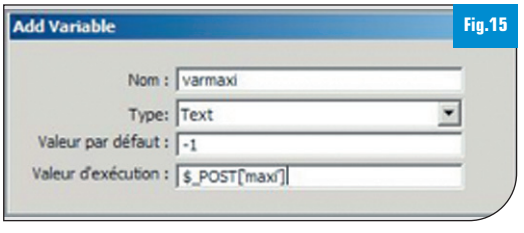












Les valeurs à filtrer	
Paramètre d'Url	Correspond à une valeur passée dans l'Url. La méthode utilisée : GET
Variable de Formulaire	Correspond à une valeur passée lors d'une validation du formulaire. Méthode : POST
Cookie	Valeur récupérée d'un cookie
Variable de session	Correspond à la valeur passée dans une session
Variable serveur	Correspond à la valeur soumise par le serveur
Valeur d'entrée	Correspond à la valeur du champ de la table

En mode avancé, nous avons plus d'options. Comme point de départ nous voulons afficher tous les enregistrements de la table compris entre deux tranches d'âge. Pour répondre à cette question, il nous faut modifier la **table clients** pour lui ajouter un champ « age » qui n'était pas présent dans le début de la présentation. Il nous faut deux pages :

selection.php > affichera deux liste /menu, le choix mini et le maxi. Voici la page dans Dreamweaver [Fig.11]

Il faut mettre les valeurs dans chaque liste / menu : il est très important de nommer le premier **« mini »** et le second **« maxi »**, car nous allons construire notre requête sur ces deux variables.

Dans la propriété de la balise **« Form »,** indiquez la page **« resultat.php »,** enregistrez votre page. [Fig.12]

La page resultat.php affichera les enregistrements compris entre les deux choix, donc nous avons besoin d'un jeu d'enregistrements capable de filtrer cette fois les paramètres du formulaire des sélecteurs mini et maxi.

Dans la page resultat.php, **Ajoutez un jeu d'enregistrements et cliquez sur le bouton « Avancé ».** Mettre un nom pour votre recordset et choisir la connexion. Vous avez remarqué que les champs SQL, les variables sont vides ; c'est normal.

Pour ajouter des champs dans la requête : Sélectionnez la table > le champ et cliquez sur le bouton **« select »,** et ainsi de suite pour tous les champs que vous voulez voir paraître à l'affichage. [Fig.13]

Maintenant, ajoutons nos deux variables **« mini » et « maxi » (bouton +).** On donne un nom à nos variables **varmini** et **varmaxi,** elles sont du type text, si c'était un lien comme tout à l'heure le type serait : Numeric, il peut être « Date » ou « Double » selon les besoins.

La valeur par défaut : mettre une valeur qui ne figure pas dans le choix, ici -1.

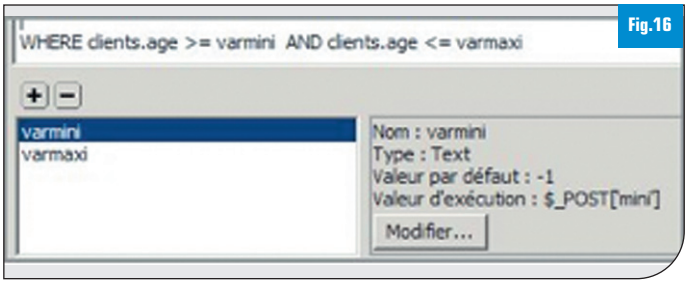


Fig.16

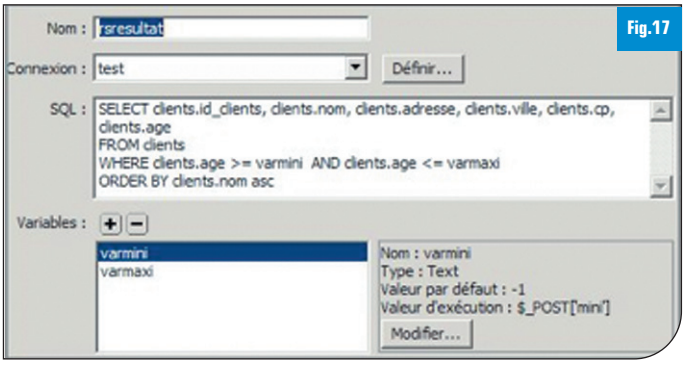


Fig.17

Valeur d'exécution : **$_POST['mini']** , ici on affecte la valeur qui provient du nom de la liste menu, on les avait nommées « mini » et « maxi », c'est à cet endroit que les erreurs sont faites lors des passages de paramètres. [Fig.14 et 15]

Il nous reste à compléter notre clause « Where ». Sélectionnez le champ **« age »** de la table clients et cliquez sur le bouton **Where** à la suite de **clients.age** saisir **>= varmini**

Refaire la même manipulation pour la valeur maxi. Sélectionnez le champ « age » de la table clients et cliquez sur le bouton **Where** à la suite de **clients.age** saisir **<= varmaxi** [Fig.16]

Sélectionnez le nom et cliquez sur le bouton **ORDER BY** pour le classement, ne pas oublier de mettre **ASC pour le tri croissant ou DESC pour l'inverse** à la suite de la commande.

Voilà le résultat: [Fig.17]

Le mois prochain nous aborderons les manipulations de données.

■ Jean Pierre **Valls**
Asp-php.net

Web
Adobe dévoile Flash Catalyst bêta

On attendait depuis des mois une version publique et stable de Flash Catalyst (ex-Thermo). Cet outil est le chaînon manquant dans les équipes de développeurs et de designers web, bref, entre ceux qui font l'interface et ceux qui font du code. Un peu comme Expression Blend de Microsoft, Catalyst permet de concevoir des interfaces dynamiques et riches sans code. Le code pouvant être rajouté et modifié par le reste de l'équipe. Surtout l'interface peut être déployée directement sur le web, l'outil générant du SWF, le format Flash. Catalyst est idéal pour maquetter rapidement un projet, tester une idée d'interface ou tout simplement pour mieux découper les projets web (Flash, Air, Flesh). Il s'intègre tout naturellement aux autres outils de création comme Photoshop et Illustrator. La collaboration est l'autre partie importante de l'outil et notamment vers Flex Builder. Parmi les principales fonctions :

- intégration des données
- puissante fonction timeline
- fonctions de Pages et d'Etats
- gestion des calques
- export SWF

La date de disponibilité n'est pas encore connue mais on parle de première moitié de 2010. Et surtout, toutes les fonctions sont désormais incluses et ne doivent plus bouger. Disponible sur MacOS X et Windows. Dans le même temps, Adobe sort une pré-version de Flash Builder, ex-Flex Builder. Il s'agit de la version 4.0 qui viendra avec Flex 4.0 et ActionScript 4.0.

egilia®

LEARNING

“Faire de vos **succès** notre **réussite**”

Formations certifiantes en informatique & management

www.egilia.com

CONTACTEZ NOS CONSEILLERS FORMATION

N°National 0 800 881 558

APPEL GRATUIT DEPUIS UN POSTE FIXE

Paris • Lyon • Lille • Aix-en-Provence • Strasbourg
Bordeaux • Toulouse • Rennes • Bruxelles • Genève

Le futur est là !

Windows 7, Java 7, Surface, Android, iPhone, robotique, Cerveau-Ordinateur, Etes-vous prêt ?

Nous vivons une rapide mutation technologique qui impacte ou impactera les utilisateurs, les développeurs. Dans ce dossier, nous allons bien entendu nous focaliser sur vous, les développeurs mais nous allons aussi vous dévoiler quelques recherches fondamentales sans doute un peu éloignées du code mais qui montrent de quoi demain sera fait.
Nous regarderons aussi l'avenir technologique au cœur des applications, des systèmes et des langages. Et vous verrez que le quotidien d'un développeur va changer assez rapidement. Bien sûr, l'informatique prédictible, le massivement parallèle, le web sémantique, ne sont pas pour demain, hormis peut être le web sémantique qui commence à arriver. C'est aussi l'informatique dite ubiquitaire. C'est-à-dire accessible partout et sur n'importe quel terminal, système. Cela passera aussi par une révolution dans le graphisme, l'affichage sur l'écran avec les premiers écrans 3D Relief. Une autre manière de percevoir l'interface, les objets applicatifs. A moyen terme, c'est la manière de piloter les applications, l'interface qui va changer, comme dans le projet Microsoft Natal, où le corps devient clavier, souris, manette de jeux. L'un des défis de l'informatique est ce que l'on appelle l'informatique prouvée (proof computing), ou comment connaître, prédire le comportement d'une application, d'un système dans x mois ou selon tel contexte.
Préparez-vous à faire votre révolution technologique durant les prochaines semaines !

■ François **Tonic**

Sony avait fait sensation, il y a 10 ans, en proposant à la vente le chien Aibo, même si sa production s'est arrêtée dès 2006. Le marché a considérablement évolué depuis. Mindstorm, le lego programmable a su dévoiler à un public très large, les joies des automates. Aujourd'hui, chaque pays renforce sa recherche. Et dans cette bataille, la France possède de nombreux atouts et des constructeurs et éditeurs reconnus mondialement.

Robotique : le prochain défi !

On critique souvent la recherche française, le manque d'innovation et pourtant, dans la robotique mondiale, notre pays représente presque 10 % des publications. *« La France est réellement bien placée »* précise M. Baillie (fondateur de Gostai). Plusieurs noms connaissent une bonne notoriété auprès des chercheurs, étudiants : Aldebaran, Gostai, Violet avec son fameux Nabaztag. Il existe aussi le constructeur Pob Technology. Récemment, une initiative nationale a été mise en place pour fédérer les efforts autour de la robotique : Cap Robotique. L'objectif est simple : faire émerger le secteur robotique et les objets communicants en soutenant, en développant le business et l'innovation. Ce sont plusieurs dizaines d'acteurs nationaux (constructeurs, écoles, laboratoires, éditeurs) qui adhèrent à cette initiative. Plusieurs projets sont déjà initiés tels que Romeo, un robot humanoïde ayant pour rôle de devenir un assistant aux personnes en perte d'autonomie. L'objectif est de produire un prototype fonctionnel d'ici fin 2011. Le budget initial est de 10 millions d'euros.

© Aldebaran

Il y a robot et robot

Il existe deux grands types de robots sur le marché. Le robot dit programmable comme en propose le magasin Génération Robot (www.generationrobots.com). Nous ne sommes pas dans des « robots jouets » mais dans de vrais robots à monter, en kit, ayant des capacités fonctionnelles étendues comme la caméra, les roues, les cerveaux moteurs, etc. Ces robots sont dits éducationnels. Pour Jérôme Laplace de Génération Robots, il y a deux profils d'acheteurs : les geeks et passionnés et les enseignants / étudiants qui aiment l'aspect programmation, personnalisation. Cependant, le frein majeur reste le prix. Lego Mindstorm coûte plus de 300 euros, et un kit robotique Pob Technology dépasse 350 euros.

L'autre catégorie est plutôt le robot « grand public » autonome, capable d'interagir avec son environnement, selon les situations, avec des capacités de communication étendues. Le Nao d'Aldebaran en est le représentant le plus connu même si aujourd'hui seule la version académique est disponible. La version « tout public » sera dévoilée dans quelques mois à un prix que l'on promet attractif.

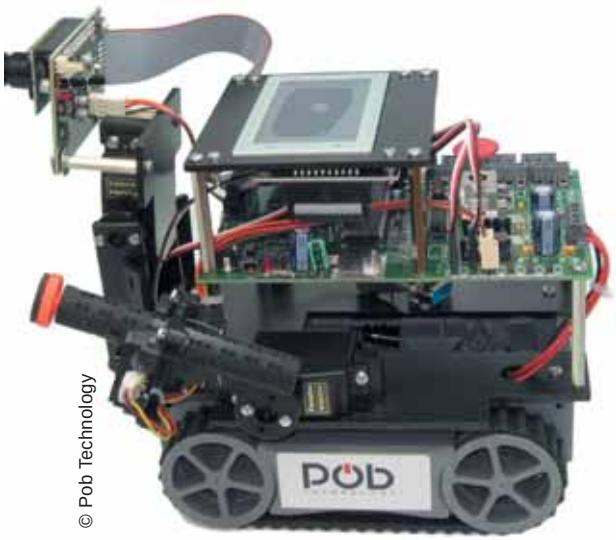
Un modèle de développement à maîtriser

Basiquement, il existe deux niveaux de programmation : le bas niveau et le haut niveau. Dans le bas niveau, nous trouvons le code embarqué dans les cerveaux, les contrôleurs, le processeur, etc. Là, le code est écrit en C++. Le haut niveau sera plus orienté fonctionnel, interaction.

© Robocup 2007

Mais programmer un robot devient rapidement complexe car ce type de paradigme exige un code massivement parallèle, un robot étant capable d'exécuter plusieurs tâches simultanément. Et surtout il possède des processus s'exécutant en arrière-fond qui vérifient l'état de tel élément, de telle fonction pour exécuter une nouvelle tâche et interagir avec l'environnement ou une situation donnée. L'un des problèmes, que l'on retrouve dans le développement traditionnel, est l'hétérogénéité des plates-formes. Car chaque constructeur de robot implémente ses propres API, librairies.

C'est pour répondre à ce problème de fragmentation du marché que le français Gostai a lancé son environnement (système + langage de script + API + outils) : Urbi. Urbi désigne le système global capable de s'exécuter sur une quinzaine de robots du marché. Urbi est parti de la recherche. *« J'étais chercheur et il me fallait un outil de programmation pour Aibo qui soit à la fois puissant et simple d'utilisation pour les étudiants »,* précise J-C Baillie (fondateur de Gostai). A partir de là, il y a trois ans, la société fut fondée et le concept Urbi s'est structuré. Il propose à la fois des outils graphiques pour créer rapidement, en quelques clics, des applications robotiques mais le développeur, le chercheur accèdent au code pur qu'il soit C++ ou UrbiScript. L'originalité d'Urbi est de pouvoir adresser le code de bas et de haut niveau.



© Pob Technology

UrbiScript, le langage de haut niveau, s'interface avec les codes C++, les API constructeurs, pour pouvoir interagir, manipuler les éléments du robot. Pour l'éditeur, UrbiScript doit être vu comme une sorte de coordinateur des morceaux de code C++. Mais comme un robot est avant tout une programmation massivement parallèle, Urbi embarque en son cœur un scheduler pour assurer la bonne exécution des tâches. Gostai développe aussi ses propres pilotes matériels pour assurer la compatibilité avec les robots.

Dès que l'on conçoit des logiciels aux mouvements et interactions complexes, la programmation robotique devient rapidement critique à maintenir et surtout à deboguer. La structure des applications n'étant pas comme dans un projet desktop ou web, pour faciliter la segmentation du code et mieux reconnaître chaque section, Urbi introduit le tag. Il permet de nommer un code et de faciliter ainsi sa manipulation dans un mode debug. Il est possible de stopper, le freezer, le unfreezer. L'objectif de Urbi est d'être le plus indépendant possible. J-C Baillie compare un peu Urbi à un Java de la robotique car une application Urbi fonctionne, sans trop de modifications, sur l'ensemble des robots compatibles.

Avec Urbi 2, l'éditeur veut aller plus loin avec une double licence. L'objectif, de l'aveu même de son fondateur, est d'étendre l'usage d'Urbi auprès des étudiants, universités. L'ambition est toujours de simplifier le développement. Et dans quelques mois, Gostai va encore innover en proposant l'accès depuis le robot à des fonctions en ligne (en mode Saas). Car sur des robots grand public, dotés de processeurs de puissance limitée, certaines fonctions ne peuvent être exécutées localement. Gostai a donc mis en place une infrastructure en ligne pour exécuter ces fonctions directement sur le web, par exemple la reconnaissance faciale. Mais on pourra aussi surveiller, voir ce qui se passe à la maison, ou dans tout autre endroit, depuis un navigateur ou son téléphone 3G…

Aldebaran : projet Nao

Autre français en pointe, la société Aldebaran avec son robot Nao. Créée par un passionné de robotique, Bruno Maisonnier, Aldebaran existe depuis 2005 et emploie aujourd'hui plus de 50 personnes. Aujourd'hui, Nao est une réalité qui s'exporte auprès des chercheurs, étudiants, laboratoires, universités. Il possède la synthèse et reconnaissance vocale, les émotions, un ordinateur embarqué avec wifi, deux caméras, un système embarqué Linux, des capteurs, etc. Il possède par défaut les comportements de base, toute liberté est donnée aux développeurs grâce aux outils livrés avec, notamment un superbe outil graphique : Choregraphe pour créer rapidement les comportements, les scénarios d'action. La version grand public est espérée pour fin 2009 ou 2010. Les plans ne sont pas encore totalement arrêtés.

Nao a eu une reconnaissance internationale grâce à la RoboCup 2008. Le robot ayant été le robot officiel de la compétition ! Il s'agit d'une compétition de football robotique. Et l'aventure continue en ligne grâce à un environnement virtuel particulièrement réaliste : robotstadium incluant un avatar virtuel de Nao (http://www.robotstadium.org/).

Demain et après demain

Au-delà de la robotique, c'est le concept des objets communicants qui se développe rapidement. Le plus connu est le français Violet avec son fameux lapin : Nabaztag. Il est capable de lire, d'entendre et d'interagir (de manière limitée) avec un autre lapin. Les communications entre objets, ce que l'on appelle le Machine to Machine (ou M2M) est en pleine explosion. La domotique va enfin pouvoir exister grâce à cette technologie. Et demain, ce sera au tour des robots de pouvoir interagir, communiquer entre eux ou avoir un objet communicant. Cependant nous a confié M. Baillie, ce n'est pas pour demain mais plutôt après demain, il faut d'abord que la robotique grand public se banalise et elle sera ensuite interfacée à tous les éléments de la maison et de la vie quotidienne.

■ François **Tonic**

WINDEV : VOUS AUSSI, DÉVELOPPEZ 10 FOIS PLUS VITE !

WINDEV : élu «Langage le plus productif du marché»

www.pcsoft.fr

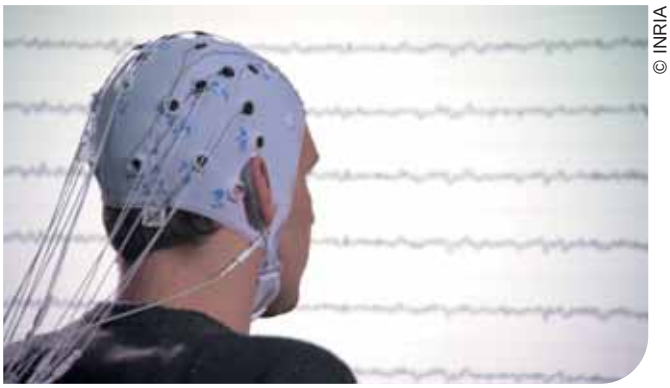
252 pages + DVD + **Version Express incluse**
+ 112 Témoignages.

Tél: **04.67.032.032** ou **01.48.01.48.88**

info@pcsoft.fr

Les interfaces cerveau-ordinateurs : le futur de l'informatique

Les plus cinéphiles se souviennent sans doute du film Firefox , dans lequel un pilote manœuvrait un avion militaire par la pensée. Depuis de nombreuses années, à travers le monde, des recherches sont menées sur ces technologies. Aujourd'hui, la France s'est positionnée sur cette informatique fondamentale qui initie l'usage informatique dans le futur. Etes-vous prêt pour OpenVibe ?



© INRIA

Avec OpenVibe, la France rattrape une partie de son retard technologique. Il s'agit du premier projet public de ce genre. Aujourd'hui, plusieurs autres laboratoires travaillent sur les interfaces cerveau-ordinateurs.

OpenVibe, auquel nous avions consacré un article dans le précédent numéro de Programmez ! sert d'interface entre le cerveau et l'application à piloter. Grâce aux outils graphiques d'OpenVibe, il est possible de définir ce que les chercheurs nomment des interfaces cerveau – ordinateur ou Brain Computer interface en anglais (ou BCI). Le BCI correspond à des « ordres directs » envoyés par le cerveau et que le système reconnaît et exécute. Ce sont des « commandes mentales ». Dans ce projet, il est possible de définir un scénario d'utilisation BCI, c'est-à-dire un enchaînement d'actions et de commandes mentales pour manipuler telle ou telle application. Car ces commandes ne seront pas les mêmes dans un traitement de texte ou dans un jeu en réalité virtuelle. Lors de la présentation officielle du projet, début mai, plusieurs applications avaient été présentées, comme le P300, un traitement de texte.

L'activité cérébrale comme commande

Dire que la pensée contrôle l'ordinateur est un peu exagéré. En réalité, c'est l'activité cérébrale qui le contrôle. Le but est de capturer celle-ci, puis la plate-forme OpenVibe essaie de la comprendre. Comme le précise Yann Renard (Ingénieur pilotant les développements de la plate-forme OpenViBE, INRIA Rennes Bretagne Atlantique), on ne sait pas ce que pense l'individu, pas encore...

Concrètement, si on prend un musée virtuel dans lequel le visiteur doit déambuler, comment OpenVibe comprendra le cerveau ? Un casque doté d'électrodes, posé sur la tête, capte et enregistre l'activité cérébrale, en fait des impulsions électriques. L'utilisateur doit avant toute chose passer par un apprentissage. L'application compatible OpenVibe va par exemple dire à la personne d'imaginer un mouvement de la main gauche pour aller à gauche, aller à droite, marcher, etc. A chaque « commande », l'utilisateur va penser, ce qui va provoquer une activité cérébrale spécifique ou tout du moins particulière que OpenVibe va capturer et traduire par de puissants algorithmes. Ainsi la commande « aller à gauche » aura une signature cérébrale particulière que OpenVibe reconnaîtra et le logiciel ira à gauche. Cette phase d'apprentissage est donc particulièrement importante car il faut que la plate-forme puisse déterminer avec la meilleure qualité possible chaque signature cérébrale et l'utilisateur doit aussi apprendre à être concentré.

L'architecture d'OpenVibe

La plate-forme s'articule sur une partie logicielle et une partie matérielle. L'architecture d'OpenVibe est assez simple,

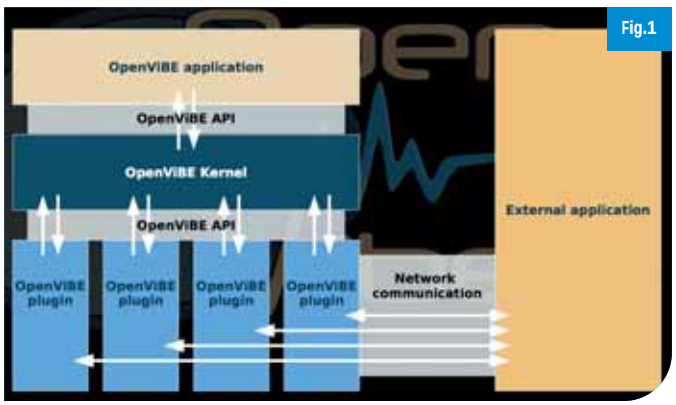


Fig.1

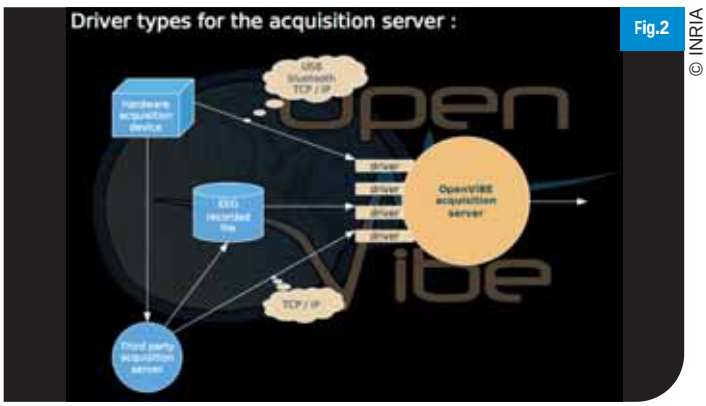


Fig.2

© INRIA

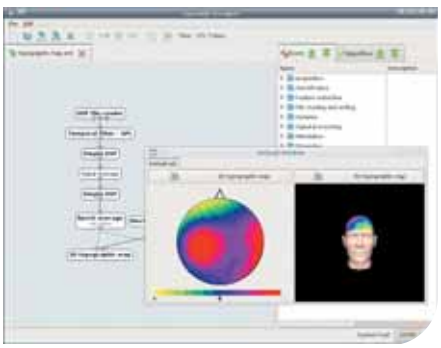
comme le montre le schéma [Fig.1]. Le noyau fait le lien entre les applications et le matériel. Grâce à une approche modulaire, on peut rajouter des plug-in, des drivers. Les applications compatibles OpenVibe reposent sur les API qui permettent de communiquer avec la plate-forme. Le noyau (kernel) expose les éléments suivants :

- gestionnaire algorithme
- gestionnaire des logs
- gestionnaire player
- gestionnaire de plug-in
- gestionnaire de visualisation
- gestionnaire de type
- gestionnaire de scénario

Côté logiciel, il faut disposer d'un serveur d'acquisition (servant à collecter les données provenant du matériel) et du designer. Ce designer est très important car c'est lui qui permet de définir, de tester et d'exécuter les scénarios d'utilisation. Bien que prévu à la feuille de route, il n'existe pour le moment pas de version embarquée d'OpenVIBE permettant de jouer des scénarios. La communication entre les outils se fait par TCP / IP.

Comme nous l'avons vu plus haut, l'acquisition des données se fait par un serveur d'acquisition. Les données proviennent d'un périphérique vu par des pilotes (drivers). Et le serveur d'acquisition s'appuie sur le driver pour récupérer les données. Le pilote a pour but de : récupérer les signaux et les événements et de les rendre disponibles au serveur. Il est à noter que l'interface du périphérique peut être de l'USB, du port parallèle ou série, etc. Et que la communication avec le serveur d'acquisition peut se faire via TCP / IP, un fichier, ou encore par COM ou équivalent. Le plug-in OpenVibe peut prendre deux formes : algorithmes, boîtes (ou boîtes d'algorithmes). D'autre part le plug-in implémente deux classes : le descripteur (IPluginObjectDecs) et le plugin lui même (IPluginObject). Il faut aussi intégrer dans un contexte OpenVibe les états du pilote. Car le driver passe par différentes phases : attente, configuration matérielle, initialisation pour démarrer l'acquisition des données, démarrage de l'acquisition, arrêt de cette tâche.

Pour les concepteurs et développeurs, il est à prendre en considération des éléments importants : les performances, la portabilité du code (en cas de multi-plate-forme), et de maintenance de celui-ci.


© INRIA

© INRIA

© INRIA

Etendre et compléter OpenVibe

Le projet a été développé en C++. Il est disponible pour Windows et Linux. Pour Windows, on dispose d'un installeur classique. Sur Linux, on dispose du code source qu'il faut ensuite compiler et packager. la communauté est d'ailleurs invitée à contribuer pour packager et maintenir des paquets sur différentes distributions Linux. Mais grâce à la mise à disposition du code et de la plate-forme en open source, les responsables du projet, et en premier lieu Yann Renard, espèrent voir fleurir de multiples projets, des pilotes, de nouvelles idées d'utilisateurs.

■ François **Tonic**

Pour quels marchés et utilisations ?

Dans un premier temps, et depuis le début du projet, OpenVibe vise les personnes handicapées pour les aider à communiquer, à accomplir des tâches. C'est notamment le cas avec P300 Speller qui est une interface pour écrire. Le système permet de sélectionner des lettres en les flashant. L'utilisateur porte son attention sur la lettre souhaitée, cela ce traduit par une onde que le système capte et interprète. Un autre champ d'utilisation est la rééducation de l'activité cérébrale par ce que les experts appellent le neurofeedback. Il permet de pallier des manques neurologiques comme certains troubles (auditifs par exemple). Il s'agit ici d'aider la personne à autoréguler son activité cérébrale. Le système envoie une représentation sensorielle de cette activité. A partir de là, la personne peut apprendre à compenser, à traiter ces manques.

Au-delà, cette technologie peut être intéressante pour les visites virtuelles, les jeux. Le marché est immense. Mais il faudra encore du travail pour stabiliser la plate-forme et surtout bénéficier d'un matériel adapté à un prix accessible.

Préparez-vous à Windows 7 dès aujourd'hui !

Avec la disponibilité de Windows 7 en version RC (Release Candidate), Microsoft a dévoilé une nouvelle bibliothèque contenant un ensemble d'API permettant de travailler, depuis du code managé, avec les nouvelles fonctionnalités de Windows 7: Windows API Code Pack for Microsoft .NET Framework (v0.85), http://code.msdn.microsoft.com/WindowsAPICodePack.

Les fonctionnalités incluses dans cette version sont les suivantes:
- support pour les objets Windows Shell namespaces, incluant les nouvelles librairies Windows 7
- Task Dialogs Windows Vista et 7
- Contrôle de Windows Explorer Browser supportant WPF et Windows Forms
- support Direct3D 11, DXG 1 et 1.1
- services de langues étendues (API)
- etc.

Il est à noter que cette bibliothèque n'est pas figée, elle sera mise à jour/finalisée peu de temps après la sortie de la version RTM (Release To Manufacture) de Windows 7. Nous allons donc voir, dans cet article, un ensemble des fonctionnalités offertes par cette bibliothèque.

Utiliser l'explorateur de fichier dans vos applications

La bibliothèque mise à disposition par Microsoft permet aux développeurs d'utiliser un contrôle nommé **ExplorerBrowser**. Celui-ci permet d'insérer l'explorateur de fichier directement au sein de vos applications (WindowsForms ou WPF): [Fig.1]

Pour l'utiliser, c'est extrêmement simple car toutes les fonctionnalités de l'explorateur sont exposées sous forme de **DependencyProperties** (il est donc possible d'utiliser le DataBinding !). De plus, un « wrapper » existe pour l'utiliser dans une application WindowsForms ! Pour le manipuler, c'est relativement simple :

```
<!-- The Explorer Browser -->
<WindowsAPICodePackPresentation:ExplorerBrowser
    x:Name="eb"
    Margin="5"
    AlignLeft="{Binding IsSelected, ElementName=lbiAlignLeft, Mode=TwoWay}"
    AutoArrange="{Binding IsSelected, ElementName=lbiAutoArrange, Mode=TwoWay}"
    CheckSelect="{Binding IsSelected, ElementName=lbiCheckSelect, Mode=TwoWay}"
    ExtendedTiles="{Binding IsSelected, ElementName=lbiExtendedTiles, Mode=TwoWay}"
    FullRowSelect="{Binding IsSelected, ElementName=lbiFullRowSelect, Mode=TwoWay}"
    HideFileNames="{Binding IsSelected, ElementName=lbiHideFileNames, Mode=TwoWay}"
    NoBrowserViewState="{Binding IsSelected, ElementName=lbiNoBrowserViewState,
 Mode=TwoWay}"
    NoColumnHeader="{Binding IsSelected, ElementName=lbiNoColumnHeader, Mode=TwoWay}"
    NoHeaderInAllViews="{Binding IsSelected, ElementName=lbiNoHeaderInAllViews,
 Mode=TwoWay}"
    NoIcons="{Binding IsSelected, ElementName=lbiNoIcons, Mode=TwoWay}"
    NoSubFolders="{Binding IsSelected, ElementName=lbiNoSubFolders, Mode=TwoWay}"
    SingleClickActivate="{Binding IsSelected, ElementName=lbiSingleClickActivate,
 Mode=TwoWay}"
    SingleSelection="{Binding IsSelected, ElementName=lbiSingleSelection, Mode=TwoWay}"
    AlwaysNavigate="{Binding IsSelected, ElementName=lbiAlwaysNavigate, Mode=TwoWay}"
    NavigateOnce="{Binding IsSelected, ElementName=lbiNavigateOnce, Mode=TwoWay}"
    ViewMode="{Binding SelectedValue, ElementName=ViewModeCombo, Mode=TwoWay}"
    ThumbnailSize="{Binding Value, ElementName=thumbnailSizeSlider, Mode=TwoWay}"
    AdvancedQueryPane="{Binding IsChecked, ElementName=lbiAdvancedQueryPane, Mode
=TwoWay, Converter={StaticResource tristateConverter} }"
    CommandsPane="{Binding IsChecked, ElementName=lbiCommandsPane, Mode=TwoWay,
 Converter={StaticResource tristateConverter} }"
    CommandsOrganizePane="{Binding IsChecked, ElementName=lbiCommandsOrganizePane, Mode
=TwoWay, Converter={StaticResource tristateConverter} }"
    CommandsViewPane="{Binding IsChecked, ElementName=lbiCommandsViewPane, Mode=TwoWay,
 Converter={StaticResource tristateConverter} }"
    DetailsPane="{Binding IsChecked, ElementName=lbiDetailsPane, Mode=TwoWay, Converter
={StaticResource tristateConverter} }"
    NavigationPane="{Binding IsChecked, ElementName=lbiNavigationPane, Mode=TwoWay,
 Converter={StaticResource tristateConverter} }"
    PreviewPane="{Binding IsChecked, ElementName=lbiPreviewPane, Mode=TwoWay, Converter
={StaticResource tristateConverter} }"
    QueryPane="{Binding IsChecked, ElementName=lbiQueryPane, Mode=TwoWay, Converter
={StaticResource tristateConverter} }"
    NavigationTarget="{Binding SelectedItem, ElementName=KnownFoldersList, Mode=OneWay}" />
```

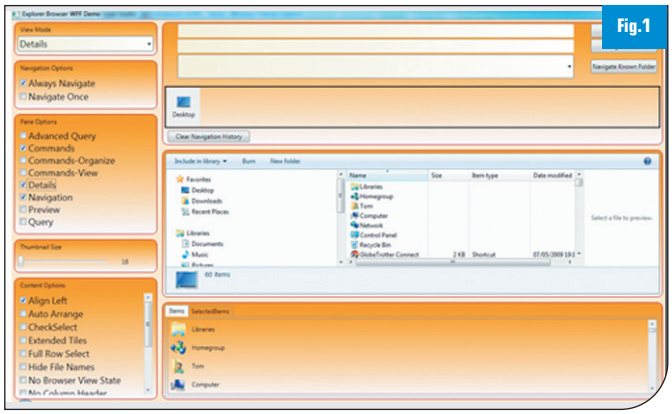


Fig.1

Comme vous pouvez le constater, un grand nombre de propriétés sont disponibles, permettant une manipulation fine et précise de ce contrôle.

Travailler avec les KnownFolders, les ShellObjects et les ShellLibraries

Il devient également très aisé de travailler avec les fichiers/répertoires de Windows 7. En effet, il est possible d'utiliser une classe (statique) nommée KnownFolders qui définit un ensemble de propriétés désignant les différents répertoires connus sur le système d'exploitation : [Fig.2]

Chaque répertoire, de type **IKnownFolder,** contient un ensemble d'objets de type **ShellObject** définissant un grand nombre de propriétés : [Fig.3]

Ainsi, il est possible de tout savoir sur un objet spécifique : [Fig.4]

Si l'objet en question est de type **IKnownFolder,** alors il est possible d'afficher les propriétés de ce répertoire : [Fig.5]

Dans le cas d'un objet de type *ShellLibrary,* on va travailler avec une **Library,** nouveauté de Windows 7 qui permet de regrouper, dans un même dossier, des fichiers provenant de répertoires différents. Dans ce cas, on va disposer de propriétés différentes à afficher/manipuler : [Fig.6]

Enfin, chaque **ShellObject** possède une propriété nommée **Thumbnail** qui permet de récupérer la vignette, sous différents formats/tailles, associée à l'objet en question : [Fig.7]

Voici un exemple lorsque l'on sélectionne la corbeille, qui est un KnownFolder : [Fig.8]

On constate donc qu'il est très aisé de travailler avec les KnownFolders, ShellObjets et ShellLibraries de Windows 7, grâce à cette bibliothèque d'API qui facilite grandement les développements !

Utiliser les icônes système standard

Vous avez également la possibilité de récupérer et d'utiliser la liste des icônes système standard disponible sur Windows 7.

Pour cela, vous allez devoir utiliser la classe **StockIcons**

```
StockIcons icons = new StockIcons();
```

Une fois cette classe instanciée, vous pouvez utiliser plusieurs propriétés qui servent à représenter les différentes icônes systèmes : [Fig.9]

Ensuite, chaque icône dispose de plusieurs propriétés que vous pouvez manipuler:

- *CurrentSize,* qui représente la taille courante de l'icône (vous avez le choix entre Large, ShellSize et Small)
- *LinkOverlay,* qui permet d'indiquer si vous souhaitez faire afficher la flèche servant à représenter l'icône comme raccourci
- *Selected,* qui affiche l'icône comme étant sélectionnée.

Voici un exemple résultant de l'affichage de toutes les icônes "système" : [Fig.10] Là encore, cela n'a rien d'extraordinaire mais permet d'avoir accès, rapidement et simplement, à l'icône correcte qui vous intéresse !

Manipuler la TaskBar

Windows 7 dispose d'une nouvelle barre des tâches (Taskbar) beaucoup plus évoluée que celle disponible dans Windows Vista:

- Possibilité de rajouter des entrées dans le menu contextuel
- Possibilité de customiser les miniatures qui apparaissent au survol de la souris.

Il pourrait être intéressant de savoir comment manipuler cette barre des tâches (et les fonctionnalités qu'elle offre) au travers d'une application .NET.

Lorsque l'on regarde la documentation associée à la barre des tâches de Windows 7, on se rend vite compte que toutes les fonctionnalités sont offertes grâce à des interfaces natives:

- **IApplicationDestinations**
- **IApplicationDocumentLists**
- **ICustomDestinationList**
- **ITaskbarList3**
- Etc.

Ainsi, avec un peu d'interopérabilité, on devrait être en mesure de faire un wrapper managé de ces interfaces. Il existe, pour le moment, 2 manières de manipuler la TaskBar de Windows Seven : **le Windows API Code Pack** ou bien la *"Windows 7 Taskbar and Libraries .NET Interop Sample Library"* (http://code.msdn.microsoft.com/Windows7Taskbar), sorte de pré-version du Code Pack et qui possède un peu plus de fonctionnalités.

Fig.2

```
/// <summary>
/// Gets the computer folder.
/// </summary>
/// <value>A <see cref="IKnownFolder"/> object.</value>
public static IKnownFolder Computer
{
    get
    {
        return GetKnownFolder(
            FolderIdentifiers.Computer);
    }
}

/// <summary>
/// Gets the Windows Vista Synchronization Manager folder.
/// </summary>
/// <value>A <see cref="IKnownFolder"/> object.</value>
public static IKnownFolder Conflict
{
        get
        {
            return GetKnownFolder(
                FolderIdentifiers.Conflict);
        }
}
```

Fig.3

```
public ShellThumbnail Thumbnail
{
    get
    {
        if (thumbnail == null)
            thumbnail = new ShellThumbnail(this);

        return thumbnail;
    }
}

private ShellObject parentShellObject = null;
/// <summary>
/// Gets the parent ShellObject
/// </summary>
public ShellObject Parent
{
    get
    {
        if (parentShellObject == null && nativeShellItem != null)
        {
            IShellItem parentShellItem = null;

            HRESULT hr = nativeShellItem.GetParent(out
parentShellItem);
```

Fig.4

```
internal static void ShowObjectData(Window1 wnd, ShellObject
shellObj)
{
    wnd.PropertiesTextBox.Text =
        String.Format(
            "Name = {0}{1}Path/ParsingName = {2}{1}Type = {3}{4}
({5}File System)",
            shellObj.Name,
            Environment.NewLine,
            shellObj.ParsingName,
            shellObj.GetType().Name,
            shellObj.IsLink ? " (Shortcut)" : "",
            shellObj.IsFileSystemObject ? "" : "Non ");
}
```

Fig.5

```
internal static void ShowKnownFolderProperties(Window1 wnd,
IKnownFolder kf)
{
    wnd.FolderPropsListBox.Visibility = Visibility.Visible;
    wnd.PropertiesGrid.RowDefinitions[1].Height = new GridLength
(150);

    Collection<KnownFolderProperty> properties = new
Collection<KnownFolderProperty>();

    properties.Add(new KnownFolderProperty { Property = "Canonical
Name", Value = kf.CanonicalName });
    properties.Add(new KnownFolderProperty { Property =
"Category", Value = kf.Category });
    properties.Add(new KnownFolderProperty { Property =
"Definition Options", Value = kf.DefinitionOptions });
    properties.Add(new KnownFolderProperty { Property =
"Description", Value = kf.Description });
    properties.Add(new KnownFolderProperty { Property = "File
Attributes", Value = kf.FileAttributes });
    properties.Add(new KnownFolderProperty { Property = "Folder
Id", Value = kf.FolderId });

    wnd.FolderPropsListBox.ItemsSource = properties;
}
```

Fig.6

```
internal static void ShowLibraryProperties(Window1 wnd,
ShellLibrary lib)
{
    wnd.FolderPropsListBox.Visibility = Visibility.Visible;
    wnd.PropertiesGrid.RowDefinitions[1].Height = new GridLength
(150);

    Collection<KnownFolderProperty> properties = new
Collection<KnownFolderProperty>();

    properties.Add(new KnownFolderProperty { Property = "Name",
Value = lib.Name});
    object value = null;

    properties.Add(new KnownFolderProperty { Property = "Library
Type Id", Value = value });

    properties.Add(new KnownFolderProperty { Property = "Path",
Value = lib.ParsingName});
    properties.Add(new KnownFolderProperty { Property = "Is Pinned
To NavigationPane", Value = lib.IsPinnedToNavigationPane });

    wnd.FolderPropsListBox.ItemsSource = properties;
}
```

Fig.7

```
internal static void ShowThumbnail(Window1 wnd, ShellObject
shellObj)
{
    try
    {
        wnd.ThumbnailPreview.Source =
shellObj.Thumbnail.LargeBitmapSource;
    }
    catch
    {
        wnd.ThumbnailPreview.Source = null;
    }
}
```

a) Afficher une barre de progression :

Vous pouvez faire afficher, dans l'icône de la barre des tâches de votre application, une barre de progression. Cela peut en effet s'avérer utile si un long traitement est en cours mais que l'application est réduite. Pour cela, il suffit d'utiliser le code suivant: [Fig.11]

Comme vous pourrez le constater, beaucoup de méthodes d'interaction avec la barre des tâches sont développées sous la forme de méthodes d'extension: [Fig.12]

A l'exécution, l'appel de la méthode permet d'afficher, sur l'icône dans la barre des tâches, une barre de progression: [Fig.13]

b) Construire une « Jumplist »

Les *JumpList* sont les menus contextuels qui apparaissent après que vous ayez fait un clic droit sur l'une des icônes située dans la barre des tâches. Ces listes peuvent être constituées de liens visités fréquemment, de tâches ou bien de vos propres catégories: [Fig.14]

Pour personnaliser la *JumpList* de votre application, il est nécessaire de travailler avec un objet de type JumpListManager. Après l'avoir initialisé avec un appel à la méthode d'extension CreateJumpListManager, il est obligatoire de s'abonner à l'évènement UserRemoveItems, qui est déclenché lorsqu'un utilisateur supprime un élément de la *JumpList*: [Fig.15]

Ensuite, il est possible d'ajouter des éléments dans les catégories Task, Frequent ou bien de définir vos propres catégories: [Fig.16]

Attention, toute modification de la *JumpList* requiert un appel à la méthode *Refresh* pour que celle-ci soit rafraîchie avec les dernières informations. Une fois cet appel effectué, il ne reste plus qu'à lancer l'application pour observer le résultat: [Fig.17]

c) Ajouter une icône à l'image de la barre des tâches

Chaque application située dans la barre des tâches dispose d'une image, qui permet de l'identifier rapidement. Avec Windows 7, il est également possible d'appliquer une icône à chacune de ces images. Pour cela, il suffit une nouvelle fois de faire appel à la méthode d'extension nommée **SetTaskbarOverlayIcon:** [Fig.18]

Lors de l'exécution de l'application, l'icône apparaît correctement au-dessus de l'image représentant l'application, dans la barre des tâches : [Fig.19]

d) Ajouter des boutons aux miniatures de la barre des tâches

Une autre des fonctionnalités offertes par la barre des tâches de Windows 7, c'est la possibilité d'avoir des *buttons* à l'intérieur des miniatures: [Fig.20]

Le wrapper .NET mis à disposition par Microsoft propose une classe nommée **ThumbButtonManager** qui sert justement à définir des boutons personnalisés au niveau des miniatures. Les objets de ce type doivent également être instanciés via un appel à la méthode d'extension **CreateThumbButtonManager**. Ensuite, il suffit d'appeler la méthode **CreateThumbButton** pour créer un bouton. Bien sûr, vous avez aussi la possibilité de vous abonner à l'évènement **Clicked** de ce bouton. C'est l'appel à la méthode **AddThumbButtons** qui vous permet d'insérer le bouton dans la miniature: [Fig.21]

Lorsque l'application est en cours d'exécution, il suffit de passer la souris sur l'image qui est dans la barre des tâches pour voir apparaître la miniature, avec les boutons que vous venez de définir. Notez tout de même que cette miniature n'apparaît que si le thème Aero de Windows 7 est activé: [Fig.22]

e) Personnaliser la miniature

La miniature affichée par Windows 7 peut être personnalisée au moyen d'une classe nommée **CustomWindowsManager** qui est instanciée grâce à la méthode statique **CreateWindowsManager.** Ensuite, il ne vous reste plus qu'à vous abonner aux évènements **PeekRequested** et **ThumbnailRequested**: [Fig.23]

Lorsque ce code est exécuté, la miniature de l'application est personnalisée selon vos souhaits: [Fig.24]

Pour la miniature en elle-même, il ne s'agit que d'un simple Bitmap qui est généré via un appel à la méthode statique **GrabWindowBitmap** située dans la classe statique **ScreenCapture** (celle-ci se trouve dans le fichier *CustomWindowManager.cs*). C'est d'ailleurs dans cette méthode que se trouve le texte *"Windows 7 Bridge"* qui est affiché sur la miniature personnalisée, via un appel à la méthode **DrawString**: [Fig.25]

Fig.8

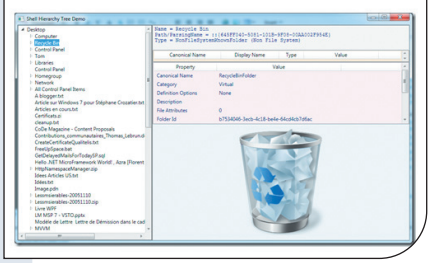


Fig.9

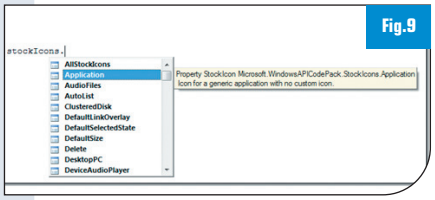


Fig.11

```
progressBar.SetTaskbarProgress();
```

Fig.10

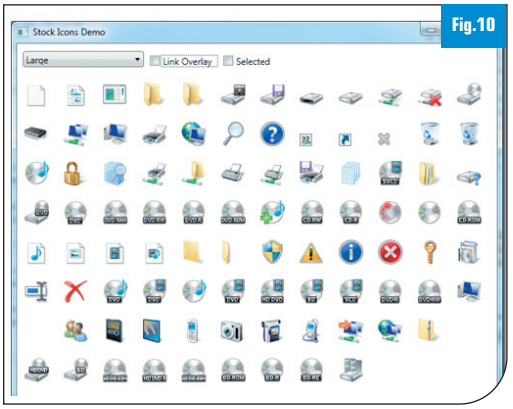


Fig.12

```
/// <summary>
/// Sets the progress bar in the containing form's taskbar button
/// to this toolstrip progress bar's progress.
/// </summary>
/// <param name="progressBar">The progress bar.</param>
public static void SetTaskbarProgress(this ToolStripProgressBar progressBar)
{
    Form form = (Form)progressBar.Control.TopLevelControl;

    //Approximation:
    ulong maximum = (ulong)(progressBar.Maximum - progressBar.Minimum);
    ulong progress = (ulong)(progressBar.Value - progressBar.Minimum);

    Windows7Taskbar.SetProgressState(
        form.Handle, Windows7Taskbar.ThumbnailProgressState.Normal);
    Windows7Taskbar.SetProgressValue(
        form.Handle, progress, maximum);
}
```

Fig.13

Fig.19

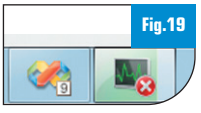


Fig.14

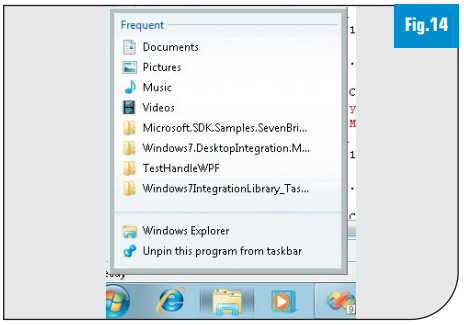


Fig.15

```
_jumpListManager = this.CreateJumpListManager();
_jumpListManager.UserRemovedItems += (o, e) =>
{
    statusLabel.Text = "User removed " + e.RemovedItems.Length +
        " items (cancelling refresh)";
    e.CancelCurrentOperation = true;
};
```

f) Spécifier le contenu de la miniature (clipping)

Par défaut, la miniature qui apparaît dans la barre des tâches représente votre application en entier. Une des nouveautés de Windows 7 est la possibilité de faire du *"clipping"* sur cette miniature et donc d'afficher une partie spécifique de votre application, en tant que miniature. Voici l'image qui s'affiche, par exemple, lorsque la miniature de l'application est entière: [Fig.26]

Cependant, grâce à un appel à la méthode **SetThumbnailClip**, il est possible de spécifier un objet de type **Rectangle** qui définira la partie de l'application à afficher dans la miniature: [Fig.27]

L'appel à cette méthode entraîne bien la création d'une miniature personnalisée: [Fig.28]

g) Assigner l'infobulle

Cette fonctionnalité peut sembler être un gadget mais à bien y réfléchir, on se rend vite compte qu'elle apporte une certaine valeur ajoutée à votre application. Vous pouvez en effet, à tout moment, personnaliser l'infobulle qui apparaît lorsque vous pointez votre souris sur l'icône de votre application, dans la barre des tâches. Pour cela, un simple appel à la méthode d'extension **SetThumbnailTooltip** et le tour est joué: [Fig.29]

Le résultat en image: [Fig.30]

h) Manipuler le Shell

Cette partie n'est pas spécifique à la barre des tâches mais concerne le système d'exploitation en lui-même. Vous avez la possibilité, au sein de votre application, de manipuler les bibliothèques de documents. Il s'agit d'une nouvelle fonctionnalité de Windows 7 qui consiste à offrir aux utilisateurs la possibilité de créer des collections d'éléments qui sont en fait composées de liens vers des fichiers présents sur votre ordinateur. L'avantage est que vous n'avez pas besoin de faire des copier/coller de fichiers dans un dossier: vous indiquez que les fichiers qui vous intéressent doivent se trouver dans la bibliothèque de documents que vous souhaitez et un simple clic sur le nom de cette bibliothèque de documents vous permet de visualiser tous les fichiers que vous avez ajoutés.

Pour manipuler les bibliothèques de documents, il est nécessaire de passer par la classe **ShellLibrary.** Celle-ci possède plusieurs méthodes statiques et non statiques: [Fig.31 et 32]

Le code de la figure 33 vous permet ainsi d'ajouter un répertoire (et son contenu) à la bibliothèque de documents spécifiée et de faire en sorte qu'il soit inséré dans le panneau de navigation.

Lorsque ce code est exécuté, via un simple appel de la méthode *AddFolder*, on constate que le contenu du répertoire Images est bien ajouté à la bibliothèque de documents et que le nom du répertoire est bien inséré dans le Treeview de navigation: [Fig.34]

Et WPF dans tout ça ?

Si vous avez fait attention au code que j'ai montré en images, vous avez sans doute remarqué qu'il est (tout du moins celui lié à la barre des tâches) spécifique aux WindowsForms. En effet, les « wrappers » développés pour le moment ne peuvent-être utilisés que dans des applications WindowsForms car toutes les méthodes d'extension reposent sur le Handle des fenêtres Windows, qui n'existe pas dans le cas des fenêtres WPF. Fort heureusement, grâce à la classe **WindowInteropHelper,** il est possible de récupérer le Handle de ce type de fenêtre: il devrait donc être possible de développer des « wrappers » similaires, pour des applications WPF. Bien sûr, cela ne sera qu'une solution temporaire, en attendant que Microsoft propose quelque chose de mieux (des « wrappers » officiels pour WPF ou la sortie de WPF 4, qui permettra de manipuler, en natif, la barre des tâches).

Les nouvelles API

Comme chaque nouvelle version d'un système d'exploitation, Windows 7 dispose de nouvelles API. Parmi les plus intéressantes, on trouve :

- Les API de capteurs : elles permettent de créer des programmes qui utilisent des capteurs (accéléromètres, luminomètres, etc.) qu'il est possible de connecter à la machine physique. Vous avez la possibilité de récupérer les différents types de capteur par catégorie, type, identifiant. Vous pouvez gérer leurs évènements (être notifié lorsque vous recevez des informations du capteur par exemple), travailler avec les don-

Fig.16

```
string shell32DllPath = Path.Combine(
    Environment.GetFolderPath(Environment.SpecialFolder.System),
    "shell32.dll");

_jumpListManager.AddCustomDestination(new ShellLink
{
    Path = Path.Combine(_thisAppDataPath, "sonnet.txt"),
    Title = "Shall I compare thee...",
    Category = "My Category",
    IconLocation = shell32DllPath,
    IconIndex = 1
});
_jumpListManager.AddCustomDestination(new ShellLink
{
    Path = Path.Combine(_thisAppDataPath, "mary.txt"),
    Title = "Mary had a little...",
    Category = "My Category",
    IconLocation = shell32DllPath,
    IconIndex = 1
});
```

Fig.17

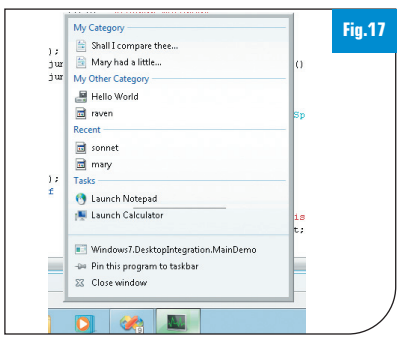


Fig.20

Fig.22

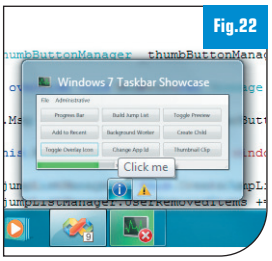


Fig.21

```
ThumbButton button2 = _thumbButtonManager.CreateThumbButton(102,
    SystemIcons.Exclamation, "Beware of me!");
button2.Clicked += delegate
{
    statusLabel.Text = "Second button clicked";
    button2.Enabled = false;
};
ThumbButton button = _thumbButtonManager.CreateThumbButton(101,
    SystemIcons.Information, "Click me");
button.Clicked += delegate
{
    statusLabel.Text = "First button clicked";
    button2.Enabled = true;
};
_thumbButtonManager.AddThumbButtons(button, button2);
```

Fig.18

```
/// <summary>
/// Draws the specified overlay icon over this form's taskbar button.
/// </summary>
/// <param name="form">The form.</param>
/// <param name="icon">The overlay icon.</param>
/// <param name="description">The overlay icon's description.</param>
public static void SetTaskbarOverlayIcon(this Form form, Icon icon, string description)
{
    Windows7Taskbar.SetTaskbarOverlayIcon(form.Handle, icon, description);
}
```

Fig.23

```
_windowsManager.PeekRequested += (o, e) =>
{
    //TODO: This doesn't actually work when
    //the window is minimized...
    e.UseWindowScreenshot = true;
};
_windowsManager.ThumbnailRequested += (o, e) =>
{
    e.UseWindowScreenshot = true;
};
```

Fig.24

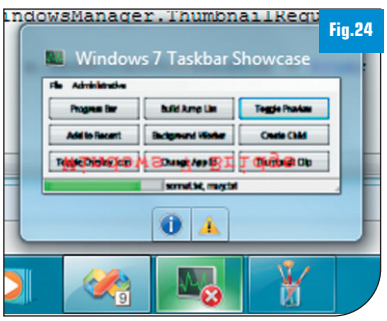


Fig.25

```
targetGr.DrawString("Windows 7 Bridge",
    new Font(FontFamily.GenericMonospace, 12),
    new SolidBrush(Color.Red),
    new PointF(10, 10));

return targetBitmap;
```

nées reçues, etc. Pour accéder à ces donnés et les manipuler, il est nécessaire d'utiliser différentes interfaces COM telles que *ISensorManager, ISensor, ISensorCollection,* etc. Mais là encore, si vous êtes développeur C# ou VB.NET, vous préférerez l'emploi des « wrappers » mis à disposition par Microsoft, qui facilitent leur utilisation : [Fig.35]

- L'API de localisation est une autre des grandes nouveautés de Windows 7. Celle-ci vous permet, par programmation, de récolter des données sur votre situation géographique tout en conservant la vie privée de l'utilisateur. Vous avez alors à votre disposition un ensemble d'interfaces (*ILocation, ILocationReport,* etc.), de méthodes (*GetReport, GetReportStatus,* etc.), d'évènements (*OnStatusChanged, OnLocationChanged,* etc.) que vous pouvez utiliser, directement en C++ ou bien en .NET (avec un peu d'interopérabilité) lorsque vous souhaitez avoir des informations sur votre situation géographique.
- Les API de *« multi-touch »* : Windows 7, sur le matériel adéquat, offre aux applications la possibilité de recevoir des interactions dites « multi-touch ». Ainsi, vous avez la possibilité de détecter et de réagir lorsque plusieurs points de l'écran sont touchés, simultanément, à la surface d'une application.

En interne, Windows envoie le message WM_GESTURE aux applications lorsqu'une manipulation, effectuée par l'utilisateur, est reconnue comme étant une *« gesture »*. Windows 7 dispose d'un ensemble d'API qui encapsulent la création et l'utilisation de ce message. Ainsi, on retrouve des méthodes comme *RegisterTouchWindow, SendMessage, PostMessage, IsTouchWindow,* etc. Lorsque l'on introduit le concept de « multi-touch », on se retrouve très vite confronté à devoir gérer des notions comme les manipulations ou l'inertie (qui est utilisée pour effectuer des calculs permettant d'animer le mouvement des objets, de la façon la plus fluide possible). Là encore, tout passe par des interfaces (*IManipulationProcessor, IInertiaProcessor*) et des évènements (*ManipulationStarted, ManipulationDelta,* etc.). Comme tout passe par des API natives, on apprécierait l'utilisation simplifiée, depuis du code .NET. Avec WPF 4, il s'agit là de quelque chose d'aisé car tout ce qui concerne le « multi-touch » a été intégré directement dans le Framework.

Ainsi, on retrouve des évènements tels que *ManipulationStarted* et *ManipulationDelta* qui permettent aux développeurs de gérer cette fonctionnalité de façon très simple. Malheureusement, WPF 4 n'est pas encore disponible en version finale et tout le monde n'aura pas la chance de pouvoir l'utiliser. Fort heureusement, Microsoft nous a gâté une nouvelle fois, grâce à la mise à disposition d'une autre bibliothèque d'interopérabilité, permettant de gérer le « multi-touch » dans des applications .NET : *« Windows 7 Multitouch .NET Interop Sample Library »* (http://code.msdn.microsoft.com/Project/Download/FileDownload.aspx?ProjectName=WindowsTouch&DownloadId=5038).

Conclusion

Comme vous avez pu le constater au cours de cet article, le développement d'applications pour Windows 7 se révèle simple, y compris pour les développeurs .NET car il n'est plus nécessaire de devoir faire soi-même de l'interopérabilité pour appeler une fonction système ou autre : cela est pris en charge par un ensemble de bibliothèques de code, dont celle qui vous a été présentée dans cet article. Bien sûr, il est à noter que certaines fonctionnalités, comme la manipulation de la barre des tâches, seront disponibles en standard dans .NET 4 / WPF 4, la prochaine version du Framework .NET.

■ Thomas **LEBRUN**
Consultant / Formateur
MVP Client Application Development
http://blogs.developpeur.org/tom

Fig.26

Fig.27

```
private void btnThumbClip_Click(object sender, EventArgs e)
{
    this.SetThumbnailClip(
        new Rectangle(btnThumbClip.Location, btnThumbClip.Size));
}
```

Fig.28

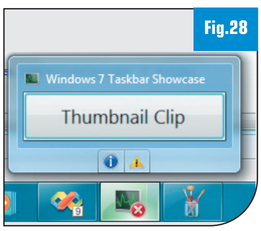


Fig.29

```
/// <summary>
/// Sets the specified form's thumbnail tooltip.
/// </summary>
/// <param name="form">The form.</param>
/// <param name="tooltip">The tooltip text.</param>
public static void SetThumbnailTooltip(this Form form, string tooltip)
{
    Windows7Taskbar.SetThumbnailTooltip(form.Handle, tooltip);
}
```

Fig.30

Fig.31

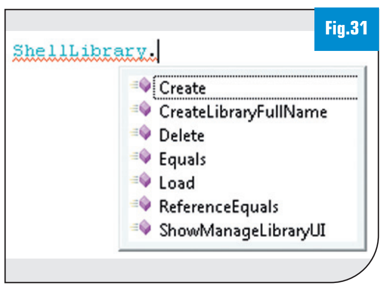


Fig.32

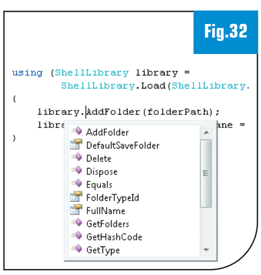


Fig.33

```
public static void AddFolder(string documentLibrary, string folderPath)
{
    using (ShellLibrary library =
            ShellLibrary.Load(ShellLibrary.CreateLibraryFullName(documentLibrary), true))
    {
        library.AddFolder(folderPath);
        library.IsPinnedToNavigationPane = true;
    }
}
```

Fig.34

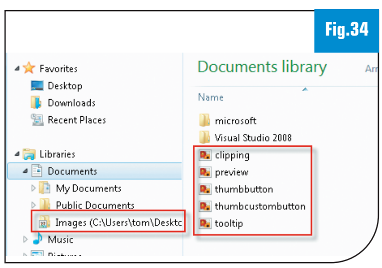
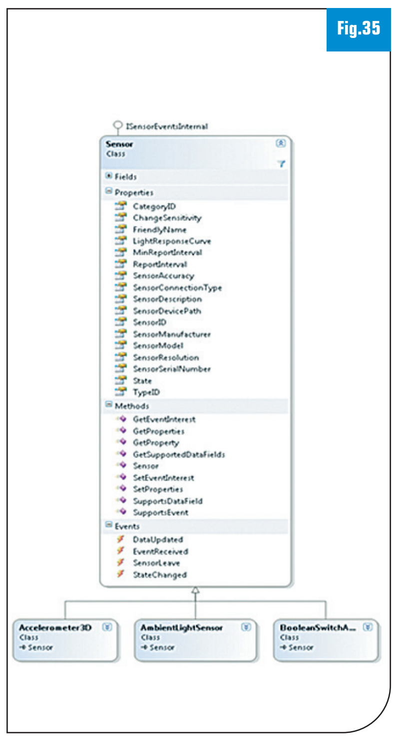


Fig.35

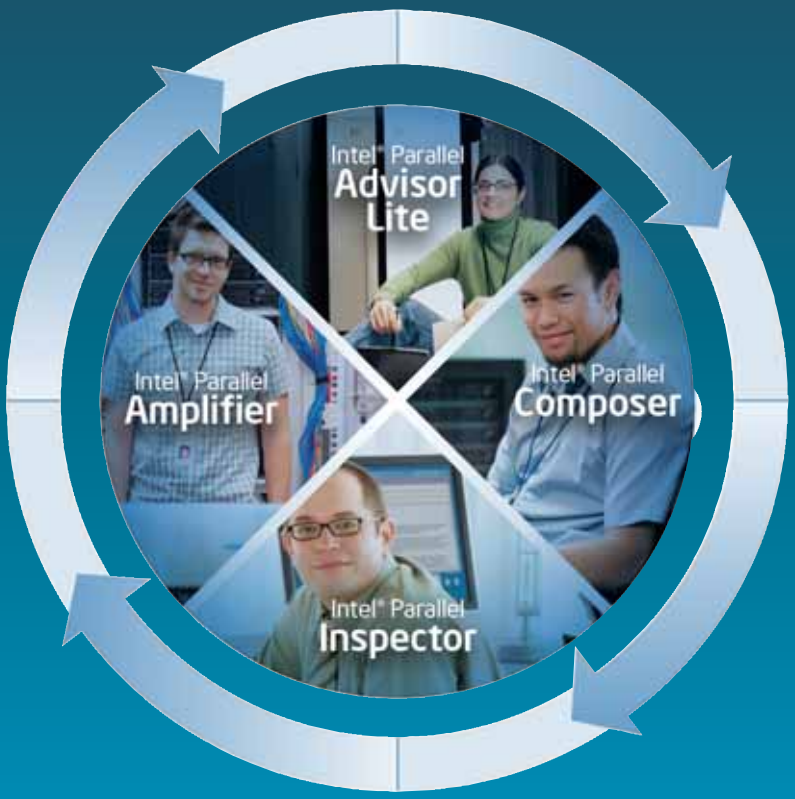


Suite du dossier : *"Le futur est là"* en page 51

Dossier Spécial Développement Parallèle

Le magazine du développement
PROgrammez !
www.programmez.com

FAITES EVOLUER VOTRE CODE !



Notre sélection d'articles de programmation parallèle, publiés en 2009

- **Pourquoi passer à la programmation parallèle ?**
- **Simplifier le développement parallèle**
- **La chasse aux bogues parallèles**
- **Plus de cœurs, plus rapide, plus de montée en charge**
- **Vous ne reconnaîtrez plus votre C++**

Pourquoi passer à la programmation parallèle ?

2e partie

Nous poursuivons ce mois-ci les raisons pour passer à la programmation parallèle et surtout les pièges à éviter et à maîtriser. Et ils sont nombreux !

Deadlock

Parfois nommé étreinte fatale, le deadlock est provoqué par des threads mutuellement verrouillés. Cette situation peut être comparée à deux personnes dessinant des diagrammes tout en se partageant un crayon et une règle. Si une personne prend le crayon et l'autre prend la règle, un blocage se produit lorsque la personne avec le crayon a besoin de la règle pour terminer son travail alors que la personne avec la règle a besoin du crayon pour terminer son travail à son tour. Les deux demandes ne peuvent être satisfaites, un blocage se produit. Pour de nombreux développeurs l'ouvrage sur les Design Patterns du fameux Gang of Four est sans aucun doute un ouvrage de référence pour tous les développeurs. Avez-vous déjà tenté de passer en parallèle un des patterns du GoF ? Par exemple l'Observer. Lors d'une présentation chez Microsoft Research, Edward A. Lee, chercheur et professeur en informatique à l'université UC Berkeley, a montré que l'utilisation de mutex peut rapidement aboutir à des étreintes fatales potentielles. Voici le fragment Java illustrant ce problème.

```
    public synchronized void addListener(listener listener)
    {
        myListeners.add(listener);
    }

    public synchronized void setValue(String newValue) {
        myValue = newValue;

   for (int i = 0; i < myListeners.length; i++) {
            myListeners[i].valueChanged(newValue);
        }
    }
```

En protégeant les méthodes *addListener* et *setValue* à l'aide du mot clef *synchronized,* on pourrait croire que le code ci-dessous supporte une utilisation partagée par plusieurs threads. Mais supposons que la méthode *valueChanged ()* cherche à obtenir un verrou sur un autre objet. Si le titulaire de ce verrou appelle la méthode *addListener (),* alors nous tombons dans le cas d'une étreinte fatale ! Cet exemple illustre une recommandation en programmation parallèle : conservez des *sections synchronisées* les plus courtes possibles.

Livelock

Le *livelock* provoque une situation de famine entre des threads qui se disputent des ressources partagées limitées souvent illustré par le fameux Dîner des philosophes. La situation est la suivante : cinq philosophes se trouvent autour d'une table. Chacun des philosophes a devant lui un plat de spaghetti et à gauche de chaque assiette se trouve une baguette. Un philosophe n'a que trois états possibles :

- Penser pendant un temps indéterminé
- Etre affamé (pendant un temps déterminé et fini sinon il y a famine)
- Manger pendant un temps déterminé et fini

Lorsqu'un philosophe a faim, il va se mettre dans l'état affamé et attendre que les baguettes soient libres pour manger. Un philosophe a besoin de deux baguettes : celle qui se trouve à gauche de sa propre assiette et celle qui se trouve à gauche de celle de son voisin de droite (c'est-à-dire les deux baguettes qui entourent sa propre assiette). Si un philosophe n'arrive pas à s'emparer d'une baguette, il reste affamé pendant un temps déterminé, en attendant de renouveler sa tentative. Le problème consiste à trouver un ordonnancement des philosophes tel qu'ils puissent tous manger, chacun à leur tour. Une solution à base de sémaphores, énoncée par Dijkstra, est souvent utilisée en programmation parallèle.

Le dîner des philosophes est souvent l'objet d'examens informatiques. Voici une implémentation Java du dîner des philosophes soumis dans le cadre d'un examen informatique en 2005.

```
import java.util.ArrayList;
import java.util.Random;

class Dining {
    Random r;
    ArrayList<Philosophe> p;
    ArrayList<Baguette> c;
    static final int NUM_PHIL=5;
    Dining() {
 p = new ArrayList<Philosophe>();
 c = new ArrayList<Baguette>();
 r=new Random();
 for (int i=0;i<NUM_PHIL;i++) {
     c.add(new Baguette());
 }

 for (int i=0;i<NUM_PHIL;i++) {
     p.add(new Philosophe(i));
 }
    }

    public void eat() {
 for (Philosophe phil : p) {
     phil.start();
 }
    }

    class Philosophe extends Thread {
 Baguette gauche;
 Baguette droite;
```

```
 int idx;
 Philosophe(int idx) {
     super();
     gauche=c.get(idx);
     droite=c.get((idx+1)%NUM_PHIL);
     this.idx=idx;
 }

 public void run() {
     while (1==1) {
   try {
       sleep(r.nextInt(1000));
       gauche.get();
      System.out.println("Le philosophe " + (idx+1) + " a pris
la baguette a gauche");
       sleep(r.nextInt(1000));
       droite.get();
      System.out.println("Le philosophe " + (idx+1) + " a pris
la baguette a droite");
      System.out.println("Le philosophe " + (idx+1) + " a mange!");
       sleep(r.nextInt(1000));
       gauche.put();
      System.out.println("Le philosophe " + (idx+1) + " a remis
la baguette a gauche");
       droite.put();
      System.out.println("Le philosophe " + (idx+1) + " a remis
la baguette a droite");
   } catch (Exception e) {
       System.out.println(e);
   }
     }
 }
    }
    class Baguette {
 private boolean taken=false;
 private int identity;

 Baguette () {
 }
 synchronized void put() {
     taken=false;
     notify();
 }
 synchronized void get() throws java.lang.InterruptedException {
     while (taken) wait();
     taken=true;
 }
    }

    public static void main(String [] args) {
 Dining d = new Dining();
 d.eat();
    }
```

Cette question est en 2 parties :

• Définissez clairement le concept de deadlock dans les systèmes multi threads. Expliquez comment un deadlock pourrait se produire dans le programme. Soyez très précis et expliquez clairement où le deadlock se produirait dans le programme (rapportez les lignes exactes de cela dans le code vu plus haut) et pourquoi (l'ordonnancement des évènements qui causerait le blocage.)

• Faites les changements nécessaires à public void run() et à la classe Baguette de manière à ce qu'il n'y ait plus de deadlock. Il se peut que vous trouviez cela difficile. Si vous n'y arrivez pas, assurez-vous au moins d'expliquer clairement votre stratégie.

L'intérêt ici n'est pas de répondre aux questions posées mais de juger de la difficulté pour y répondre.

Malheureusement il existe bien d'autres problèmes encore un peu plus subtils dont les conséquences peuvent être désastreuses. Tous ces problèmes rendent la programmation parallèle très difficile et souvent repoussante pour les développeurs.

Pourquoi les codes parallèles sont-ils difficile à maintenir ?

Un autre point regrettable avec l'usage des technologies actuelles est la complexité d'écriture pour implémenter du code parallèle. Pour illustrer ce point, prenons deux exemples. Le premier implémente une méthode naïve en C++0x dédiée à la multiplication de matrices.

```
void MatrixMult(int size, double** m1, double** m2, double** result)
{
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            result[i][j] = 0;
            for (int k = 0; k < size; k++) {
                result[i][j] += m1[i][k] * m2[k][j];
            }
        }
    }
}
```

Le second exemple implémente cette fois en C# une méthode dédiée à la recherche multicritères dans une liste mémoire de bébés.

```
IEnumerable<BabyInfo> Search(IEnumerable<BabyInfo> babies, Query
Info qi)
{
    var results = new List<BabyInfo>();
    foreach(var baby in babies)
    {
        if (baby.Name == qi.Name &&
            baby.State == qi.State &&
            baby.Year >= qi.YearStart &&
            baby.Year <= qi.YearEnd)
        {
            results.Add(baby);
        }
    }
    results.Sort((b1, b2) =>
        b1.Year.CompareTo(b2.Year));

    return results;
}
```

Adaptons ces codes afin d'exploiter les technologies parallèles actuelles. En premier le code C++.

```
void MatrixMult(int size, double** m1, double** m2, double**
 result) {
  int N = size;
  int P = 2 * NUMPROCS;
  int Chunk = N / P;
  HANDLE hEvent = CreateEvent(NULL, TRUE, FALSE, NULL);
  long counter = P;
  for (int c = 0; c < P; c++) {
    std::thread t ([&,c] {
      for (int i = c * Chunk;
           i < (c + 1 == P ? N : (c + 1) * Chunk); i++) {
        for (int j = 0; j < size; j++) {
          result[i][j] = 0;
          for (int k = 0; k < size; k++) {
            result[i][j] += m1[i][k] * m2[k][j];
          }
        }
      }
      if (InterlockedDecrement(counter) == 0)
        SetEvent(hEvent);
    });
  }
  WaitForSingleObject(hEvent,INFINITE);
  CloseHandle(hEvent);
}
```

En second le code C#.

```
IEnumerable<BabyInfo> Search(IEnumerable<BabyInfo> babies, Query
Info qi)
{
    var results = new List<BabyInfo>();
    int partitionsCount = Environment.ProcessorCount;
    int remainingCount = partitionsCount;
    var enumerator = babies.GetEnumerator();

    try {
      using (var done = new ManualResetEvent(false)) {
        for(int i = 0; i < partitionsCount; i++) {
          ThreadPool.QueueUserWorkItem(delegate {
            var partialResults = new List<BabyInfo>();
            while(true) {
              BabyInfo baby;
              lock (enumerator) {
                if (!enumerator.MoveNext()) break;
                baby = enumerator.Current;
              }
              if (baby.Name == qi.Name && baby.State == qi.State &&
                  baby.Year >= qi.YearStart && baby.Year <= qi.YearEnd) {
                partialResults.Add(baby);
              }
            }
            lock (results) results.AddRange(partialResults);
          if (Interlocked.Decrement(ref remainingCount) == 0) done.Set();
          });
        }
        done.WaitOne();
        results.Sort((b1, b2) => b1.Year.CompareTo(b2.Year));
      }
    }
    finally
    {
      if (enumerator is IDisposable) ((IDisposable)enumerator)
.Dispose();
    }

    return results;
}
```

Sans être un expert du multithreading, vous conviendrez que ces nouvelles versions sont bien plus compliquées et regorgent de défauts importants : le partitionnement statique du traitement sans adhérence avec le nombre de CPU présents sur la machine courante, un mécanisme de synchronisation trop intrusif et très coûteux sur le plan des ressources (surcharge CPU provoquée par des passages fréquents du mode utilisateur au mode noyau), un usage direct de l'entité thread couplée au système d'exploitation entraînant des coûts substantiels pour le système d'exploitation (1 Mo de mémoire par thread). On note que le Framework .NET n'échappe pas à l'héritage de la culture séquentielle, mais son incarnation est relativement moins rugueuse qu'en langage C++. Par nature, le code géré est plus simple à appréhender, cependant il reste compliqué lorsqu'on parle de parallélisme.

"Des logiciels écrits avec des threads, des sémaphores, et des mutex sont incompréhensibles pour les humains et ne devraient pas être dignes de confiance.": Edward A. Lee chercheur / professeur en informatique à l'université UC Berkeley.

CONSÉQUENCES POUR LES ENTREPRISES

Pourquoi les entreprises boudent-elles le parallélisme ?

Du côté des entreprises, la crise actuelle n'est pas forcément propice pour tenir compte de cette évolution matérielle. Nous venons d'un univers où les Hertz nous apportaient régulièrement un surplus de puissance et le récent passage à des matériels multi-cœurs pour tous les postes de travail, passe relativement inaperçu auprès des DSI. Pour les entreprises, les développeurs doivent avant tout se concentrer sur leur métier en conjuguant technique et fonctionnel, et non sur des problèmes de programmation parallèle qui sont incompatibles avec leurs attentes. En effet, les problèmes de multithreading sont chronophages et risqués. Comment être certain que les tests seront fiables alors que le multithreading est par nature indéterminisme ? Comment s'assurer que la qualité du produit final est conforme aux attentes de toutes les plates-formes d'exécution que le produit rencontrera au cours de son existence ?

Pour toutes ces raisons, les entreprises, en général, ne sont pas encore prêtes pour que les équipes de développement tirent parti des nombreux cœurs qui équiperont leurs prochains achats. Quelques rares cas, où traditionnellement les besoins fonctionnels réclament des performances élevées (moteurs de recherche, éditeurs de bases de données, site web d'une audience mondiale, Imagerie médicale ...), font exception car ils disposent déjà d'experts sur le sujet. Notons que la société Evans Data, cabinet d'analyse, rapporte, dans une étude sur les habitudes des développeurs mondiaux, que sur les six derniers mois de 2008, seulement **3 %** des développeurs auraient introduit dans leur code des instructions capables de gérer le parallélisme des multi-cœurs.

Les entreprises se tourneront vers le parallélisme lorsque le marché sera mature

Le passage à la programmation parallèle pour l'essentiel des entreprises sera sans doute long et coûteux. La mise à niveau des développeurs applicatifs est essentielle mais elle ne se fera que lorsque les offres de programmation permettront de développer du code parallèle sans être un expert du sujet. Les offres actuelles sont à la fois trop rugueuses et trop compliquées pour qu'un développeur applicatif puisse en tirer parti facilement. Enfin, même en disposant de nouvelles offres de programmation moins élitistes, le besoin de formation sera sans doute nécessaire. D'un côté, les universités et les écoles devront augmenter la part des cours consacrés à la programmation parallèle. D'un autre côté, les établissements de formation devront ajouter à leur catalogue des formations permettant de développer en parallèle efficacement. L'objectif sera de permettre à une large audience de développeurs d'acquérir des connaissances suffisantes pour produire du code parallèle de bonne qualité sans être un expert du sujet.

Il est parfaitement imaginable que de nombreuses entreprises décident de se lancer dans des projets de migration pour adapter leurs codes séquentiels au monde parallèle. Lorsque ce moment viendra, les offres logicielles permettant de développer plus facilement du code parallèle devront être disponibles pour une large population de développeurs. Ces prochaines offres devront non seulement proposer de nouveaux modèles de programmation, mais aussi de nouveaux outils afin de faciliter la mise au point, les tests, le profilage des codes parallèles. Naturellement, ces prochains outils devront être accessibles au plus grand nombre de développeurs. C'est donc un énorme challenge que doivent résoudre les éditeurs de logiciels de développement.

En conclusion

Comme nous l'avons mentionné, la banalisation des processeurs multi-cœurs sur le marché grand public a rendu urgent le besoin d'une démocratisation des technologies de programmation parallèle. Les développeurs ne pourront pas développer à grande échelle des applications parallèles avec les offres actuelles.

La balle est donc bien dans le camp des éditeurs logiciels qui se doivent de faciliter la programmation parallèle. Les langages plus populaires que sont Java, .NET et C++ doivent recevoir de la part des éditeurs de langages des solutions qui adressent un public large.

Dans un premier temps, ces nouvelles offres ne suffiront néanmoins pas à rendre autonomes des millions de développeurs. Sur le plan de l'architecture logicielle, de nouvelles règles devront êtres apprises sur la manière de modéliser ou de penser les algorithmes parallèles. Les écoles, les universités et les établissements de formation devront accompagner cette mutation technologique qui s'étalera sans doute sur plusieurs décennies.

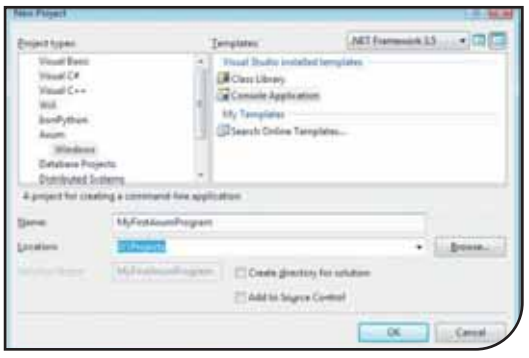
Nous sommes donc face à un point d'inflexion qui engage toute l'industrie informatique à réviser ses modèles techniques. Les conséquences sur notre économie sont difficiles à prévoir, mais faisons confiance à l'imagination humaine pour trouver de nouveaux usages tirant parti de cette nouvelle donne technologique.

Ressources

- Loi de Moore: http://fr.wikipedia.org/wiki/Loi_de_Moore
- Intel micro architecture Netburst: http://fr.wikipedia.org/wiki/NetBurst
- The Free Lunch is Over: http://gotw.ca/publications/concurrency-ddj.htm
- Intel processeur Nehalem: http://fr.wikipedia.org/wiki/Nehalem
- Principe d'incertitude: http://fr.wikipedia.org/wiki/Principe_d'incertitude
- Dîner des philosophes: http://fr.wikipedia.org/wiki/D%C3%AEner_des_philosophes
- Making Concurrency Mainstream: http://ptolemy.eecs.berkeley.edu/~eal/videos/EALatMSR_Jan07/lecture.htm
- Résolution de 11 problèmes possibles dans votre code multithread: http://msdn.microsoft.com/fr-fr/magazine/cc817398.aspx
- Windows 7 capable de gérer 256 cœurs: http://blogs.msdn.com/devpara/archive/2009/03/09/windows-7-et-le-concurrency-runtime.aspx
- Le b-a ba du développement parallèle avec la plate-forme .NET: http://msdn.microsoft.com/fr-fr/vcsharp/dd582579.aspx
- Pourquoi les développeurs boudent le multi-cœur: http://www.lemagit.fr/article/processeur-developpement-parallelisme/2369/1/pourquoi-les-developpeurs-boudent-multi-coeur/

■ Bruno **Boucard** (boucard.bruno@free.fr) est spécialisé dans les technologies Microsoft, anime avec d'autres spécialistes le blog Développement parallèle de Microsoft France: http://blogs.msdn.com/devpara/default.aspx

Le projet Axum de Microsoft : un langage parallèle



Axum est un projet interne à l'éditeur visant à concevoir un langage dédié à la montée en charge, la sécurité des applications, et la réactivité, tout en proposant une simplification dans le développement. Axum est avant tout un langage de développement parallèle dans un contexte .Net. Pour le moment, Axum est en incubation dans les laboratoires et donc susceptible d'être modifié. Il reprend une syntaxe proche du C et du C# tout en s'inspirant des langages Ada et SR. Axum reprend donc de nombreux éléments de C# 3 comme les expressions, les requêtes LINQ, les déclarations de méthodes ou encore les types. Il définit plusieurs catégories dites de haut niveau : les channels, schéma, réseaux, agents et domaines, contrôles de flux. Pourquoi créer un nouveau langage dédié au parallèlisme ? Pour l'équipe d'Axum, le parallélisme a besoin d'un langage adapté répondant aux spécificités de ce modèle de développement, tout en simplifiant la programmation et notamment tout ce qui concerne la gestion des threads ou encore la synchronisation et la coordination.

Site : http://msdn.microsoft.com/en-us/ devlabs/dd795202.aspx

Simplifier le développement parallèle !

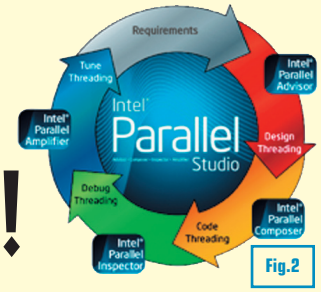
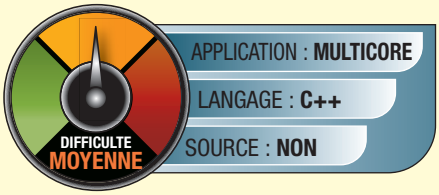


Fig.2

Alors que les processeurs multi-cores (multi-cœurs) prolifèrent, améliorer les performances logicielles, et mieux utiliser la puissance processeur, demeure un enjeu important pour rester compétitif, surtout pour les applications développées en C ou C++. Mais jusqu'à présent le développement parallèle restait complexe à aborder pour le développeur. Programmez ! vous invite à découvrir la nouvelle solution Intel, avec un tout nouveau plug-in Visual Studio, qui va radicalement changer votre vision du développement parallèle.



L'année 2009 sera non seulement l'année de la prolifération en masse des processeurs à 4 cœurs (Quad-Core) et le début des processeurs à 8 cœurs (Octo-Core), mais aussi l'année ou le développeur sera de plus en plus confronté aux performances des applications, de son code source sur ces processeurs. Bref, comment tirer parti des performances pour accélérer les applications Windows ? Les techniques de programmation dite parallèle et le " multi-threading " sont difficiles à maîtriser. Elles requièrent souvent un travail intensif d'apprentissage et de structuration et généralement une refonte partielle ou totale du code. Les outils d'aide à l'implémentation du multi-threading disponibles sur le marché sont rares et encore souvent complexes : Intel propose depuis quelques années des outils et des librairies multi-plates-formes pour la programmation sous C et C++. Jusqu'à présent, ces outils étaient surtout réservés aux développeurs de calculs hautes-performances ou à l'imagerie numérique.

PARALLEL STUDIO : LE PREMIER PAS !

Annoncée en août 2008, la collaboration entre Intel et Microsoft a abouti au lancement d'Intel Parallel Studio. Celle-ci supportera également la future Concurrency Runtime de Microsoft qui permettra finalement d'avoir une infrastructure commune de gestion des ressources cœurs de processeur pour toutes librairies et langages, que ce soit .net ou natif. Intel Parallel Studio se présente sous la forme de plug-in comprenant 4 modules distincts (et utilisables séparément) qui permettront d'aborder la programmation parallèle directement sous l'environnement Visual Studio. Aujourd'hui, 3 des 4 modules sont disponibles et prêts à être installés :

1. Parallel Composer
2. Parallel Inspector
3. Parallel Amplifier
4. Parallel Advisor (disponible ultérieurement)

UNE INSTALLATION SIMPLE ET UNE PARFAITE INTÉGRATION

Les 3 modules sont simples à installer à partir d'un installateur commun et s'intègrent parfaitement dans l'environnement Visual Studio 2005 ou 2008. Chaque module peut-être installé séparément.

Après installation, les modules sont disponibles soit directement en barre d'outils [Fig.1]soit par le menu "outils " (" tools ").

Partant d'une méthodologie éprouvée de développement parallèle en 4 phases [Fig.2], chaque module s'imbrique dans une de ces phases :

1. Phase **" Design "** (Parallel Advisor), répondant aux questions par où commencer ? quels segments de code bénéficieront le plus du multi-threading ?
2. Phase **" Codage "** (Parallel Composer)
3. Phase **" Debug "** (Parallel Inspector)
4. Phase **" Tuning "** (Parallel Amplifier)

Tous les modules sont utilisables et compatibles avec le compilateur Microsoft Visual C++ (et aussi avec le compilateur Intel C++).

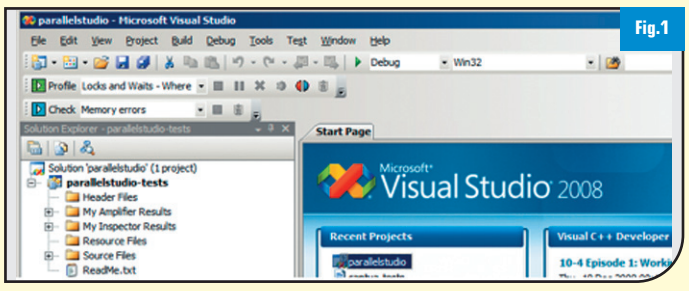


Fig.1

Les 2 barres d'outils pour Inspector (" Check ") et Amplifier (" Profile ")

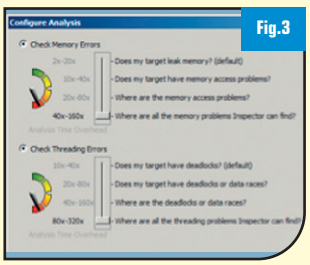


Fig.3

Configuration des analyses du module Inspector

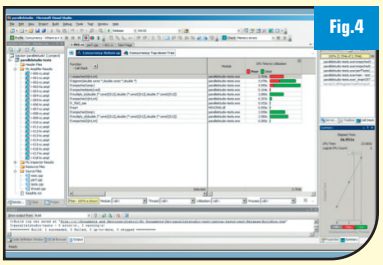


Fig.4

Analyse du niveau de parallélisme ("concurrency analysis")

Parallel Composer complémente Visual Studio avec de nouvelles fonctionnalités et méthodes de multi-threading au niveau applicatif : il comprend un compilateur haute performance C/C++, compatible avec le compilateur Microsoft ainsi que la librairie multimédia Intel Performance Primitives (IPP) et la librairie de multi-threading Threading Building Blocks (TBB). Le compilateur supporte dès à présent le standard OpenMP 3.0, les fonctions *lambda,* ainsi que les fonctions : *autovectorization, auto-parallelization* et *spawn.* La librairie TBB est un élément clé du Composer. Elle offre une nouvelle approche relativement simple du multi-threading basée sur des templates C++, ce qui rend le code aisément portable et très performant.

Parallel Inspector est l'outil le plus innovant du trio (en attendant le quatrième outil Parallel Advisor). En principe, il s'agit d'un debugger spécifiquement adapté aux méthodes de développement parallèle. Le but étant de donner confiance au développeur et que son code ne va pas générer des bugs imprévisibles et destructifs.

Inspector permet de trouver rapidement et sans failles les erreurs classiques et difficiles à détecter du multi-threading : les " race conditions " et les " deadlocks ". Inspector intègre également un analyseur de mémoire très performant pour détecter les erreurs et les fuites de mémoire. L'outil, très flexible, peut être facilement configuré pour varier l'intensité et la durée des analyses, et permettre ainsi de faire des tests rapides ou bien des tests exhaustifs [Fig.3].

Parallel Amplifier est l'outil d'analyse de la performance et se place dans la 4e phase de la méthodologie (phase de " tuning "). L'outil permet de trouver les goulots d'étranglement et de donner des éléments de réponse à la question cruciale du développement parallèle : est-ce que mon code utilise au mieux les ressources et le potentiel de performance du processeur multi-core ?

Les fonctionnalités de Parallel Amplifier sont multiples et se basent principalement sur des analyses de performance en profondeur au niveau du processeur :

1. Analyse "Hotspot" des segments de code gros consommateur de ressources processeurs.
2. Analyse du niveau de parallélisme ("concurrency analysis") permettant d'une part de repérer les segments de code qui sont parallèles et ceux qui ne le sont pas, et d'autre part d'identifier les segments de code qui sont exécutés de façon sérielle et limitent ainsi la montée en charge (" scalability "), voir [Fig.4].
3. Analyse de "Locks and Wait" pour détecter les objets de synchronisation et de blocage qui consomment le plus de temps d'attente.

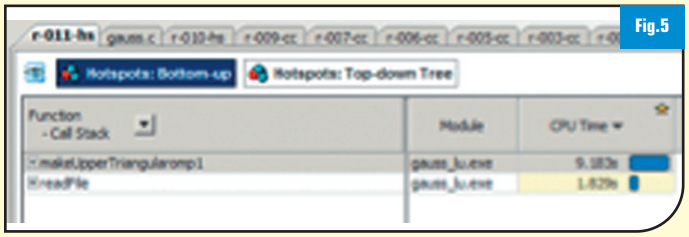


Fig.5

La première fonction crée un " hotspot "

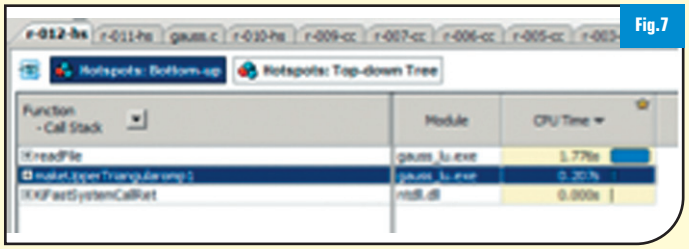


Fig.7

UNE MISE EN ŒUVRE PERMANENTE

L'intérêt principal de Parallel Studio réside dans la simplicité d'utilisation et l'intégration dans Visual Studio, ce qui permet une mise en œuvre permanente tout au long du processus de développement.

EXEMPLE SIMPLE

Prenons l'exemple d'un algorithme de résolution de matrice creuse (en anglais, " sparse matrix solver") utilisant la méthode par élimination de Gauss (méthode LU); un algorithme classique pour résoudre des problèmes d'éléments finis en mécanique.

Après compilation du code en C++, nous utilisons Amplifier pour trouver un " hotspot " éventuel. L'écran de résultats nous montre une utilisation du processeur excessive pour une des fonctions [Fig.5].

Un double-click sur la fonction nous envoie sur les lignes code responsables du " hotspot " : [Fig.6]. Après analyses des matrices, nous changeons le code de la façon suivante :

```
#pragma omp parallel for private (i,j,k, term, temp)

      for (i = 0;i < dimension - 1; i++){
            temp = A[i][i];
            colindex = (i >> 1) << 1;
            for (j = i + 1 ; j < dimension; j++){
                  term = A[j][i] / temp;
                  //for (k = i; k < dimension; k++){
                  for (k = colindex; k < colindex + 2; k++){
                        A[j][k] = A[j][k] - term * A[i][k];
                  }
                  b[j] = b[j] - term * b[i];
            }
      }
```

Le résultat est immédiat : [Fig.7]. Un des grands avantages de Parallel Studio est de permettre des analyses comparatives côte-à-côte [Fig.8]. Cet exemple relativement simple montre la facilité d'utilisation de Parallel Studio dans le quotidien de l'utilisateur de Visual Studio. Le parallélisme va devenir une impérative constante et normale dans le processus de développement logiciel.

En attendant, nous vous recommandons d'installer Parallel Studio (ou bien le ou les modules qui vous intéressent) et de commencer immédiatement avec de premières expériences. Plus d'informations sont aussi disponibles en anglais sur le site http://www.go-parallel.com/.

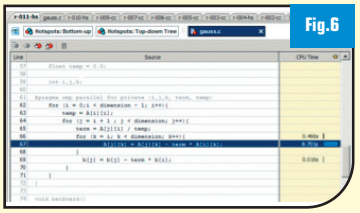


Fig.6

■ Ralph de **Wargny**

Directeur de la division outils de développement d'Intel pour la région Europe centrale et du sud.

ralph.wargny@intel.com

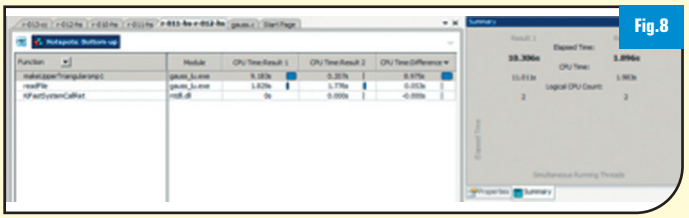
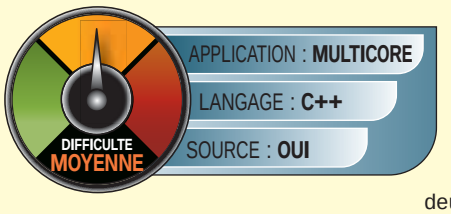


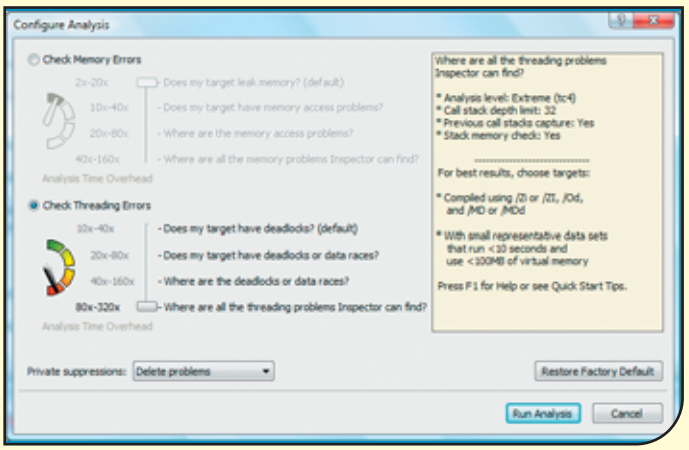
Fig.8

La chasse aux bogues parallèles

Nous avons été formés à programmer en mode sériel et passer à la programmation parallèle n'a rien d'évident, mais plusieurs technologies existent pour vous faciliter la vie : Intel Threading Building Blocks ou le standard OpenMP par exemple.



Une fois les concepts de base assimilés, implémenter un programme parallèle qui utilise à peu près efficacement un processeur à deux ou quatre cœurs va assez vite si votre algorithme s'y prête. Mais il est surtout très facile de transformer un logiciel sériel sans bogue en un logiciel multithreadé fortement bogué sans que ce soit évident, ni à détecter ni à corriger. Pour répondre à ce problème, Intel a compris très tôt qu'il fallait aider les développeurs avec des outils logiciels. Pour les développeurs sous Visual Studio en C / C++ cet outil est Intel Parallel Studio, et plus spécifiquement Parallel Inspector pour la partie isolation et correction de bogues parallèles que nous allons étudier aujourd'hui.



UN EXEMPLE SIMPLE POUR VISUAL STUDIO

Partons d'un exemple concret : un logiciel qui trouve les nombres premiers dans un intervalle donné. Partons du code parallélisé par OpenMP, après ajout d'une simple ligne de "pragma" au code sériel.
Visual studio (2005 ou 2008) doit être installé, puis téléchargez et installez Intel Parallel Studio. Créez un nouveau projet nommé "premiers" d'application en ligne de commande et décochez les en-têtes précompilés.
Remplacez le contenu de "premiers.cpp" par le contenu du fichier téléchargé et sauvez, ensuite cliquez droit dans l'explorateur de solution sur "premiers". Dans le menu vous avez un sous-menu "Intel Parallel Composer" en plus, dans lequel vous pouvez choisir d'utiliser le compilateur Intel. Cliquez et observez la nouvelle icône bleue "C++" utilisée pour votre solution. Nous avons un code qui fait appel à OpenMP, mais encore faut il activer la bonne option de compilation : demandez les propriétés de votre projet, allez dans "C / C++ / Langage / General". Vous avez un item OpenMP support que vous devez mettre a / OpenMP. Maintenant, compilez en mode debug et exécutez-le plusieurs fois sur une machine parallèle (Multi-Core, HyperThreading, ...) en surveillant dans le gestionnaire de processus Windows que vous utilisez plusieurs cœurs.

EXEMPLE SIMPLE MAIS BOGUÉ

Le décompte de nombre premiers est différent pour des exécutions successives, ce qui est bien sûr impossible, nous avons donc un bogue. La version sérielle n'avait pas ce problème, il s'agit donc d'un problème dû à l'exécution parallèle. En rendant notre logiciel parallèle avec OpenMP nous avons probablement omis de protéger certaines parties, mais lesquelles ?
Il est parfois facile comme ici de détecter manuellement un bogue parallèle, dans certaines conditions expérimentales c'est presque impossible. Le bogue parallèle est un animal sournois et dur à chasser, il peut se montrer très discret sur votre station de développement et n'apparaître que sur votre serveur de production.
Intel Parallel Inspector est capable de détecter des problèmes potentiels dans l'exécution parallèle même s'ils n'arrivent pas dans les conditions de votre exécution. Il suffit juste d'exécuter la partie de votre logiciel que vous souhaitez analyser compilée en parallèle dans Parallel Inspector et il trouvera à coup sur les problèmes dans votre code, magique ! Il faut qu'Inspector ait accès aux symboles pour remonter jusqu'au code source à l'endroit précis du problème, donc compilez en mode debug.

ANALYSE DES ERREURS DE THREADING

Lancez maintenant une analyse directement depuis Visual Studio par le menu "Outils / Intel Parallel Inspector / Check threading errors". Vous pouvez demander différents niveaux d'analyse : Parallel Inspector instrumente les accès mémoire, ce qui est assez lent en soi mais qui ralentira aussi l'exécution. Il faut donc soumettre une charge de travail réduite pour que l'exécution ne dure pas trop longtemps. Par contre cela n'influe pas sur la fiabilité des résultats, nous cherchons ici à détecter des bogues, pas à augmenter ou mesurer la performance.

Au début de l'exécution, vous voyez défiler dans les logs les librairies utilisées et celles qui sont compilées en debug, elles sont instrumentées, ce qui pour un gros logiciel est une opération intensive, mais qui sera réutilisée pour de futures exécutions.

UN RÉSULTAT CLAIR ET CONCIS RELIÉ À VOTRE SOURCE

Voici le rapport qui apparaît : vous avez une liste des problèmes en haut, P1, P2, P3 qui sont dans les 3 cas des “data race”, soit des interactions de différents threads vers les mêmes variables. Pour chaque problème vous avez dans la liste du bas un détail des interactions. En effet, pour un problème donné exécuté par différents threads vous aurez une liste d'interactions entre différents couples de threads. Ici c'est l'exécution de la même ligne 73 du source qui interagit entre différents threads, double-cliquez sur la ligne et vous verrez la vue complète de 2 threads exécutant deux morceaux de code en même temps (ici le même code, mais pas toujours). La ligne 73 est une variable partagée (gProgress) qui est incrémentée, tout simplement. Une incrémentation est une lecture puis une écriture, selon que la lecture dans le thread du haut arrive avant ou après l'écriture du thread du bas, le résultat est différent. Et bien sûr, deux écritures simultanées sont aussi un problème.

C'est donc une erreur de programmation parallèle ! Toute variable partagée entre différents threads et accédée en écriture doit être protégée. Double-cliquez sur le source pour aller dans l'édition du source à la bonne ligne. La solution est déjà dans le code fourni, ligne 72 qu'il suffit de décommenter en laissant le dièse en début de ligne (la solution proposée n'est pas optimale mais fonctionne). Une section critique s'assure que la région protégée (ici la ligne qui suit) ne sera exécutée que par un seul thread à la fois.

Retour au rapport, allez a l'erreur P2 ligne 103 et décommentez la ligne 102 qui est la solution. Puis P3, qui est aussi ligne 102. Ce n'est pas un bogue de l'outil, mais simplement que nous avons deux bogues bien distincts sur la même ligne, une concerne le tableau, l'autre la variable incrémentée. L'outil raisonne en termes d'accès mémoire, de variables et vous indique les lignes fautives.

Sauvez le source, fermez le rapport et recompilez. Vous pouvez maintenant relancer l'analyse d'erreurs de threads. Un bogue peut en cacher un autre. Cette fois nous avons un bogue un peu plus complexe, une interaction entre 2 threads mais qui exécutent un morceau différent du code, lignes 75 et 73. La ligne 75 lit la variable gProgress qui est incrémentée juste avant, ligne 73. L'incrémentation en elle-même est maintenant protégée grâce a la section critique, mais si le thread du bas incrémente la variable alors que le thread du haut passe de la ligne 73 à 75 nous avons un bogue. Il faut donc s'assurer que tout le bloc 73 à 78 est exécuté par un seul thread à la fois d'une seule traite, en pratique étendre la section critique avec des accolades, ouverte ligne 73 et fermée ligne 78. Recompilons et exécutons à nouveau dans Parallel Inspector: cette fois 0 erreur ! Nous avons maintenant un logiciel parallèle sans bogues de parallélisme !

LES LIBRAIRIES AUSSI !

Vous avez corrigé les bogues que vous avez créés, mais en tant que développeur vous êtes responsables de tout ce qui est appelé par votre application. En effet, toute librairie lancée depuis une région parallèle doit aussi être garantie sans bogues parallèles. Bonne nouvelle, Parallel Inspector trouvera aussi ces bogues, par contre, faute de version debug de ces librairies et du code source vous ne pourrez les corriger, mais au moins vous saurez qu'il ne faut pas utiliser ces librairies en parallèle.

CONCLUSION

Vous avez pu voir dans cet exemple que les bogues parallèles sont complexes à isoler et corriger mais qu'Intel Parallel Inspector facilite et accélère grandement le travail. Il ne se substitue pas à l'apprentissage des concepts de base du parallélisme, mais vous accompagne dans votre projet en garantissant la qualité du code et des librairies.

Si nous exécutons plusieurs fois le nouveau binaire, le résultat est cette fois le même, ce qui est bon signe, mais le temps d'exécution est plus long, C'est une des vérités de la programmation parallèle, un logiciel bogué va souvent plus vite !

■ Paul **Guermonprez**
Ingénieur logiciel – Intel.

Démarrez le développement parallèle avec la toute dernière boîte à outils Intel®

Consultez la section spéciale de programmez.com/intel.php
Téléchargez la version d'évaluation, vidéos, actu, pratique, etc.

Plus de cœurs, plus rapide, plus de montée en charge : du multi-core au " many-core "

Ecrire des applications parallèles se simplifie, et nous avons vu précédemment que chasser les bugs parallèles était rapide et intuitif avec Intel Parallel Inspector. Reste maintenant le principal problème : utiliser pleinement tous les cœurs simultanément.

Une station de travail multi-core a typiquement 8 cœurs, 16 vus par l'OS avec la nouvelle technologie SMT (Simultaneous Multi Threading) et dans l'avenir beaucoup plus. Le consensus actuel dans l'industrie des processeurs est que les gains de performance viendront essentiellement par l'ajout de cœurs : nous sommes à l'orée de l'ère du " many-core ".

QUATRE DÉFIS

C'est Gene Amdahl qui nous explique dans la loi éponyme le premier défi : un logiciel parallèle est très vite limité par les parties non parallèles. En pratique imaginez que 20% de votre logiciel est sériel et 80% parallèle, vous pouvez supposer qu'il tirera bien parti d'une machine multi-core. Or si vous ajoutez une infinité de cœurs et que la partie parallèle de 80% en tire parfaitement bénéfice, votre logiciel n'ira que 5 fois plus vite. Pour 8 cœurs il n'ira que 3,3 fois plus vite. Il faut donc à tout prix réduire ces régions sérielles.

Le second problème de performance vient de la synchronisation entre threads dans la partie parallèle. Une variable, dont l'étendue est vaste et englobe la région parallèle, doit être protégée pour les écritures depuis différents threads, par exemple par une section critique. Cette section critique garantira qu'un seul thread accédera à cette variable à un moment donné, créant une mini région sérielle à décompter de la partie parallèle. L'idéal est que les threads puissent travailler indépendamment sans avoir à se synchroniser ou à s'attendre les uns les autres.

Le troisième problème vient de l'équilibre de charge entre threads. Pour que votre tâche s'exécute le plus vite possible vos threads doivent avoir la même charge de travail. Si vous en donnez plus à l'un qu'aux autres il finira plus tard et les autres attendront. Si votre charge est facile à prédire vous pouvez la scinder en parties égales dès le début et la répartir entre vos threads, si elle ne l'est pas, scindez la en petites parties et faites une queue où les threads iront piocher au cours du travail.

Le quatrième problème est le coût de la communication entre threads. Il est timide et ne se montre que lorsque vous avez traité les trois autres. Imaginez par exemple que, pour avoir une balance correcte entre threads, vous ayez scindé votre charge en une multitude de petites tâches et mis en place une queue de travail. Il faudra, certes, très peu de temps à chaque thread pour aller chercher un élément de la queue mais mis bout à bout cette communication est un poids. Tout est question de mesure : une granularité des éléments dans la queue assez fine pour avoir une bonne balance, mais pas trop fine pour minimiser les communications.

INTEL PARALLEL AMPLIFIER

Ces problèmes sont presque impossibles à isoler par une simple analyse de l'exécution ou en ajoutant des *printf* à la main. Intel Parallel Amplifier fait partie de la suite Intel Parallel Studio, son rôle est de vous aider à améliorer la performance de votre logiciel multithreadé. Il est important d'avoir déjà résolu les bugs parallèles de votre logiciel : une fois debuggué, votre logiciel ira probablement beaucoup moins vite et c'est justement sur ces ralentissements qu'Amplifier est utile. Amplifier propose trois analyses intégrées à Visual Studio : par *point chaud* " hot spot ", par *concurrence* et enfin par *blocages* et *attentes*. Il travaille d'un côté avec des compteurs matériels du processeur pour connaître les temps d'exécution avec une infime précision, et de l'autre avec la librairie de threading que vous utilisez. Aucune interaction avec le cadre de travail haut niveau, donc aucun problème de compatibilité que vous travailliez au niveau du thread, avec OpenMP ou Intel Threading Building Blocks, par exemple.

La première analyse est assez simple à comprendre, Amplifier exécute votre binaire et vous dit quelles sont les fonctions chronophages, mais rajoute aussi une colonne dans la vue du source pour vous montrer le détail ligne par ligne. Simple à dire, moins à programmer.

Cette analyse ne repose pas sur une simple instrumentation des appels de fonction mais sur un décompte matériel dans le processeur du temps d'exécution de chaque micro instruction. Amplifier fait ensuite le lien avec l'assembleur en mémoire puis le code. La précision du décompte au niveau du processeur est redoutable mais l'analyse n'impacte pas le temps d'exécution.

La seconde analyse, par concurrence, entre plus profondément dans le comportement parallèle de votre application. L'analyse vous renseigne tout d'abord sur le temps passé dans différents niveaux de concurrences, par exemple si vous exécutez sur un dual-core avec deux threads en exécution (l'idéal), ou un seul. Vous n'aurez qu'un seul thread en exécution si vous êtes dans une région sérielle ou bien potentiellement dans une section critique d'une région parallèle.

Cette donnée est affichée pour tout le logiciel, les fonctions et même pour chaque ligne de code. Non seulement il a fallu qu'Amplifier analyse le temps pris par chaque micro instruction en interagissant avec le décompte du processeur, mais qu'en plus il aligne les résultats de différents threads.

Au final, vous pouvez différencier les lignes parallèles des lignes sérielles.

La troisième et dernière analyse détecte les attentes et blocages (Wait and Locks). Il s'agit ici pour Amplifier d'instrumenter les communications entre threads pour détecter des attentes, comme dans le cas d'un défaut de balance, ou des blocages.

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.

L'INFORMATION SUR MESURE

L'INFORMATION EN CONTINU

www.solutions-logiciels.com

EXEMPLE

Pour commencer, il faut établir un temps de base en prenant une charge de travail assez importante pour que le temps soit significatif. La charge doit stresser différentes parties de votre code comme le feraient des charges de production, et de la même manière. C'est un point simple mais souvent sous-estimé. Lorsque vous calculez des temps, lancer une exécution pour chauffer la machine (au sens propre du terme, pour que la fréquence du processeur remonte) puis quelques exécutions pour faire une moyenne. Le cas du jour est un calcul de nombres premiers dans un intervalle d'entiers. Nous l'avons parallélisé avec OpenMP, puis debuggé avec Parallel Inspector en ajoutant deux sections critiques. Temps de base : 22,5s.

POINTS CHAUDS

Lançons l'analyse par point chaud (menu outils, sous-menu " Intel Parallel studio " et enfin " hot spots ") pour commencer. Le résultat est très simple : un tableau des fonctions (avec modules et arbre d'appel), classés par temps processeur. Cliquez sur une fonction et vous verrez le source avec le temps processeur pour chaque ligne. Astuce : si votre ligne est très longue et que vous voulez savoir quelle partie de la ligne est fautive, répartissez-là sur deux lignes, recompilez et relancez l'analyse.

Dans notre cas, c'est la fonction chargée d'indiquer l'avancement du calcul qui arrive en tête avec les 2/3 du temps. Je clique sur la fonction et le source correspondant m'indique que c'est surtout le printf. Après réflexion je remarque que le printf est demandé beaucoup plus souvent que nécessaire. Je change le code et calcule le temps : 5.26s (4.3x plus vite). Je relance la même analyse et la fonction d'avancement est presque invisible, cette fois c'est la fonction de calcul principale qui occupe l'essentiel du temps processeur.

CONCURRENCE

L'analyse de la concurrence affiche en bas à droite un graphique à barres qui indique pour différents niveaux de concurrence le temps passé. Je passe une partie importante du temps avec un seul thread en exécution ou même aucun. Inquiétant, car j'ai un dual-core. [Fig.1]

Je regarde le détail par fonction et vois réapparaître la fonction d'avancement. Elle n'accapare plus le processeur mais empêche les threads de s'exécuter en parallèle. Le détail du source est sans appel : la section critique qui protège l'incrémentation de *gProgress* est pointée. Il existe une fonction Windows qui protège l'incrémentation en parallèle d'une variable : " InterlockedIncrement ". Fonction très spécialisée et extrêmement optimisée. Je remplace donc mes deux sections critiques, recompile et calcule : 4,98s (6% plus vite). Le gain peut paraître faible mais la loi d'Amdahl nous enseigne qu'une région sérielle est extrêmement pénalisante pour un grand nombre de cœurs. J'ai en fait résolu un problème gravement handicapant sur une machine largement multi-core, mais peu visible sur mon dual-core. Après avoir relancé l'analyse par concurrence c'est la partie de calcul intensif qui est pointée. Ce n'est donc plus un problème de concurrence.

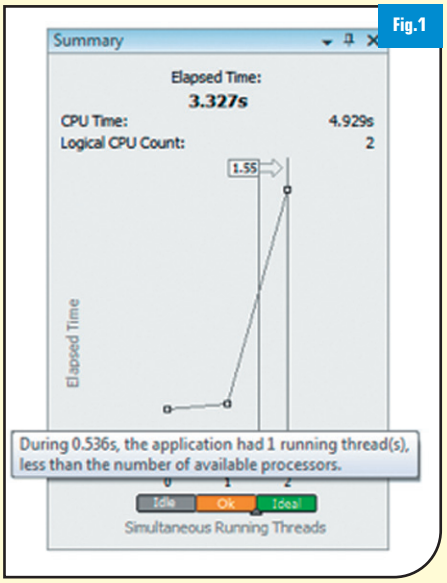


Fig.1

ATTENTES ET BLOCAGES

Il est maintenant temps de lancer la dernière (et plus amusante) des analyses : attentes et blocages. Le résultat est intuitif : je vois en bas à droite que je passe une grande partie de mon temps avec un seul thread, et dans la vue par fonction que tout cela est dû à la fonction qui lance le calcul parallèle. Le détail du source pointe la ligne de pragma OpenMP " parallel for ". J'ai probablement un problème de répartition de charge entre mes threads par OpenMP. [Fig.2]

L'instruction *parallel for* accepte des arguments. Pour commencer testons le cas simple : découpe de la charge en deux parties identiques avec " parallel for schedule (static,2) ". Résultat : 4.41s, bien, mais si je tente de perturber un peu la machine pendant l'exécution le temps augmente rapidement (un des threads travaille alors plus vite que l'autre). Avec (static,8) je suis plus robuste face aux perturbations et atteint un respectable 4.12s. Allons jusqu'à (static,2000000) : 4.42s, les deux threads finissent, quoi qu'il arrive, en même temps mais nous voyons le coût de la gestion de queue qui augmente. L'idéal se trouve probablement avec (guided) où la queue donne de gros morceaux au début et de petits à la fin, pour optimiser l'équilibre balance/communication, résultat : 4.09s. Le graphique nous indique que le logiciel s'exécute à 99% avec deux threads, le rêve ! Vous pouvez maintenant escompter une excellente utilisation d'un grand nombre de cœurs.

En pratique, il est préférable que votre logiciel accepte des paramètres différents (ou les calcule à l'exécution) pour s'adapter à des configurations matérielles différentes.

CONCLUSION

L'optimisation de logiciels parallèles est parfois complexe mais souvent passionnante. Un peu de méthode (pensez à Amdahl !) et le bon outil vous seront très utiles dans cette aventure. Intel Parallel Amplifier est tellement intuitif et précis que vous regretterez de n'avoir que votre logiciel à optimiser.

■ Paul **Guermonprez** - *Ingénieur logiciel Intel*

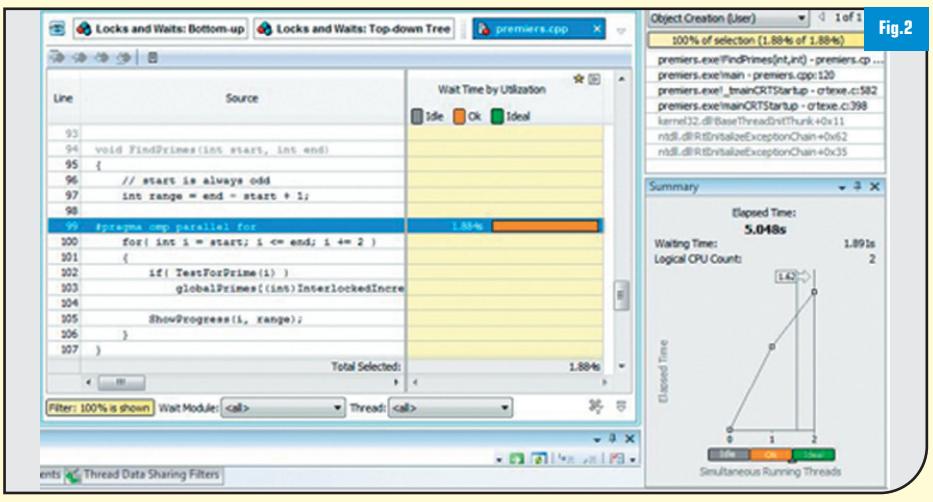
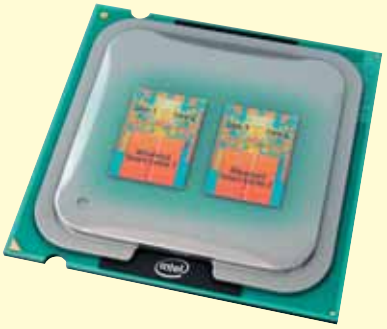


Fig.2

Vous ne reconnaîtrez plus votre C++



Intéressons-nous dans cet article à *Composer*, qui est le compilateur associé à diverses librairies comme Intel Threading Building Blocks et Intel Performance Primitives (TBB et IPP).

De nos jours, un compilateur ne peut se contenter de produire un binaire minimaliste, même optimisé. Il doit vous aider à adopter une syntaxe plus claire, apporter un éclairage nouveau sur votre code et surtout aider à paralléliser votre logiciel simplement. Un développeur n'est jamais trop exigeant avec son compilateur.

Une fois Parallel Studio installé, pour utiliser le compilateur Intel il suffit de cliquer droit sur votre solution et demander sa conversion (la procédure est réversible). Il est important de noter que le compilateur et les librairies sont disponibles sur d'autres plates-formes, et que les performances sont bonnes sur d'autres processeurs que ceux d'Intel, portabilité avant tout ! Parmi toutes les fonctionnalités de Parallel Composer, parlons donc des différentes manières de paralléliser votre code, des extensions C++ d'Intel et des fonctions lambda.

OPENMP

OpenMP est un standard qui permet de paralléliser rapidement un code avec un minimum de changements par rapport à la version sérielle. Le standard existe depuis plus de 10 ans, il a prouvé son utilité dans le calcul scientifique et bénéficie d'un soutien sans faille de l'industrie ; vous pouvez donc l'utiliser en production sans crainte. Historiquement OpenMP a commencé par l'ajout de pragmas simples, ces commentaires très spéciaux lus par certains compilateurs. Par exemple une ligne "#pragma omp parallel for" placée juste avant une boucle *for* permet de la paralléliser. Différentes itérations de la boucle seront alors envoyées à différents threads (créés automatiquement par OpenMP), et ce de manière totalement transparente et dynamique lors de l'exécution. Vous pouvez même spécifier quel algorithme de distribution vous souhaitez, tout cela sans changer une seule ligne de code. Vous pouvez certes programmer des threads en bas niveau avec C / C++, mais pourquoi ne pas laisser faire la machine virtuelle OpenMP ? Connaître OpenMP c'est l'adorer !

Si votre compilateur ne comprend pas OpenMP, votre code sera compilé en sériel, donc aucun risque. Pour appliquer les instructions, il doit savoir mettre en place un cadre de calcul parallèle dynamique lors de l'exécution autour de votre code et de vos données, toute la complexité est donc reportée du développeur vers le compilateur. Tâche d'autant plus complexe que le standard, au début limité à de simples fonctions, permet maintenant la parallélisation de cas beaucoup plus complexes. Le support d'OpenMP dans sa version récente 3.0 devient donc un critère important lors du choix d'un compilateur.

Dans sa mouture 3.0, OpenMP apporte notamment des améliorations dans la récursivité. Il n'est pas simple de paralléliser une liste de tâches dont on ne connaît pas l'étendue à l'avance, ou qui ont des dépendances entre elles, deux caractéristiques de bien des algorithmes récursifs. Le nouveau concept de "Task" mettra une partie de votre code dans la liste d'attente des tâches à calculer en parallèle (que vous n'avez pas à gérer), quels que soient le nombre et la complexité des tâches ou la nature récursive de leurs appels.

EXTENSIONS INTEL C++

Le concept d'OpenMP réside dans l'usage de pragmas, annotations certes intelligentes et transparentes de votre code mais pas très respectueuses d'un style d'écriture "naturel" en C++. C'est pour cela qu'Intel propose l'emploi de nouvelles extensions C++ : vous pouvez écrire "__par" au lieu de "#pragma omp parallel for". Si le concept vous plaît, n'hésitez pas à le dire sur les forums de software.intel.com et nous étendrons ce type de mots clés a l'avenir.

VECTORISATION ET OOO

Nous parlons souvent de parallélisation au niveau logiciel dans le but d'utiliser différents cœurs, mais la parallélisation peut aussi se faire au niveau matériel dans l'utilisation simultanée de différentes unités de calcul d'un même cœur. Soit vous utilisez ces unités pour effectuer des calculs différents et indépendants, et c'est que vous avez un processeur "Out Of Order" (sans ordre déterminé). Soit vous les utilisez pour des calculs similaires sur des données différentes,

nous parlons alors de vectorisation.
Le calcul Out Of Order est une propriété de certains processeurs (dont les Core2 d'Intel) ou tout est géré automatiquement et de manière transparente, le compilateur intervient peu dans ce mécanisme.
Par contre, la vectorisation nécessite l'emploi d'un jeu d'instructions spécifiques, les SSE (Streaming SIMD Extensions), LRBni (Larrabee New Instructions) ou futures Intel AVX (Advanced Vector eXtensions). Ces instructions sont généralement ajoutées par le compilateur si votre code le permet mais il vous faut lui en faire la demande par des arguments lors de la compilation (ces arguments peuvent être changés par des menus interactifs dans Visual Studio). Des exemples d'accélérations intéressantes concernent les opérations arithmétiques simples, ou l'emploi de "valarray".

DIAGNOSTIC

Par défaut, le compilateur Intel vous affichera un niveau de diagnostic comparable à celui de Visual Studio. Mais il est souvent très instructif de demander un affichage plus détaillé. Pour commencer, l'option "diag-enable:level1" pourra déceler des incohérences comme des arguments incorrects ou des tests logiques inutiles.
L'option "vec-report3" vous informera, elle, de la façon dont l'unité de vectorisation automatique du compilateur perçoit votre code, pas seulement le résultat de l'optimisation mais les raisons d'une non optimisation. Lorsque le compilateur ne peut déterminer avec certitude que deux données sont indépendantes, il suppose qu'elles sont dépendantes et empêche de vectoriser une boucle ou elles sont présentes. Bien souvent, de simples redéfinitions de variables temporaires permettent de prouver l'indépendance et de vectoriser. Et comme Parallel Composer est intégré à Visual Studio vous verrez les messages de log liés à votre code.

FONCTIONS LAMBDA

Si deux développeurs C++ se croisent ces derniers temps, il est probable que le sujet de discussion dérive rapidement vers les fonctions lambda. Probable aussi que la discussion soit quelque peu théorique, faute d'avoir un compilateur capable de les comprendre jusqu'à présent et une expérience pratique de ces fonctions.
Pour résumer, une fonction lambda est une fonction déclarée à la volée lors de l'appel et dont la syntaxe est simplifiée. Si vous n'avez besoin d'une fonction qu'une seule fois, pourquoi la déclarer complètement et ne pas se concentrer sur son code utile à la place ? Présentées sous cet angle les fonctions lambda peuvent apparaître comme un léger raccourci syntaxique sans conséquences. Elles sont en fait une révolution dans le monde du C++ : elles marquent le passage d'une programmation procédurale vers une programmation fonctionnelle (en simplifiant). Ce n'est peut être pas tout à fait un hasard si Intel s'y intéresse de si près. Le support des fonctions lambda est demandé par nombre de développeurs et Intel se doit donc de l'incorporer dans le compilateur, mais plus généralement, la programmation fonctionnelle devrait à long terme permettre de tirer parti au mieux d'une architecture largement parallèle, et ce avec un code simple et naturel. La programmation fonctionnelle n'entre pas en concurrence avec des technologies de calcul parallèle comme OpenMP ou TBB (qui utilisent parfois des principes de programmation fonctionnelle), les deux approches sont complémentaires.

IPP

Lorsque vous voulez optimiser l'utilisation matérielle de votre logiciel, la première action à prendre est souvent de remplacer vos librairies par des librairies optimisées, parallèles et "thread safe" (dont le comportement en mode parallèle est garanti par le vendeur). Parallel Composer inclut la librairie Intel Performance Primitives ou IPP qui peut être distribuée gratuitement avec votre logiciel, ne vous privez donc pas de l'utiliser. IPP offre un panel de fonctions très large : codecs vidéo, traitement d'image, reconnaissance faciale et vocale, compression, cryptographie, ray tracing ainsi que des fonctions plus bas niveau de traitement de signal.
Pour le traitement d'image par exemple, profitez du Deffered Mode Image Processing. Le DMIP vous propose de définir tous les filtres et opérations que vous souhaitez sur une image comme un pipeline avant de l'exécuter en optimisant l'utilisation du cache processeur.
Utiliser un codec vidéo IPP dans votre logiciel offre une performance et une portabilité optimale assez simplement. Par contre, utiliser certaines fonctions de traitement de signal pour faire de la reconnaissance faciale est plus ardu. (Mais si ce sujet vous intéresse, la librairie Open Computer Vision est très amusante, plus facile à utiliser et disponible sous licence BSD).
Si vous suivez l'actualité des processeurs de très près, vous avez peut être vu que le Core i7 "Nehalem" est sorti depuis peu et qu'il intègre un nouveau jeu d'instructions "STTNI" spécialisé dans le traitement de chaînes. Si vous travaillez dans la fouille de données profitez en dès maintenant avec des fonctions IPP comme "ippsFind*Any()".
N'hésitez pas à regarder la documentation d'IPP, vous trouverez probablement des idées pour accélérer ou améliorer des parties de votre logiciel.

ET BIEN PLUS

Le compilateur Intel est déjà très complet et utile dans sa configuration actuelle avec OpenMP, IPP et TBB mais il le sera encore plus à l'avenir. Lorsque vous souhaiterez utiliser les ressources d'un accélérateur graphique comme Larrabee de manière transparente pour des simulations physiques ou une intelligence artificielle, il est fort possible que le compilateur joue un rôle central.
Si vous voulez programmer en parallèle, en fonctionnel ou tout simplement créer des binaires plus rapides je vous encourage donc à tester Intel Parallel Composer et à donner votre avis sur les forums de software.intel.com.

■ Paul **Guermonprez**
Ingénieur logiciel Intel

OFFRE LIMITÉE

SAVOIR c'est POUVOIR !

EN CADEAU
Accès illimité de 10 jours à la Bibliothèque Numérique
des Editions ENI

A choisir parmi les Bouquets :
- **Développement :** Plus de 30 livres numériques disponibles, de ADO.Net 3.5 à Flex
- **Microsoft :** Plus de 30 livres numériques disponibles, de Windows Server 2008 à Silverlight
- **Open Source :** Plus de 20 livres numériques disponibles, de Nagios à Java EE

Pour tout abonnement au magazine Programmez un accès de 10 jours à la Bibliothèque Numérique de votre choix vous est offert. *Consultez :* ***www.eni-bibliotheque.fr***

Offre limitée aux 1000 premiers souscripteurs.

*11 numéros par an : 49 €**

Economisez 16,45 €*

** au lieu de 65,45€ prix de vente au numéro, Tarif France métropolitaine*

Choisissez votre formule
Inclus : accès à la Bibliothèque Numérique !

- **Abonnement 1 an au magazine : 49 €**
  (au lieu de 65,45 € tarif au numéro) *Tarif France métropolitaine*
- **Abonnement Intégral : 1 an au magazine + archives sur Internet et PDF : 59 €** *Tarif France métropolitaine*
- **Abonnement PDF / 1 an : 30 €** *- Tarif unique*
  Inscription et paiement exclusivement en ligne
  www.programmez.com
- **Abonnement Etudiant : 1 an au magazine : 39 €**
  (au lieu de 65,45 € tarif au numéro) *Offre France métropolitaine*

+ Abonnement INTÉGRAL

ACCÈS ILLIMITÉ aux ARCHIVES du MAGAZINE pour 0,84€ par mois !

Cette option est réservée aux abonnés pour 1 an au magazine, quel que soit le type d'abonnement (Standard, Numérique, Etudiant). Le prix de leur abonnement normal est majoré de 10 € (prix identique pour toutes zones géographiques). Pendant la durée de leur abonnement, ils ont ainsi accès, en supplément, à tous les anciens numéros et articles/dossiers parus.

OUI, *je m'abonne* ***Vous pouvez vous abonner en ligne et trouver tous les tarifs*** www.programmez.com

PROGRAMMEZ *avec Accès 10 jours à la Bibliothèque Numérique des Éditions ENI ! (offre limitée)*

☐ **Abonnement 1 an au magazine : 49 €** (au lieu de 65,45 € tarif au numéro) *Tarif France métropolitaine*

☐ **Abonnement Intégral : 1 an au magazine + archives sur Internet et PDF : 59 €** *Tarif France métropolitaine*

☐ **Abonnement Etudiant : 1 an au magazine : 39 €** (joindre copie carte étudiant) *Offre France métropolitaine*

Choisissez votre Bouquet de la Bibliothèque Numérique : ☐ *Développement* ☐ *Microsoft* ☐ *Open Source* **(Attention, e-mail indispensable)**

☐ M. ☐ Mme ☐ Mlle Entreprise : Fonction :

Prénom : Nom :

Adresse :

Code postal : Ville :

Tél :

E-mail : @

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

A remplir et retourner sous enveloppe affranchie à : Programmez ! - Service Abonnements - 22 rue René Boulanger - 75472 Paris Cedex 10.
abonnements.programmez@groupe-gli.com

Offre limitée, valable jusqu'au 30 août 2009

Le renvoi du présent bulletin implique pour le souscripteur l'acceptation pleine et entière de toutes les conditions de vente de cette offre.

Conformément à la loi Informatique et Libertés du 05/01/78, vous disposez d'un droit d'accès et de rectification aux données vous concernant. Par notre intermédiaire, vous pouvez être amené à recevoir des propositions d'autres sociétés ou associations. Si vous ne le souhaitez pas, il vous suffit de nous écrire en nous précisant toutes vos coordonnées.

PROG 121

PROgrammez! Le magazine du développement

PLATE.
RONDE.

FAITES EVOLUER VOTRE CODE.

L'innovation en parallèlisme.

Donnez à vos logiciels les outils pour être compétitifs. Avec Intel® Parallel Studio vous n'avez pas besoin d'être un expert en parallèlisme pour optimiser vos applications pour le multicore. Fait pour les applications sérielles d'aujourd'hui et les innovations parallèles de demain.

Testez gratuitement: **www.intel.com/software/products/eval**

intel Software Products

© 2009-2010, Intel Corporation. Tous droits réservés. Intel et le logo Intel sont des marques d'Intel Corporation, déposées ou enregistrées aux États-Unis et dans d'autres pays. *Les autres noms et désignations peuvent être revendiqués comme marques par des tiers.

iPhone : le téléphone devient enfin intelligent et connecté

Apple ne met plus en avant les qualités de son téléphone mais communique exclusivement sur les applications. C'est ce qui fait le succès de l'iPhone. Comment développer son application iPhone ? Comment la déployer ? Toutes les réponses ici.

LES PRÉ-REQUIS

• L'iPhone utilise comme système d'exploitation une version allégée de Mac OS X et Apple a construit l'ensemble du kit de développement autour de sa plate-forme de développement existante : ***XCode*** pour l'éditeur de code, ***Interface Builder*** (ici nommé IB) pour la création d'interfaces en WYSIWYG, ***Cocoa Touch*** une surcouche de *Cocoa* pour les API et enfin ***Objective-C*** pour le langage. Le SDK est téléchargeable gratuitement et le site http://developer.apple.com/iphone regorge de documentation, de vidéos et d'exemples. Même un parfait novice peut commencer à développer pour peu qu'il soit motivé. Attention : Mac obligatoire.

• Le SDK fournit un simulateur d'iPhone, mais ce dernier est assez limité en termes de fonctionnalités. Par exemple, il ne permet pas de faire de la 3D ou de recevoir les évènements de l'accéléromètre. Un iPhone est par conséquent obligatoire, d'autant plus que certains bugs peuvent n'apparaître que sur l'appareil.

• Abordons la partie qui fâche : pour développer sur iPhone, il faut payer. Avant de pouvoir, ne serait-ce que tester une application sur son propre iPhone, il est nécessaire de souscrire à l'« *iPhone Developer Program* » pour la modique somme de 99$ par an. Ce programme donne accès à un portail Web où il est possible de générer un certificat permettant de signer une application et ainsi de la déployer.

OUVRONS LE CAPOT

Nous allons développer une « boîte à meuh », ce n'est pas novateur mais nécessite plusieurs des API de l'iPhone, à savoir les animations, l'audio et l'accéléromètre. Ce chapitre présente à la fois les premiers outils du SDK et les grands concepts du développement iPhone, les plus pressés pourront se rendre au chapitre suivant où la partie code démarre réellement.

Créons un nouveau projet « *Moo* » dans XCode avec le modèle « *View-Based Application* ». XCode se charge de créer la structure minimale du projet : [Fig.1]

L'éternel main

Le point de départ d'une application est le fichier ***main.m*** :

```
#import <UIKit/UIKit.h>

int main(int argc, char *argv[]) {

    NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];
    int retVal = UIApplicationMain(argc, argv, nil, nil);
    [pool release];
    return retVal;
}
```

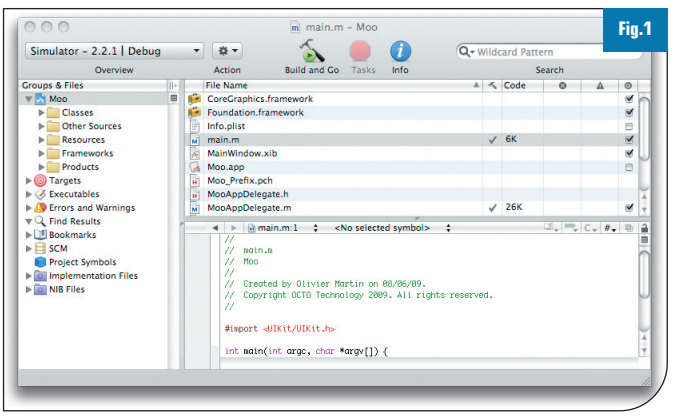
Notre application iPhone est lancée par l'invocation de la fonction *UIApplicationMain,* elle fait partie de *UIKit,* un des frameworks composant *Cocoa Touch.* Elle est précédée par la création d'un pool mémoire *NSAutoreleasePool,* qui sera détruit une fois que la fonction *UIApplicationMain* aura rendu la main, c'est-à-dire une fois que le programme aura quitté. On peut donc déjà voir ici que la gestion de la mémoire est entièrement à la charge du développeur et c'est le cas pour toutes les applications Objective-C. Il est nécessaire d'être très rigoureux pour ne pas risquer de fuites mémoire, mais en contrepartie on obtient de meilleures performances.

InterfaceBuilder entre en scène

Une fois cette méthode lancée, elle va charger le fichier *Info.plist.* Il s'agit d'un ensemble de clés-valeur au format XML, définissant différentes propriétés de l'application comme son nom, sa version ou bien encore l'icône à afficher dans *SpringBoard.* L'une de ces propriétés nous intéresse ici :

```
<key>NSMainNibFile</key>
<string>MainWindow</string>
```

C'est cette clé qui fait le lien avec *InterfaceBuilder (IB)*, elle spécifie le nom de l'interface à charger. On trouve donc assez logiquement le fichier MainWindow.xib dans les ressources de notre projet. Double-cliquons dessus pour lancer *InterfaceBuilder.* [Fig.2]



IB permet de créer une interface graphique par drag&drop de composants, dont plusieurs catégories sont dédiées à l'iPhone. Son côté intéressant est qu'il est complètement dissocié du code : il permet de créer une définition d'interface stockée dans un fichier XML (.xib) qui au moment de la compilation est transformée en un graphe d'objets sérialisés remontés en mémoire au chargement de l'application. Pour pouvoir interagir avec ces composants, on branche la référence d'un composant graphique à un pointeur dans le code, ou bien un évènement généré par l'interface à une fonction. De cette façon les deux mondes code et interface sont complètement dissociés et peuvent évoluer indépendamment. La hiérarchie d'objets déjà présente va nous permettre d'expliquer quelques concepts de programmation iPhone :

- **File's Owner :** c'est celui qui « détient » ce fichier xib et n'est pas créé à son chargement, dans notre cas il s'agit du singleton *UIApplication*.
- **First Responder :** le premier objet à être invoqué lorsqu'un événement est déclenché.
- **MooAppDelegate :** cette classe fait partie de notre projet et elle implémente *UIApplicationDelegate*, c'est-à-dire qu'elle peut recevoir certains messages envoyés à l'application. Il s'agit d'un design pattern très utilisé dans Cocoa : la délégation permet à un objet de recevoir des messages spécifiés par un protocole, ici *UIApplicationDelegate*, il s'agit d'une alternative à l'héritage de classe pour faire de la réutilisation.
- **MooViewController :** un autre design pattern, plus classique MVC. Détail intéressant, sa vue est définie dans un autre fichier xib, ceci permet de découper l'interface en autant de fichiers que nécessaires qui peuvent être chargés tardivement et améliore le travail collaboratif sur ces fichiers.
- **Window :** la fenêtre principale.

Retour au code

L'interface graphique va donc être chargée en mémoire, et La méthode applicationDidFinishLaunching est ensuite invoquée sur la classe MooAppDelegate avec la variable window pointant sur le bon objet en mémoire, comme par magie.

```
#import "MooAppDelegate.h"
#import "MooViewController.h"

@implementation MooAppDelegate

@synthesize window;
@synthesize viewController;


- (void)applicationDidFinishLaunching:(UIApplication *)application {

    // Override point for customization after app launch
    [window addSubview:viewController.view];
    [window makeKeyAndVisible];
}

- (void)dealloc {
    [viewController release];
    [window release];
    [super dealloc];
}

@end
```

DU CODE, DU CODE !

L'interface graphique

Pour commencer nous allons faire en sorte d'afficher une image, qui jouera un son lorsqu'elle sera cliquée, il faudrait plutôt dire « touchée » puisqu'il n'y a pas de souris, vous me pardonnerez cet abus de langage. Dans XCode, on commence par rajouter une image dans les ressources du projet : [Fig.3] Editons maintenant notre fichier *MooViewController.xib* dans IB où l'on rajoute un nouveau bouton (Round Rect Button). On aurait pu simplement rajouter une ImageView mais les boutons permettent plus simplement de recevoir des évènements utilisateur. On spécifie ensuite le type du bouton à *Custom* et son arrière plan à l'image que nous avions rajouté plus haut. [Fig.4] Pour que notre code puisse être invoqué lorsque la vache est cliquée, il faut maintenant faire le lien entre XCode et IB. Nous allons donc rajouter la fonction suivante dans XCode :

MooViewController.h :

```
-(IBAction)cowClicked:(id)sender;
```

MooViewController.m :

```
-(IBAction)cowClicked:(id)sender {
 NSLog(@"Cow clicked");
}
```

Compilez le projet et repassez dans IB, sélectionnez le « File's Owner » qui est notre MooViewController et dans la fenêtre d'in-

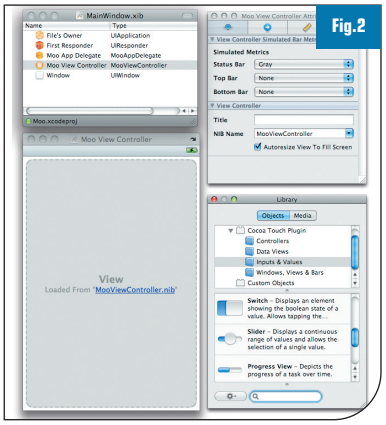


Fig.2

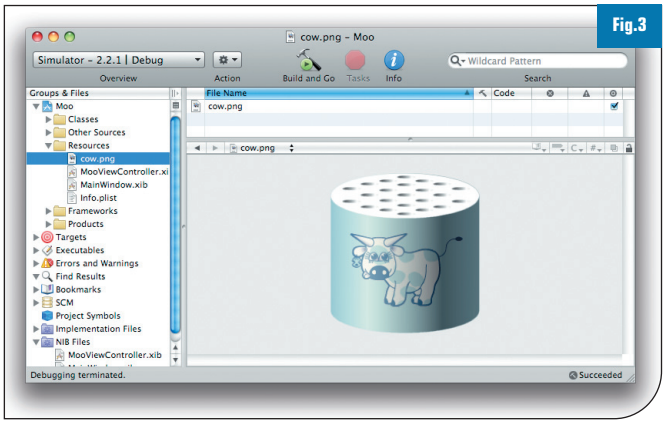


Fig.3

Fig.4

formation, ouvrez le deuxième onglet qui correspond aux connexions de l'objet. Vous pouvez voir maintenant l'action « *cowClicked* » que nous venons de rajouter. Branchons maintenant notre bouton à cette action, pour cela il suffit de faire glisser le rond en face de l'action sur notre bouton. Une fenêtre apparaît alors affichant les différentes actions disponibles : [Fig.5]
On choisit ensuite « *Touch Up Inside* » qui correspond à un clic sur le bouton. Notre programme va pouvoir maintenant recevoir ce type d'événement.

Animons tout ça

CoreAnimation est une API bas niveau permettant d'appliquer des matrices de transformations sur un layer. Apple a simplifié son utilisation dans le SDK iPhone en rajoutant des appels directement sur la classe *UIView* (la classe parente des contrôles graphiques).
Modifions donc notre *callback cowClicked* de manière à effectuer une première rotation de 180° et ensuite à revenir à la position initiale. Cette action sera lancée sur la vue de notre bouton qui doit d'abord être déclarée au niveau de la classe *MooViewController* et branchée dans IB.

MooViewController.h :

```
@interface MooViewController : UIViewController {
  UIButton *cowButton;
}
@property (nonatomic, retain) IBOutlet UIButton *cowButton;
...
```

MooViewController.m :

```
@implementation MooViewController

@synthesize cowButton;

...

-(IBAction)cowClicked:(id)sender {

  NSLog(@"Cow clicked");

  [UIView beginAnimations:nil context:NULL];
  [UIView setAnimationDuration:0.3];
  [UIView setAnimationBeginsFromCurrentState:YES];
  [UIView setAnimationDelegate:self];
  [UIView setAnimationCurve:UIViewAnimationCurveEaseIn];
  [UIView setAnimationDidStopSelector:@selector(endCowAnimation)];
  // Callback de fin
  cowButton.transform = CGAffineTransformRotate(cowButton.trans
form, M_PI); // Rotation de 180 degrés
  [UIView commitAnimations];
}


-(void)endCowAnimation {

  [UIView beginAnimations:nil context:NULL];
  [UIView setAnimationDuration:1];
  [UIView setAnimationBeginsFromCurrentState:YES];
  [UIView setAnimationCurve:UIViewAnimationCurveEaseOut];
  cowButton.transform = CGAffineTransformRotate(cowButton.trans
form, -M_PI); // Rotation inverse
  [UIView commitAnimations];
}
```

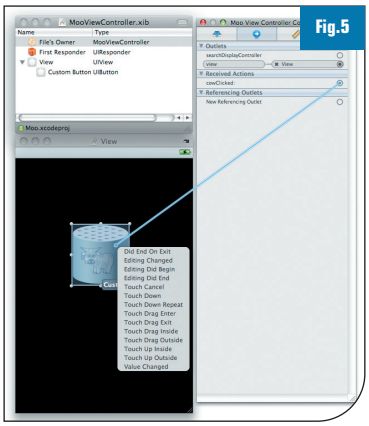


Fig.5

Fig.6

Un bloc animation se déclare de la manière suivante :
- L'initialisation avec l'appel [UIView beginAnimations]
- La possibilité de paramétrer cette animation : durée, vélocité (vitesse constante, accélération sur la fin et/ou début), callback appelé à la fin de l'animation permettant d'enchaîner plusieurs transformations, etc.
- La ou les transformations que l'on applique aux vues via leur propriété transform (elles sont appliquées par rapport à la position initiale)
- La fin du bloc et le lancement de l'animation à l'appel [UIView commitAnimations]

Ce qui donne le résultat obtenu [Fig.6].

Du son

Il ne reste plus maintenant qu'à rajouter un son pour avoir une première version de notre exemple. Tout d'abord on rajoute un fichier son *moo.wav* dans XCode par simple drag & drop. Dans la section *Frameworks* on pointe vers une nouvelle bibliothèque gérant ce genre de fonctionnalités : *AudioToolbox*.
On modifie ensuite le fichier MooViewController.h pour rajouter l'import approprié et une référence vers un objet *SystemSoundID* qui représente une ressource sonore.

MooViewController.h :

```
...
#include <AudioToolbox/AudioToolbox.h>

@interface MooViewController : UIViewController {
  ...
  SystemSoundID mooSound;
}
...
```

Dans l'implémentation MooAppDelegate.m, on initialise cette ressource sonore et on la lance dans notre méthode *cowClicked* :

MooViewController.m :

```
...
- (void)viewDidLoad {
```

```
    // Initialize the Moo sound
   id soundPath=[[NSBundle mainBundle] pathForResource:@"moo"
 ofType:@"wav"];
    CFURLRef baseURL=(CFURLRef) [[NSURL alloc] initFileURL
WithPath:soundPath];
    AudioServicesCreateSystemSoundID(baseURL, &mooSound);
    CFRelease(baseURL);

}

- (void)dealloc {
 AudioServicesDisposeSystemSoundID(mooSound);
    [super dealloc];
}

-(IBAction)cowClicked:(id)sender {
 ...
 AudioServicesPlayAlertSound(mooSound);
 ...
}
```

La touche finale : l'accéléromètre

Notre vache est maintenant capable de se retourner et d'émettre un son délicat lorsqu'on la clique, mais un exemple de boîte à meuh ne serait pas complet sans pousser jusqu'à l'une des fonctionnalités les plus fun de l'iPhone : l'accéléromètre.

Il existe un objet global [UIAccelerometer sharedAccelerometer] envoyant périodiquement des évènements au travers d'une méthode contenant en paramètre l'accélération sur les 3 axes x, y et z. Il nous suffira de répondre au protocole *UIAccelerometerDelegate* et de spécifier à l'accéléromètre que notre contrôleur est son delegate. Le comportement d'une boite à meuh est de sonner une fois qu'on l'a retournée puis remise à l'endroit, nous allons donc le détecter et déclencher notre méthode *cowClicked* comme ci-dessous :

```
#define kUpdateFrequency 20  // Hz
#define kFilteringFactor 0.05
BOOL upsideDown = NO;
float accelX = 0;
float accelY = 0;

- (void)viewDidLoad {
     ...
 // Request events from the accelerometer
   [[UIAccelerometer sharedAccelerometer] setUpdateInterval
:(1.0 / kUpdateFrequency)];
    [[UIAccelerometer sharedAccelerometer] setDelegate:self];
}

- (void)accelerometer:(UIAccelerometer *)accelerometer didAc
celerate:(UIAcceleration *)acceleration {

    // Filter events with a low pass filter
    accelX = acceleration.x * kFilteringFactor + accelX *
(1.0 - kFilteringFactor);
    accelY = acceleration.y * kFilteringFactor + accelY *
(1.0 - kFilteringFactor);
    float angle = 180 * atan2(accelY, accelX) / M_PI;

   NSLog(@"angle : %f, (accels x / y): %f, %f", angle, accel
X, accelY);

    if (angle > 10) {
        upsideDown = YES;
    }
    else if (angle < -10 && upsideDown == YES) {
        upsideDown = NO;
        [self cowClicked:nil];
    }
}
...
```

OPTIMISATION ET DISTRIBUTION

Les outils d'aide au diagnostic

La phase d'optimisation n'est pas à sous-estimer, rappelez-vous que la gestion de la mémoire est à la charge du développeur et une fuite mémoire est un motif suffisant pour être refusé sur l'AppStore. On dispose de :

- I**nstruments** permet de collecter des métriques sur les ressources de l'iPhone : CPU, consommation mémoire, etc.
- **Shark** permet d'avoir une vue sur ce qui se passe au niveau système : threads, interrupts, appels système.

Ad-Hoc, In-House et AppStore

La distribution d'applications passe par l'utilisation d'*iTunes* qui permet de synchroniser les iPhone ou bien encore de déployer les certificats et applications, *Program Portal* un portail web pour gérer ses applications et enfin *iTunes Connect* un autre portail dédié à l'AppStore.

Avant tout il existe un mode de distribution nommé « **ad-hoc** » qui permet de distribuer directement son application à 100 personnes maximum dans le but de réaliser du bêta testing. *Program Portal* permet d'associer un identifiant iPhone à une application et génère un certificat à installer sur l'appareil en question avec l'application via *iTunes*.

« **In-house** » est un mode de distribution réservé aux grandes entreprises qui peuvent par ce biais gérer complètement la distribution d'applications maison.

Et enfin, **l'AppStore** est la plate-forme de distribution grand public gérée entièrement par Apple. N'importe qui peut ainsi bénéficier de toute la puissance marketing d'Apple et d'un canal de vente gratuit. Une fois l'application prête, elle doit être soumise par le biais d'*iTunes Connect* et ce sont les employés d'Apple qui décideront si oui ou non elle peut rejoindre l'AppStore et surtout quand. Ce processus peut se révéler très frustrant, les causes de refus étant multiples : ergonomie en désaccord avec les principes d'Apple, mauvaises performances ou tout simplement jugée de mauvais goût, les exemples sont légions sur le net.

■ Olivier **Martin**
Expert senior au sein d'OCTO Technology, cabinet d'architectes en systèmes d'information
(http://blog.octo.com, http://universite-du-si.com)

Développez de A à Z sur mobile, avec Android

Android est le dernier arrivant sur le marché des systèmes mobiles. Dans cet article, nous allons nous intéresser au développement d'une application pour Android – sans nous y attarder – en détaillant les différentes étapes de la configuration de l'environnement de développement jusqu'à la publication sur le Market Android.

Le virus Android commence tout doucement à saisir la France avec l'arrivée – malheureusement tardive par rapport à d'autres pays – des premiers appareils Android sur le marché. Vous trouvez l'Android Market, le « App Store » Android, pas suffisamment étoffés à votre goût ? Vous voudriez les compléter avec votre application ? Cet article est fait pour vous ... développeur Java.

Installer et configurer votre environnement de développement

Pour la majorité des développeurs Java, l'utilisation d'un IDE comme Eclipse est aujourd'hui devenue monnaie courante. C'est pourquoi Google s'appuie sur cet IDE gratuit pour proposer son ensemble d'outils de développements et de tests, que vous retrouverez sous la forme d'un plugin pour Eclipse nommé ADT – Android Development Toolkit – dans sa version 0.9.

Avant toute chose, il faut vous procurer la dernière version du SDK 1.5 sur le site de Google à cette adresse : http://developer.android.com/sdk/1.5_r1/index.html. Décompressez l'archive dans un répertoire nommé 'android' et localisé - par exemple - à la racine de votre lecteur c. Félicitations, vous venez d'installer le SDK Android ! Celui-ci contient tous les outils nécessaires à la réalisation d'une application : compilateur / outil de package .apk, débogage, émulateur, exemples d'applications, documentation, etc. Néanmoins, si vous décidez de vous mettre au développement Android vous ne pourrez pas vous passer d'un IDE Java tel qu' Eclipse. La prochaine étape est donc de vous rendre sur le site de Eclipse (www.eclipse.org) et de télécharger la dernière version – quelle que soit l'édition – d'Eclipse Ganymède. Une fois téléchargée, suivez les instructions pour 'installer' l'IDE puis lancez-le. Notez que vous devrez également avoir la dernière version du JDK pour pouvoir exécuter ce dernier. Une fois dans Eclipse, vous allez utiliser le système de mise à jour pour ajouter le module ADT. Pour cela, suivez les étapes :

- Allez dans **Help > Software Updates ...**
- Cliquez sur **Add Site ...**
- Entrez l'adresse suivante : http://dl-ssl.google.com/android/eclipse/ puis cliquez sur **OK**
- Revenez dans l'onglet **Available Software**
- Sélectionnez la case à cocher **Developer Tools** qui apparaît, en cliquant sur la croix correspondant à l'url ajoutée. Puis cliquez sur **Finish** pour installer le plugin et redémarrez Eclipse pour finaliser l'installation.
- Après redémarrage de l'IDE, vous allez devoir configurer le plugin pour spécifier l'emplacement du SDK Android : allez dans le menu **Windows > Preferences...** et choisissez **Android** dans le menu de gauche puis dans la partie de droite cliquez sur **Browse...** pour sélectionner l'emplacement où vous aviez décompressé le SDK Android.

Vous voici fin prêt pour développer votre première application. Sachez que vous pouvez également utiliser Ant pour compiler votre projet, cependant ce sujet sort du cadre de cet article. [Fig.1]

Votre première application avec ADT

L'objectif de cette partie n'est pas de réaliser une application complète – vous pouvez retrouver l'article Programmez de Mai 2009 sur la création d'une application complète - mais de vous montrer comment vous servir d'Eclipse afin de parvenir à vos fins. La première chose que vous pourrez constater après l'installation d'ADT est la présence en haut à droite des perspectives qui vous seront utiles (par défaut seule Java est disponible, ajoutez-les en cliquant sur l'icône '+' de gauche) : Java pour avoir un ensemble de vues orienté pour éditer vos fichiers Java, Debug afin de proposer un ensemble de vues de débogage et enfin DDMS que nous découvrirons en détail un peu plus bas (seule cette dernière est propre à ADT). [Fig.2].

Un autre changement est la présence d'un nouveau type de projet Android. Cliquez sur **File > New > Project ...** pour afficher la

Fig.1

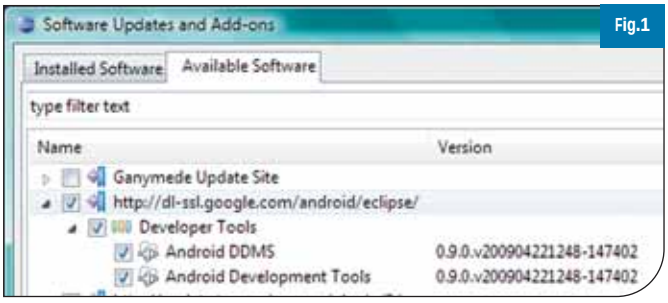
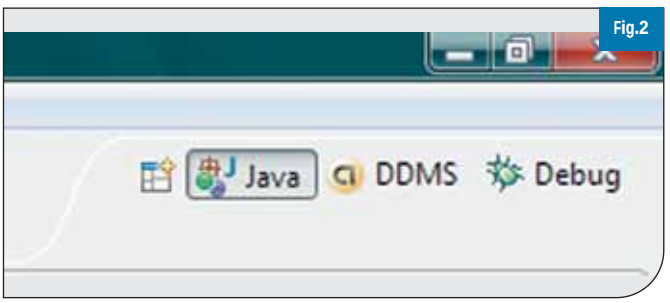


Fig.2



fenêtre de sélection de nouveau projet et choisissez **Android.** [Fig.3]
La fenêtre de l'assistant de création de projet Android vous propose plusieurs options dont le choix de la plate-forme supportée Android 1.1 – version 2 des API – ou Android 1.5 – version 3 des API – et la possibilité de créer une activité de départ. Remplissez les autres champs à votre convenance afin de spécifier un espace de nom, le nom de l'application (que vous pourrez modifier plus tard) et enfin le nom du projet. Une fois toutes les valeurs spécifiées, cliquez sur **Finish** pour générer votre première application Android. [Fig.4]
Le projet généré par Eclipse possède une certaine structure sur laquelle nous devons nous attarder quelques instants. Cette hiérarchie de fichiers/ répertoires permet au plugin ADT de vous aider dans votre développement sur Android en générant une classe R, cependant cela sort du cadre du présent article. Voici la structure telle que proposée par Eclipse après création de votre projet : [Fig.5]
Voici le détail des éléments les plus importants :

- Le dossier **src** est le dossier dans lequel vous allez retrouver toutes les sources des classes Java sur lesquelles vous allez travailler,
- Le dossier **gen** est le dossier dans lequel ADT génère les classes, celle qui nous intéressera le plus est la classe R,
- Le dossier **assets** permet de placer des ressources (son audio mp3, etc.) qui ne seront pas traitées par ADT et qui seront insérées en brut dans le fichier final .apk de l'application,
- Le dossier **res** contenant l'ensemble des ressources du projet (icônes, images, structure des layout, des valeurs, etc.) qui sert de base à ADT pour générer la classe R qui vous aidera à travailler avec l'ensemble de vos ressources,
- Un fichier **AndroidManifest.xml** qui représente le manifest de déploiement de l'application dans lequel vous allez notamment déclarer vos activités – assimilé à un écran de votre application – et les filtres des intentions (pour savoir à quoi vos activités réagissent),
- Et pour finir, un fichier **default.properties** qui est propre à la configuration de ADT dans votre projet.

Vous pouvez d'ores et déjà compiler votre application mais l'émulateur n'est pas encore configuré pour exécuter l'application.

Configurer son émulateur pour exécuter, tester et déboguer votre application

Cette nouvelle version du SDK apporte les AVD (Android Virtual Device) – peut être assimilée à une configuration de l'émulateur - que vous devrez créer avant toute chose afin de pouvoir exécuter votre application. Voici les étapes de création d'AVD :

- Lancez une console de ligne de commande et naviguez dans le répertoire **tools** de votre SDK Android,
- Avant de sélectionner une cible de déploiement, exécutez la commande suivante afin de visualiser toutes les cibles disponibles : ***android list targets***
- Dans la liste qui s'affiche, retenez l'identifiant de la version de la plate-forme Android sur laquelle vous souhaitez développer (ici 2 par exemple) :

```
id:1
     Name: Android 1.1
     Type: platform
     API level: 2
     Skins: HVGA (default), HVGA-L, HVGA-P, QVGA-L, QVGA-P
id:2
     Name: Android 1.5
     Type: platform
     API level: 3
     Skins: HVGA (default), HVGA-L, HVGA-P, QVGA-L, QVGA-P
```

- Exécutez la commande suivante pour créer votre AVD : **android create avd –name <nom de l'AVD> –target <ID de la cible>**
- Ensuite, vous pourrez si vous le désirez, créer un profil personnalisé de votre émulateur. Pour cette fois, saisissez simplement *no*, ou appuyez sur la touche entrée sans saisie, pour accepter la configuration matérielle par défaut.

Une fois votre AVD créé, vous allez devoir spécifier à Eclipse que vous souhaitez l'utiliser. Retournez dans Eclipse et cliquez sur **Run** puis choisissez le type d'exécution Android. Si vous n'avez pas encore configuré de cible pour l'exécution de votre application, la fenêtre suivante s'ouvrira en vous demandant de choisir la configuration que vous souhaitez utiliser (vos AVD en fonction du niveau du SDK s'afficheront dans cette fenêtre). [Fig.6]
Une fois votre AVD sélectionné, cliquez sur Run, l'émulateur Android démarre et charge votre application Android. Ne fermez pas l'émulateur à chaque fin d'exécution ! Votre application s'installera à nouveau chaque fois que vous voudrez l'exécuter depuis Eclipse et vous gagnerez du temps pour la tester. [Fig.7]
Voilà, vous avez configuré votre environnement de développement et exécuté votre première application Android. Cependant, rares sont les personnes pouvant se targuer de développer des applications parfaites et optimisées du premier coup, voici donc venu le temps de tester votre application.

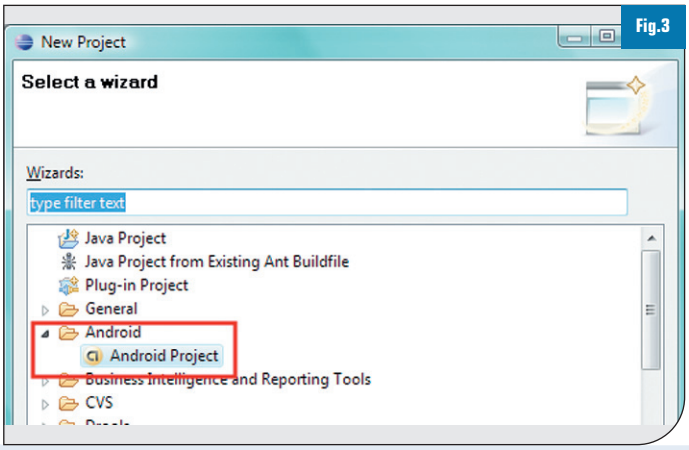


Fig.3

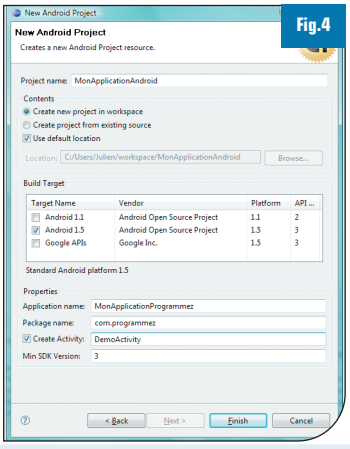


Fig.4

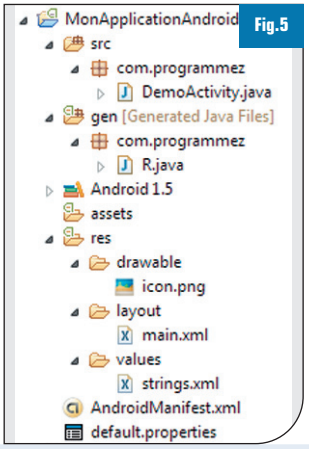


Fig.5

Tester et déboguer votre application

Tester votre application n'est pas seulement un gage de qualité, c'est bien souvent l'occasion de savoir aussi si votre application se lance et de regarder comment votre code s'exécute dans la Dalvik VM (la VM d'Android qui exécute votre application). Pour placer des points d'arrêt dans votre code, cliquez sur la ligne désirée dans la marge de la fenêtre d'édition de code (ou Ctrl + Shift + B). Lancez à nouveau votre application en débogage en cliquant sur le bouton - identifié par un insecte - à gauche de l'icône d'exécution. Si la perspective Debug n'est pas ouverte, Eclipse vous demandera de l'ouvrir pour vous. [Fig.8]

Utilisez les touches F5, F6 et F7 pour avancer pas à pas dans votre application une fois le point d'arrêt atteint. Déboguer une application est essentiel mais savoir comment celle-ci se comporte en termes de gestion de thread et de mémoire l'est tout autant ; surtout pour du développement d'appareils à ressources limitées comme des téléphones. C'est la perspective DDMS, que vous pouvez sélectionner en haut à gauche de l'IDE, qui vous servira pour remplir ces tâches. Cet ensemble de vues permet de s'attacher à un processus de l'émulateur pour le déboguer, connaître l'activité de ses threads et la gestion de la mémoire.

Vous pourrez aussi à partir de cette interface, simuler un appel entrant, un SMS, le GPS, etc. Vous pourrez également prendre une capture de l'affichage de l'émulateur et explorer le contenu du disque système de l'émulateur et des autres périphériques. [Fig.9]

Développement sur appareil physique

Développer sur un émulateur permet de tester son application de façon générale. Cependant, chaque appareil possède des spécificités : quantité de mémoire disponible, puissance du processeur, taille de l'écran, position des boutons, … et les implémentations d'Android par les constructeurs par rapport à l'émulateur peuvent être légèrement différentes.

Par conséquent, avant de publier une application, vous devrez toujours vous assurer que celle-ci est pleinement fonctionnelle sur un appareil physique tel que le G1 ou le G2 (seuls appareils disponibles à l'heure de l'écriture de cet article).

Pour pouvoir lancer une application sur votre appareil physique, branchez-le à votre ordinateur, puis spécifiez à ADT de lancer votre application sur votre appareil à l'aide du choix de la cible dans la configuration de lancement. [Fig.10]

Les téléphones Android du marché

Les appareils Android ne sont pour le moment pas encore tout à fait implantés dans l'hexagone : Orange est le seul à distribuer le HTC Dream G1, l'appareil Android de première génération et déjà disponible aux Etats Unis depuis un an, il supporte la version 1.1 d'Android. Mais il y a fort à parier que cela se fera avec le temps, déjà son concurrent SFR devrait sortir le HTC Magic, équipé d'Android 1.5 – alias « Cupcake » - et d'autres constructeurs devraient rapidement lui emboîter le pas, parmi lesquels Samsung, Acer, Dell, etc., d'après les rumeurs et annonces officielles. Un troisième modèle HTC, le « Hero », s'illustre déjà par quelques photos et vidéos sur la toile. Pour le moment ces téléphones ne semblent pas encore tout à fait prêts pour un usage professionnel (pas de synchronisation du calendrier et des contacts pour Exchange ou Lotus Notes par exemple), mais il est certain que les éditeurs combleront vite ce vide en proposant leurs propres applications Android.

Fig.A

Fig.B

Certains constructeurs semblent déjà intéressés pour intégrer le système d'exploitation dans leurs prochains netbooks. Une initiative peut-être encore précoce pour le moment au vu des applications disponibles et des usages permis par le système; à moins que chaque éditeur n'apporte son lot de nouveautés et d'applications propres … en route vers un nouveau monde des distributions multiples/forks à la façon de Linux ou retour vers J2ME ? Seul l'avenir nous le dira ! Le HTC Dream aussi appelé « G1 » est commercialisé en France par Orange [Fig.A]

Le HTC Magic ou « G2 » est sorti courant Mai chez SFR [Fig.B]

Fig.6

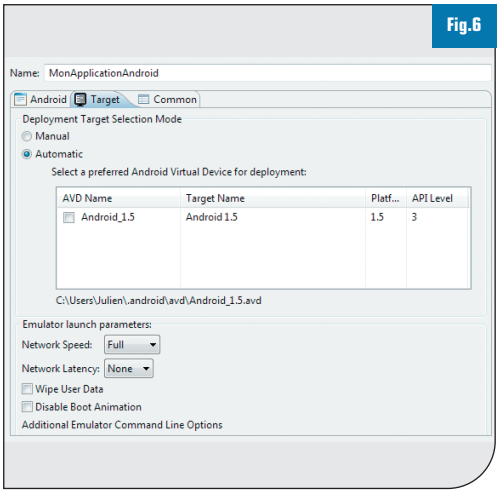
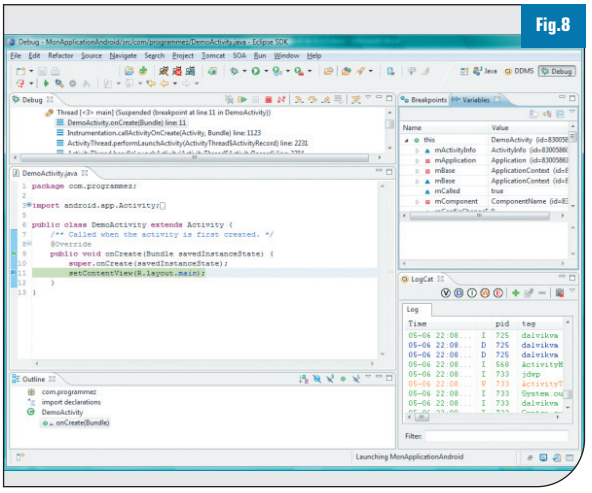


Fig.7

Fig.8



Publication sur le Android Market … les étapes de préparation

Votre application est conçue, déboguée et validée, il vous reste encore une dernière étape : la partager avec les autres utilisateurs. Pour cela, vous devrez vous inscrire sur l'Android Market moyennant la somme de 25$ afin de pouvoir publier vos applications (pour vous enregistrer, rendez-vous à cette adresse : http://market.android.com/publish). Cependant, ne vous y précipitez pas tout de suite, il vous reste encore quelques étapes avant de pouvoir publier votre application. Vous devez tout d'abord versionner votre application avec les attributs XML android:versionCode et android:versionName dans le manifest Android.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res
/android"
      package="com.programmez "
      android:versionCode="2"
      android:versionName="1.1">
   <application android:icon="@drawable/icon" android:label
="@string/app_name">
        ...
    </application>
</manifest>
```

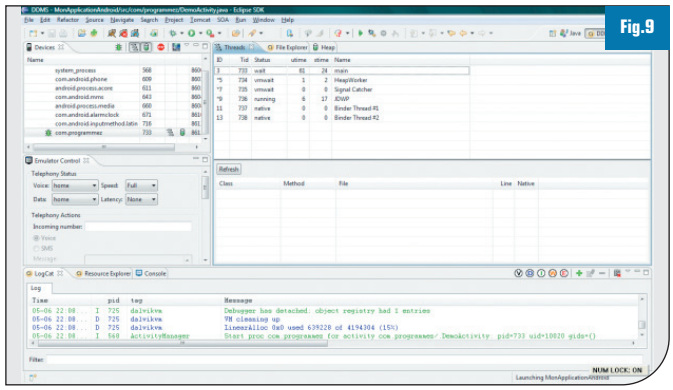


Fig.9

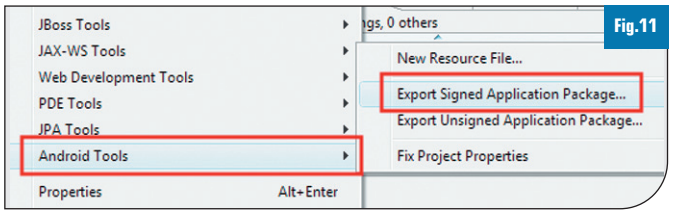


Fig.11

La valeur *versionCode* est utilisée en interne pour vérifier les mises à jour alors que la valeur de *versionName* est uniquement présente pour montrer la version de l'application à l'utilisateur, sans aucun impact sur un quelconque mécanisme interne.
La dernière étape consiste à signer l'application – le fichier .apk - à l'aide d'un outil : *keytool* et *jarsigner* ou plus simplement avec ADT directement en effectuant un clic droit sur le nom du projet dans l'explorateur > **Android Tools > Export Signed Application Package …** [Fig.11] Créez ou utilisez un *keystore* existant - en remplissant les champs demandés – puis définissez l'emplacement de destination de votre fichier .apk signé puis cliquez sur **Finish.** [Fig.12]
Une fois votre application signée, n'hésitez pas à également tester celle-ci en l'installant sur un appareil physique. Sans cela, tous vos efforts risquent d'être vains si les utilisateurs n'arrivent pas à le déployer surtout s'ils ont payé pour obtenir votre application.
La publication sur le Market Android nécessite un enregistrement préalable. Une fois effectué, vous pourrez télécharger votre application via le formulaire de publication. A l'heure de la rédaction de cet article, la France n'est pas éligible pour la publication d'applications payantes. [Fig.13] Avant de publier votre application, vérifiez bien que celle-ci rentre dans le cadre des critères d'acceptation de l'Android Market.

Conclusion

Cet article s'est volontairement voulu pragmatique dans la mise en œuvre et le cycle de développement d'une application Android, de la configuration du poste du développeur jusqu'à la publication sur l'Android Market. Cependant, il faudrait bien plus que quelques pages pour détailler la réalisation complète d'une application. Pour de plus amples informations sur la création d'une application Android, Google met à disposition des développeurs une section composée d'un guide et d'un tutorial (Notepad Android) pour vous aider à appréhender cette plate-forme qui est, somme toute, relativement vaste.

Les références

- Zone développeurs Android : http://developer.android.com
- Android Market : http://market.android.com/publish
- Blog Google Android : http://android-developers.blogspot.com

■ Julien **Chable** – *Consultant/Développeur indépendant SharePoint/.NET/JEE et Android*
http://julien.chable.net

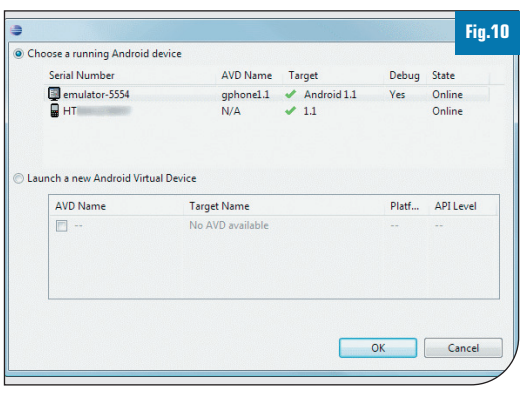


Fig.10

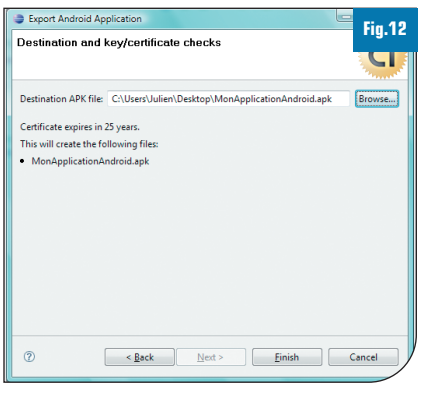


Fig.12

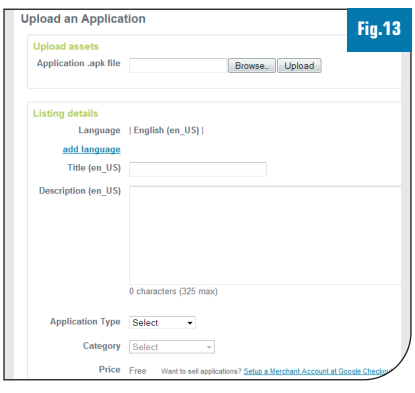


Fig.13

Java 7 : La dernière ligne droite

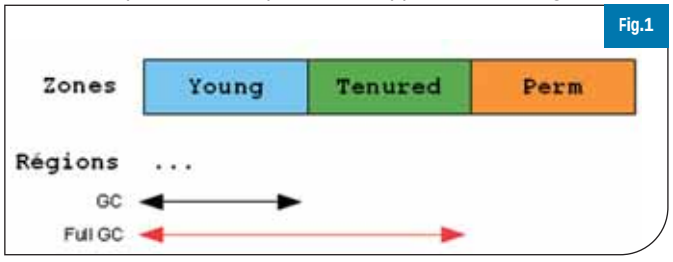
Annoncée il y a bientôt 2 ans de cela, la première version entièrement open-source du JDK de Sun amorce sa dernière ligne droite. Après de longs mois de débats sur le contenu de cette nouvelle mouture, qui ont agité la blogosphère des leaders Java, les traits de la version 7 se sont dessinés plus précisément ces derniers mois, permettant d'avoir une meilleure vision de son contenu.

Garbage Collector G1

On commence tout d'abord par une amélioration du JDK qui ne sera pas forcément visible et utile au commun des développeurs Java, mais qui devrait s'avérer très utile pour les grosses applications en production. Elle concerne le Garbage Collector (GC) utilisé par la machine virtuelle Java de Sun. Jusqu' alors, elle était dotée d'un GC générationnel dont le principe est de traiter de manière séparée les objets créés suivant leur ancienneté. Plus précisément, le tas est composé de 3 sections principales dites "Young", "Tenured" et "Perm" elles-mêmes divisées en zones secondaires [Fig.1]. Les deux premières sections sont utilisées pour gérer le cycle de vie d'un objet créé, et ce dernier sera éventuellement déplacé lors du passage du GC d'une zone à l'autre. La saturation de la section "Tenured" s'avérant très coûteuse puisqu'elle nécessite un appel au Full GC (déclenchable également via la méthode System.gc() mais de manière non prédictible) qui réalise un traitement en profondeur, beaucoup plus gourmand en ressources. La section "Perm" est quant à elle un peu différente des deux autres puisqu'elle contient les objets n'ayant pas besoin d'être nettoyés par le GC, par exemple les classes chargées par le class loader lors de l'exécution d'un programme.

Bien qu'assez performant, ce GC a vécu et présente un certain nombre de défauts qu'il était temps de corriger. En premier lieu, son manque de prédictibilité ne permet pas de prévoir efficacement les temps de pause destinés au passage du GC. En outre, il n'a pas été conçu pour bénéficier des environnements multiprocesseurs et ne tire donc pas parti de la parallélisation et de la concurrence. Pour répondre à ces problématiques, Sun s'est dirigé vers un GC de type Garbage-First (G1) qui bénéficie d'une meilleure granularité dans le découpage des régions destinées au stockage des objets [Fig.2]. Ce découpage plus fin s'accompagne d'un ensemble de métriques liées à ces régions et qui permettent au GC d'avoir une vision globale de leur état et de savoir précisément ce qu'il doit faire du contenu de ces zones. Cette meilleure utilisation du tas va permettre en pratique de paralléliser les traitements du GC et donc de profiter pleinement des environnements multiprocesseurs qui se développent à vitesse grand V. En outre, les métriques mises en place permettent une bien meilleure prédiction des temps de pauses destinés au GC.

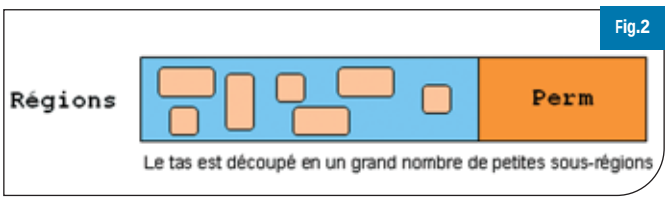
Déjà présent au sein du JDK 7 depuis le début de l'année, l'intégration du G1 au sein de l'update 14 du JDK 6 est d'ores et déjà réalisée.



Représentation mémoire du GC générationnel actuel

Jigsaw et la modularisation du JDK

En constante évolution depuis sa création, le JDK n'a cessé de grossir au cours des dernières années pour atteindre une taille de quasiment 50 Mo aujourd'hui. Ceci rendant les applications assez lentes au démarrage. L'apparition du Java Kernel au sein de l'update 10 du JDK 6 montre bien la volonté de Sun de le dégraisser grandement et de le rendre plus modulaire afin d'optimiser le temps de chargement des applications Java. C'est dans cette optique que sont nées les JSR 277 et 294 dont le but est de pouvoir rendre modulaire le langage au cœur même de la JVM. Compte tenu de l'enjeu que représente la gestion des dépendances d'une application, ces JSR ont donné lieu à de vifs échanges et ont finalement vu Sun réunir les experts travaillant dessus au sein du projet Jigsaw, dont le scope de travail a été réduit. Désormais, ce projet a pour but de définir un système modulaire bas niveau, accessible également aux développeurs et pleinement supporté par Sun, mais qui malheureusement ne sera pas intégré aux spécifications de Java SE 7. Ainsi, il est bien possible que seule l'implémentation du JDK 7 de Sun le supporte ! A terme, il est fort probable qu'après la sortie de cette nouvelle release une JSR soit constituée pour répondre de manière standard à ces problématiques. Actuellement, la tendance serait de se tourner vers le système modulaire proposé par le framework OSGI déjà grandement supporté par ailleurs et considéré comme un standard dans le domaine. En pratique, l'implémentation retenue donnera lieu à l'introduction du mot clé module applicable sur un package mais également sur une classe. Son utilisation permettant de définir un nouveau niveau de visibilité puisqu'en l'appliquant on précise que la classe correspondante est seulement visible des autres classes du même module. Sur le même modèle que le fichier de méta-données package-info.java, le fichier module-info.java est introduit et permet de spécifier un certain nombre d'informations sur un module donné :



Représentation mémoire du GC G1

```
@Version("2.1")
@MainClass("fr.programmez.Main")
@ImportModules(
    @ImportModule(name="java.se.core", version="1.7+"),
   @ImportModule(name="fr.module1", version="0.1", reexport
="true")
)
@ExportResources({"fr/programmez/resources/**"})
module fr.programmez ;
```

Ici, on voit bien que l'on définit un module sur le package fr.programmez en précisant son numéro de version ainsi que ses dépendances et la version de ces dernières. La volonté de modulariser le JDK se retrouve bel et bien dans ce simple exemple puisque l'on peut voir qu'il sera possible de préciser quel module de l'API Java standard est nécessaire à notre programme. Ainsi, seules les parties nécessaires seront éventuellement téléchargées au premier lancement du programme. Le fichier de la classe principale Main aurait le contenu suivant :

```
module fr.programmez ;
package fr.programmez ;

module class Main {
 …
}
```

Une fois l'application compilée, la création du module se réalise via l'outil JAM qui va créer une archive jam suivant les informations définies dans le fichier module-info.java. Au lancement de l'application, la machine virtuelle est ensuite capable de prendre en compte les informations apportées par le système JAM. Enfin, il est important de souligner que l'introduction du mot-clé module ne vient pas briser la compatibilité ascendante du langage puisqu'il ne sera reconnu que là où il peut être utilisé, c'est-à-dire à l'extérieur des blocs de déclaration d'interfaces ou de classes. Ce système de module ne sera pas une obligation puisqu'il sera encore possible de continuer à utiliser ses programmes comme on le fait actuellement, et ce par souci de compatibilité toujours.

NIO 2

Jusqu'à présent, force est de constater que les fonctionnalités permettant de manipuler des Entrées / Sorties en Java n'étaient guère avancées et qu'il était bien souvent nécessaire de bidouiller plus ou moins pour arriver à ses fins. Qui n'a jamais eu de problème lors de la suppression d'un fichier et est resté étonné de ne voir aucune erreur Java levée ? Qui n'est pas resté frustré devant l'impossibilité d'étendre l'API actuelle ? Ces exemples plus que basiques illustrent bien les manques actuels. Poursuivant le travail entamé par le projet NIO depuis le JDK 1.4, le projet NIO 2 apporte une nouvelle API de fichier, définie dans le nouveau package dédié java.nio.file . La notion de système de fichier représentée par la classe FileSystem en est le cœur, de même que la notion de path représentée par la classe éponyme Path [Fig.3].
L'exemple suivant illustre l'utilisation de ces dernières :

```
Import java.io.* ;
import java.nio.file.* ;
import static java.nio.file.StandardOperation.* ;

// récupération du système de fichier par défaut
FileSystem fileSystem = FileSystems.getDefault() ;
Path homeDir = fileSystem.getPath("/home/sylsau") ;
// récupération en passant par la classe utilitaire Paths
Path homeDir = Paths.get("/home/sylsau") ;
// résolution de chemins
Path relativeTmp = Paths.get("tmp") ;
Path absoluteTmp = relativeTmp.resolve(homeDir) ;
// Ecriture dans un fichier
OutputStream outputStream = absoluteTmp.newOutputStream(CREATE,
 APPEND) ;
outputStream.write(…) ;
```

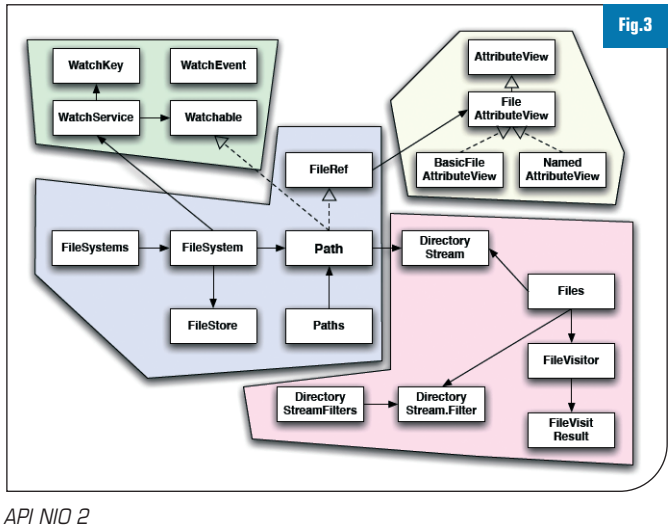
NIO 2 intègre également des fonctionnalités haut niveau permettant de copier ou de déplacer directement des fichiers représentés par des objets Path via les méthodes copyTo et moveTo de cette classe. Une simplification agréable pour le développeur jusqu'ici obligé de réaliser ses propres fonctions de copie et de déplacement. La création de liens symboliques est désormais possible et la gestion des permissions sur les fichiers se voit dotée d'une hiérarchie de classes basée sur l'interface FileAttributeView. Le parcours des éléments d'un répertoire est également amélioré et simplifié avec l'itérateur DirectoryStream récupérable depuis un objet Path et qui offre en outre la possibilité de sélectionner les éléments respectant un certain pattern. Il est également possible de parcourir les éléments représentés par un objet Path et de leur appliquer un traitement. La méthode walkFileTree de la classe Files apporte cette fonctionnalité et est un bon exemple de mise en pratique du design pattern Visitor. Ces derniers points sont illustrés ci-dessous :

```
// récupération de tous les fichiers java d'un répertoire
Path dir = Paths.get("/home/sylsau/files") ;
DirectoryStream<Path>  javaFiles  =  dir.newDirectoryStream
("*.java") ;

for(Path javaFile : javaFiles)
     System.out.println(javaFile.getName()) ;

// parcours récursif des éléments d'un fichier et application
 d'un traitement spécifique
```

Fig.3



API NIO 2

```
Files.walkFileTree(dir, new SimpleFileVisitor(dir)) ;

// récupération des attributs d'un fichier
Path file = Paths.get("/home/sylsau/file.txt") ;
BasicFileAttributes attributes = Attributes.readPosixFile
Attributes(file, true) ;
Set<PosixFilePermission> permissions = attributes.permissions() ;
permissions.owner() ; …
```

D'autre part, NIO 2 introduit également un ensemble de classes permettant la surveillance de fichiers ou répertoires représentés par des objets Path. Il est possible de préciser le type d'évènements pour lesquels on souhaite être alerté comme le montre l'exemple suivant :

```
import static java.nio.file.StandardWatchEventKind.*;

Path dirToWatch = Paths.get("/home/sylsau");
WatchService watcher = FileSystems.getDefault().newWatch
Service();
WatchKey key = deploy.register(watcher, ENTRY_CREATE, ENTRY
_DELETE, ENTRY_MODIFY);

for(;;){
    key = watcher.take();
    for(WatchEvent<?> event : key.pollEvents()){
         // récupération du type de l'event
         Kind kindEvent = ev.kind();
         …
    }
}
```

Ces quelques exemples sont loin de couvrir les importantes possibilités offertes par NIO 2 mais ont pour but de vous donner envie de les tester par vous-même puisque l'API est d'ores et déjà implémentée dans la dernière milestone disponible du JDK 7. Enfin, NIO 2 ne se limite pas à cela et vient terminer le travail entamé par NIO sur les sockets, avec l'apparition de l'interface NetworkChannel. En outre, les entrées / sorties asynchrones sont désormais supportées pour les fichiers et les sockets via l'utilisation des objets Future, et elles supportent la gestion des pools de threads.

JSR 166y et framework de parallélisation

Introduite avec le JDK 5 et basée sur la JSR 166, l'API Concurrent avait permis de rendre accessibles au plus grand nombre des fonctionnalités de la programmation concurrente réservées jusque là aux développeurs confirmés. En constante évolution depuis, avec l'ajout d'un certain nombre de classes spécifiques aux traitements parallèles dans la version 6 du JDK, l'API continue son petit bonhomme de chemin avec le JDK 7 et la JSR 166y. Au programme, l'introduction du mécanisme de fork / join permettant d'exécuter facilement des traitements parallèles et ainsi, de gagner en performance. Au cœur de cette API se trouvent la classe ForkJoinPool et un ensemble de tâches permettant la décomposition de problèmes. L'utilisation de cette classe n'est cependant pas automatique puisqu'il est nécessaire de bien analyser le problème rencontré pour préparer au mieux la décomposition en sous-problèmes … Le numéro 115 de Programmez ! donne un bon aperçu de l'utilisation du mécanisme de fork / join à l'aide de la classe RecursiveAction. Un des scénarii les plus fréquents d'utilisation de la parallélisation concerne les traitements effectués sur des collections d'objets Java. Dans cette optique, le framework fournit un niveau d'abstraction encore plus élevé avec la classe ParallelArray. Liée à un objet ForkJoinPool, elle permet de bénéficier de la parallélisation des fonctions de filtrage et de map, notamment comme le démontre l'exemple suivant :

```
// instanciation du ForkJoinPool avec le nombre maximal de
processeurs disponibles
ForkJoinPool forkJoinPool = new ForkJoinPool(getRuntime().
availableProcessors()) ;
MyData[] myData = … ;
ParallelArray<MyData> myDatasArray = new ParrallelArray<My
Data>(forkJoinPool, myData) ;

// Filtre sur une propriété de MyData par exemple
Ops.Predicate<MyData> isValid = new Ops.Predicate<MyData>(){
     public boolean op(MyData myData){
          return myData.isValid() ;
     }
} ;

// Map sur une valeur entière de myData
Ops.Predicate<MyData> values = new Ops.ObjectToInt<MyData>(){
     public int op(MyData myData){
          return myData.value() ;
     }
} ;

// Application simple sur myDatasArray
SummaryStatistics<Integer> summary = myDatasArray.withFilter
(isValid).withMapping(values).summary ;
System.out.println("Moyenne = " + summary.average() + " / Max =
 " + myDatasArray.get(summary.indexOfMax()).getName()) ;
```

Compte tenu de l'importance que va prendre le multi-cœur dans les années à venir, c'est vraiment une très bonne chose pour le monde Java et l'on peut espérer que d'autres améliorations viendront se greffer à l'API Concurrent dans le futur et combler le retard de Java dans ce domaine.

Swing Application Framework

Bien qu'ayant grandement évolué depuis sa création, il souffre d'un certain nombre de manques et oblige bien souvent les développeurs à écrire des applications complexes avec les moyens du bord ! Pour remédier à cela, le projet d'un framework Swing Application est né. Son but principal est de simplifier et de rendre plus robuste les applications utilisant Swing dans le futur. Son champ d'action est assez vaste et prometteur :

- Gestion du cycle de vie d'une application avec des étapes bien définies
- Injection de ressources et gestion de l'internationalisation (i18n)
- Persistance des informations de la session avec notamment la sauvegarde de la configuration de l'affichage et de la position des fenêtres de manière automatique
- Meilleure gestion de la notion d'Action avec en sus le support de tâches s'exécutant en arrière-plan

Au centre du framework, la classe Application qui permet la ges-

tion du cycle de vie via les méthodes *launch, startup, shutdown* et *exit* [Fig.4]. La gestion des actions se fait facilement via la nouvelle annotation @Action permettant de marquer une méthode de la classe de l'application. En outre, l'injection de ressources se fait de manière transparente via l'utilisation d'un fichier properties comme on peut le constater dans l'exemple ci-dessous :

```
// exemple simple d'application avec un Hello Programmez ! ;
public class HelloProgrammez extends Application{
      @Override
      protected void startup(String[] args){
            JButton button = new JButton();
            button.setName("myButton");
            button.setAction(getAction("sayHello"));
            JPanel panel = new JPanel();
            panel.add(button);
            show(panel);
      }

      @Action
      public void sayHello(ActionEvent ae){
            System.out.println("Hello Programmez ! ;)");
      }
      private Action getAction(String actionName) {
             ApplicationContext ac = ApplicationContext.get
Instance();
          return ac.getActionMap(getClass(), this).get(action
Name);
      }

      public static void main(String[] args){
            launch(HelloProgrammez.class, args);
      }
}
// contenu du fichier HelloProgrammez.properties
Application.title=HelloProgrammez
myButton.font=times italic 12
myButton.text=Say Hello
```

Enfin, le JDK 7 apportera également quelques nouveautés au framework Swing en lui-même avec l'intégration probable d'un certain nombre de composants issus du projet SwingX tels que le JXLayer ou bien le DatePicker.

Annotations de types

Apparues avec le JDK 5, les annotations Java constituent un outil très apprécié des développeurs mais leur portée reste pour le moment limitée. Avec cette nouvelle mouture du JDK, les annotations pourront être utilisées sur des types Java pour spécifier plus précisément le comportement attendu d'une méthode ou bien d'une variable. Plusieurs annotations feront ainsi leur apparition et seront détectées dès la phase de compilation ! Parmi celles-ci, on peut citer @Nullable, @NonNull et @ReadOnly notamment dont le nom indique clairement le comportement attendu. Cette fonctionnalité particulièrement intéressante a déjà son lot de détracteurs puisqu'elle risque de rendre les programmes beaucoup plus verbeux comme on peut le constater :

```
public class UnModifiableList<T> implements @ReadOnly List<
```

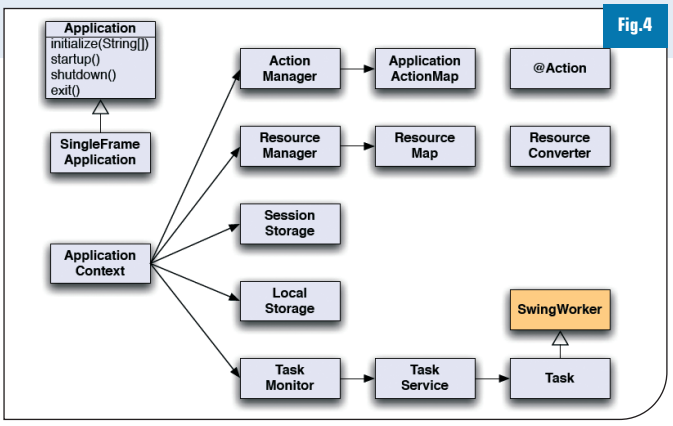


Fig.4

Swing Application Framework

```
@ReadOnly T>{ …}
…
List<@NonNull String> list = new @ReadOnly ArrayList<String>();
```

Project Coin

Depuis la sortie de Java 6 en 2007, de nombreuses propositions ont été faites pour simplifier la syntaxe du langage Java. Sun ayant mis à contribution la communauté Java afin de recueillir l'avis du plus grand nombre, le projet Coin a été créé pour fédérer et regrouper toutes ces idées de petites évolutions du langage. La présentation du projet à la grande messe JavaOne au début du mois de Juin a donné un bon aperçu des principales idées qui devraient être implémentées dans le JDK 7. On retiendra ainsi :

- Le switch avec String : switch(str){ case "foo" : return 1; case "bar" : … }
- La création et l'utilisation de listes et de maps comme pour les tableaux : Map<Integer, String> map = {1 : "One", 2 : "Two"}; List<String> list = ["a", "b"]; map[0] = "one"; …
- Le multi catch d'exceptions : try{ … } catch(Exc1 | Exc2 ex) { … }
- L'introduction de l'opérateur ?: pour gérer plus facilement les tests de valeurs nulles : Integer wrapper = null ; int i = wrapper ?: -1 ;
- L'opérateur diamond <> pour simplifier la création d'objets génériques : Map<String, List<String>> anagrams = new HashMap<>();
- Gestion des null lors d'appels de méthodes via l'opérateur ?. : if(obj != null) { return obj.method(); } deviendrait ainsi return obj?.method(); qui retournerait null si obj vaut null et ce de manière implicite

Ces petits sucres syntaxiques ne viendront pas bouleverser la vie du programmeur Java à coup sûr, mais ils peuvent représenter un gain de temps certain comme c'est le cas avec la création de l'opérateur ?. .

Java 7 dans les mois à venir …

Supportant déjà le G1 ainsi que NIO 2, le JDK 7, sorti récemment en milestone 3, va grandement évoluer dans les mois qui viennent avec l'implémentation des fonctionnalités décrites dans cet article.

Petit tour d'horizon du planning des mois à venir

Juillet 2009	M4	Report des updates de java 6 update 10
Septembre 2009	M5	Swing Application Framework
Octobre 2009	M6	Project Coin
Décembre 2009	M7	Contenu restant à déterminer
Février 2010	Final Milestone	
Avril 2010	Release finale JDK 7 attendue …	

■ Sylvain **Saurel** – Ingénieur d'Etudes Java / JEE
ACP – www.acp-qualife.fr - sylvain.saurel@gmail.com

Microsoft Surface : quand la table devient interactive !

Vous en avez sans doute entendu parler et peut-être eu la chance de l'essayer, la table Microsoft Surface a le vent en poupe depuis quelque temps. Présente sur de nombreux événements tels que les Microsoft TechDays, Microsoft Days, WygDay et d'autres encore, le grand public semble montrer un réel engouement à son égard. Ce que l'on appelle le Surface Computing commence à être pris au sérieux !

La table Surface embarque la technologie de multitouche développée par Microsoft. Elle est capable d'interagir avec des objets externes via des tags afin de pouvoir par exemple, importer, manipuler et transférer des médias. Au delà de l'aspect technologique, cette table vient apporter une véritable dimension sociale aux applications nouvelles générations.

Au sein de cet article nous allons découvrir et aborder le développement d'applications Surface.

Les pré-requis

Le développement d'applications Surface est assez restreint étant donné que l'éditeur n'a pas encore ouvert le SDK 1.0 au public. Seuls les détenteurs d'une Surface ont la chance de pouvoir réaliser leurs premiers développements Surface. Voici une liste des pré-requis.

Pré-requis logiciels pour le développement Surface

- Une version 32 bits de Windows Vista à savoir : Windows Vista Business, Windows Vista Enterprise ou Windows Vista Ultimate
- Windows Vista SP1
- Microsoft Visual C# 2008 Express Edition ou Microsoft Visual Studio 2008 (avec .NET Framework version 3.5)
- Microsoft XNA Framework Redistributable 2.0
- Microsoft Surface SDK 1.0, Workstation Edition

Pré-requis matériels

- Un écran capable d'afficher une résolution égale ou supérieur à 1280x960 ou un écran widescreen avec une résolution de 1440x900.

Cette résolution est nécessaire pour pouvoir lancer le Surface Simulator. Le Surface Simulator est une application qui permet de débugger et tester votre application Surface dans un contexte virtuel, ainsi vous n'êtes pas obligé de déployer votre application sur une véritable Surface pour pouvoir l'utiliser. Si votre écran ne supporte pas les résolutions énoncées vous ne pourrez pas lancer ce simulator.

- Une carte graphique supportant Windows Aero

Si votre carte graphique ne supporte pas Windows Aero, Surface Simulator et Surface Shell ne fonctionneront pas correctement.

La configuration matérielle d'une Surface

- Windows Vista Business Edition
- Processeur : Intel core 2 duo E6400 2.13GHz 2.13 Ghz
- Mémoire vive : 2 GB
- Carte Graphique : Radeon X1650

La table Surface et son fonctionnement

Surface se présente comme un caisson composé d'un ordinateur, de lampes, de caméras et d'une plaque tactile [Fig.1].

Lorsqu'on démarre une Surface, on s'aperçoit très rapidement qu'il s'agit d'un Windows Vista. Une fois Windows Vista démarré, un Shell dédié à la gestion d'applications Surface se lance automatiquement en plein écran. Ce Shell permet de choisir et lancer des applications. Les applications en question étant des applications WPF. Néanmoins, elles sont agrémentées d'une couche « Surface » qui permet de gérer, par exemple, les événements tactiles, d'avoir de nouveaux contrôles, gérer les tags, etc.

Le contrôle ScatterView

Le contrôle *ScatterView* est sans doute le contrôle le plus représentatif de la table Microsoft Surface, il permet en zéro ligne de code de rendre manipulable et étirable n'importe quel objet. En effet, Le contrôle *Scatterview* de Surface permet de manipuler des éléments WPF ou Surface dans une zone spécifiée. Ainsi chaque enfant d'un *Scatterview* est manipulable et étirable à souhait pour autant que le développeur ait décidé de vous laisser

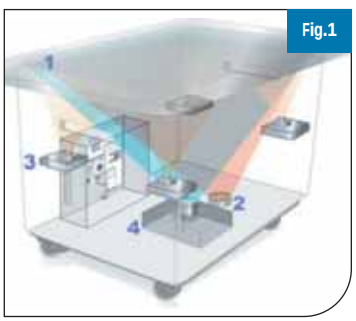


Fig.1

Fig.2

cette possibilité. Voici un exemple de code XAML à implémenter :

```
<s:ScatterView x:Name="mainScatterView">
          <s:ScatterViewItem />
</s:ScatterView>
```

Bien entendu, un *ScatterViewItem* peut comprendre des conteneurs, des images, des vidéos, des contrôles personnalisés, etc.[Fig.2]

Le Databinding et le contrôle ScatterView

Comme vous l'avez compris, Surface repose sur WPF et permet donc de profiter de tous les avantages de cette technologie. Nous allons apprendre à *DataBinder* vos photos directement avec un *ScatterView,* comme nous aurions pu le faire avec une *ListBox WPF.* Ainsi toutes les photos de votre collection, seront insérées dans le contrôle *ScatterView* et seront étirables et manipulables.
Dans le fichier, *SurfaceWindow1.xaml.cs* qui n'est autre que le code *behind* de la fenêtre, nous allons insérer le code suivant :

```
protected override void OnInitialized(EventArgs e)
{
    base.OnInitialized(e);
const string shellKey = @"HKEY_LOCAL_MACHINE\SOFTWARE\Micro
soft\Windows\CurrentVersion\explorer\Shell Folders";
string  imagesPath  =  (string)Microsoft.Win32.Registry.Get
Value(shellKey, "CommonPictures", null) + @"\Sample Pictures";
    Try
    {
             mainScatterView.ItemsSource =
            System.IO.Directory.GetFiles(imagesPath, "*.jpg");
    }
    catch (System.IO.DirectoryNotFoundException)
    {
        // Handle exceptions as needed.
    }
}
```

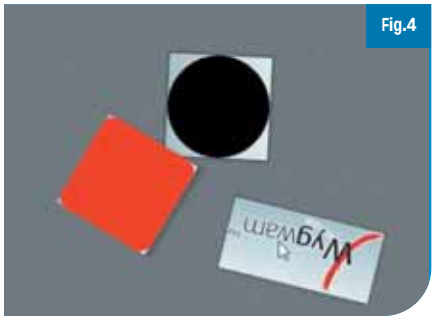


Fig.4

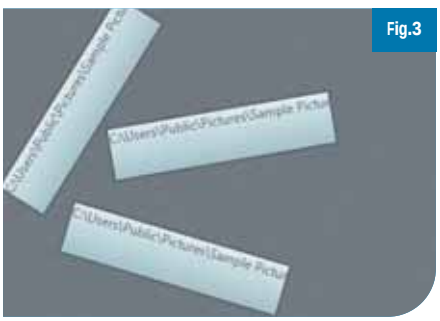


Fig.3

Regardons le résultat. [Fig.3]
Il ne reste plus qu'à modifier le *DataTemplate* de nos items pour pouvoir afficher la photo plutôt que le *ToString* des fichiers. Regardons côté XAML comment modifier ce template :

```
<s:SurfaceWindow
    x:Class="ScatterViewDataBind.SurfaceWindow1"
   xmlns=http://schemas.microsoft.com/winfx/2006/xaml/presentation
    xmlns:x=http://schemas.microsoft.com/winfx/2006/xaml
    xmlns:s=http://schemas.microsoft.com/surface/2008
    Title="ScatterViewDataBind">
    <s:SurfaceWindow.Resources>
      <ImageBrush x:Key="WindowBackground"
                  Stretch="None" Opacity="0.6"
                  ImageSource="pack://application:,,,/
Resources/WindowBackground.jpg"/>
    </s:SurfaceWindow.Resources>
    <Grid Background="{StaticResource WindowBackground}" >
      <s:ScatterView x:Name="MainScatterView">
        <s:ScatterView.ItemTemplate>
          <DataTemplate>
            <Image Source="{Binding}"/>
          </DataTemplate></s:ScatterView.ItemTemplate>
      </s:ScatterView>
    </Grid>
</s:SurfaceWindow>
```

Voici le résultat : [Fig.2]
En très peu de lignes de code il est donc possible d'arriver à des effets graphiques et ergonomiques très intéressants. Comme vous pouvez le remarquer, si au sein de votre collection d'images vous avez un PNG transparent, malheureusement le style du *ScatterViewItem* par défaut embarque un dégradé de gris, limitant ainsi l'intérêt de votre image transparente.

Personnaliser un ScatterView

En pratique, le contrôle *Scatterview* Surface propose comme tout contrôle WPF un style par défaut. Nous allons voir comment le mettre à nu pour pouvoir profiter de la transparence du fond (background) des enfants : les *ScatterViewItem*. [Fig.4]
En redéfinissant le template d'un *ScatterViewItem,* cela supprime tous les styles par défaut. Voyons comment appliquer cela aux enfants du ScatterView via XAML.

```
<s:ScatterView>
     <s:ScatterView.ItemContainerStyle>
          <Style TargetType="{x:Type s:ScatterViewItem}">
               <Setter Property="Template">
                    <Setter.Value>
                      <ControlTemplate TargetType="{x:Type
 s:ScatterViewItem}">
                                <ContentPresenter Horizontal
Alignment="{TemplateBinding  HorizontalContentAlignment}"
   VerticalAlignment="{TemplateBinding   VerticalContent
Alignment}"  SnapsToDevicePixels="{TemplateBinding  SnapsTo
DevicePixels}"/>
                           </ControlTemplate>
```

```
                    </Setter.Value>
                </Setter>
            </Style>
        </s:ScatterView.ItemContainerStyle>
    <s:ScatterViewItem />
    <s:ScatterViewItem />
    <s:ScatterViewItem />
</s:ScatterView>
```

[Fig.4]

A présent, voyons comment interdire la rotation d'un ScatterViewItem.

```
<s:ScatterViewItem CanRotate="False" />
```

Il en est de même pour la fonctionnalité de "redimensionnement".

```
<s:ScatterViewItem CanScale="False" />
```

Voilà, vous êtes à présent capable de personnaliser un *ScatterView* et ses *ScatterViewItem* à votre guise, que ce soit d'un point de vue rendu ou fonctionnel.

Les Contacts [Fig.5]

Les Contacts répertorient tous les événements Multitouche de la table Surface. C'est grâce à eux que vous allez pouvoir capter les événements utilisateurs sur vos différents contrôles. Les Contacts sont par défaut exclusivement dédiés aux contrôles Surface. Voici la liste des contacts à laquelle il est possible de s'abonner :

- ContactChanged : Levé au moindre contact et modification de contact avec la Surface
- ContactDown : Levé lorsque vous appuyez sur la surface de la table
- ContactEnter : Levé lorsque vous entrez dans un containeur avec votre doigt par exemple
- ContactHoldGesture : Levé lorsque vous laissez votre doigt appuyé sur la Surface
- ContactLeave : Levé lorsque vous quittez la surface tactile
- ContactTapGesture : Levé lorsque vous appuyez sur un bouton par exemple
- ContactUp : Levé lorsque vous relâchez le ContactDown

Cette multitude d'événements permet de gérer la plupart des cas ergonomiques qu'il est possible de rencontrer lors de développement d'applications Surface. Néanmoins, il reste quelques cas non traités, pour cela Microsoft a mis à notre disposition un ContactHelper pour faciliter les manipulations complexes d'objet, mais tout d'abord, commençons par le fonctionnement classique des Contacts.

Fig.6

Fig.5

S'abonner à un événement Contact

Par défaut, seuls les éléments Surface sont aptes à écouter les événements Contacts. Pour cela, il suffit, comme à l'accoutumée, d'ajouter un eventhandler :

```
void SurfaceWindow1_Loaded(object sender, RoutedEventArgs e)
{
  LayootRoot.ContactDown += new ContactEventHandler(Layoot
Root_ContactDown);
}

void LayootRoot_ContactDown(object sender, ContactEventArgs e)
{
  MessageBox.Show("ContactDown !");
}
```

Les Contacts et WPF

Si vous êtes amené à migrer des applications WPF vers la plateforme Surface, vous allez être confronté à de nouveaux choix ergonomiques. Afin d'éviter de modifier tout votre existant, il peut être intéressant de venir greffer le système de Contacts sur des éléments WPF classiques. Voici comment enrichir le contrôle Image de WPF de la notion de Contact :

```
Contacts.AddContactDownHandler(votreImage, votreHandler);
```

Le ContactHelper.cs

La principale utilité du ContactHelper est la possibilité de récupérer le DraggedDelta. Le DraggedDelta n'est autre que la surface parcourue lors d'un Contact.

```
Vector draggedDelta = ContactsHelper.DraggedDelta(e.Contact,
  monControle);
```

Ainsi il est possible de récupérer la distance parcourue en X : draggedDelta.X et en Y draggedDelta.Y. L'objet **e** devant être de type ContactEventArgs. [Fig.6]

■ Guillaume **ANDRE**
Wygwam - Développeur RIA
http://www.guillaumeandre.com/

Enquête exclusive

Etes-vous satisfait de votre job ?

71% regrettent l'absence de perspectives d'évolution

Programmez ! a réalisé une enquête afin de mesurer votre satisfaction au travail*. Les réponses varient de façon importante selon les fonctions. Nous avons regroupé sur deux catégories principales les répondants : développeurs/ingénieurs d'une part et chefs de projet/architectes de l'autre. Les réponses diffèrent beaucoup entre ces deux profils. Mais un point commun s'impose : plus de 71% des salariés interrogés manquent de perspective d'évolution !

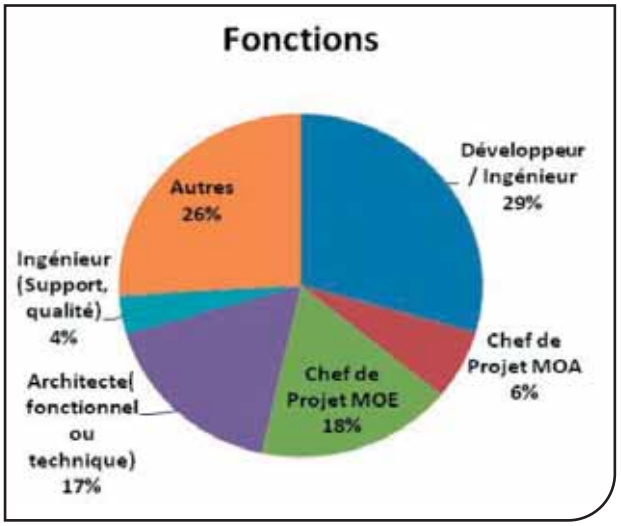
Vous aimez l'ambiance de la société

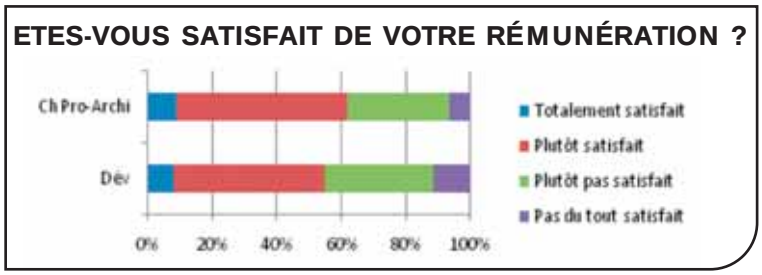
Commençons par une bonne nouvelle. Ce sera une des seules... ! La plupart des répondants aiment l'ambiance et le cadre de travail. Plus de 82% d'entre vous sont positifs pour qualifier les « conditions de travail (temps de transport, locaux, relations entre collègues...) ». Vous êtes 30% à vous déclarer très satisfaits, et 52,6% assez satisfaits.

No Future !

En revanche, à la question « Avez-vous de bonnes perspectives d'évolution au sein de votre entreprise ? », les réponses sont accablantes : Non, pour 62% des développeurs, et pour 56% des chefs de projets / architectes. En toute logique, vous ne pensez pas faire de vieux os chez votre employeur : moins du quart des répondants pensent qu'ils seront dans l'entreprise dans 3 ans. Dirigeants, chouchoutez vos cadres (chefs de projets, architectes) : ils sont 42,5% à hésiter !

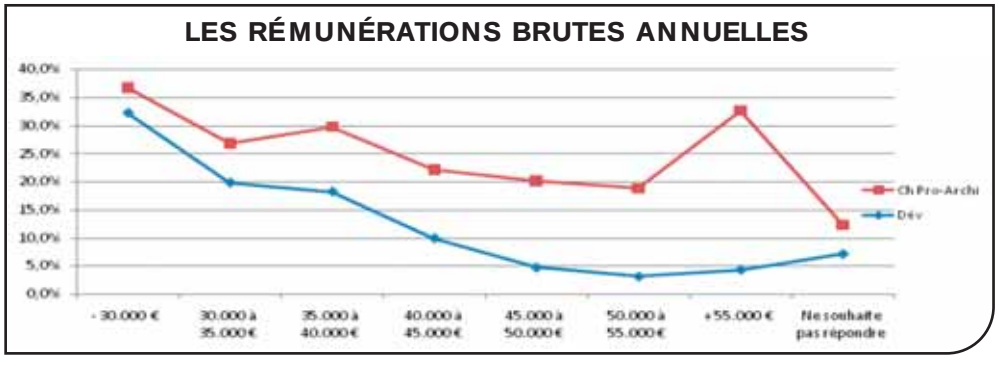
Le salaire vient en seconde cause d'insatisfaction, suivi par le « manque d'intérêt du travail », et le manque de « considération ». Ce dernier reproche est plus marqué chez les développeurs (52%) que chez les chefs de projet / architectes, qui en souffrent quand même pour 41% d'entre eux.

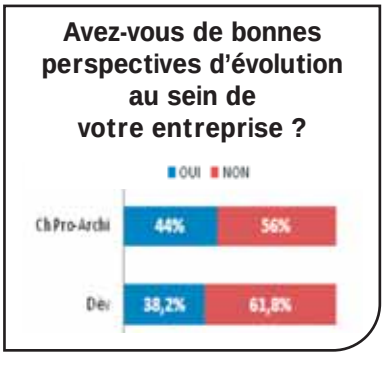




Pensez-vous être encore dans votre entreprise dans 3 ans ?

	Développeurs	Chef Projet / Archi.
Oui	23,30%	**24,40%**
Non	**47%**	33,10%
Ne sait pas	29,60%	**42,50%**





** Enquête effectuée en deux vagues : décembre 2008 et mai 2009 . Population étudiée : visiteurs du site programmez.com et cadres informatiques sollicités par e-mailing (fichier d'éditeurs logiciels.)*

Une forte demande d'amélioration des compétences

Dans les 3 années à venir, comment voyez-vous votre progression de carrière ?

	Développeurs	Chef Proj. / Archi.
Développement de compétences techniques supplémentaires.	52,9%	21,90%
Encadrement, direction de projet.	29,3%	56,10%
Evolution vers le fonctionnel.	5,40%	12,30%
Changement d'orientation (commercial, marketing, ressources humaines, ...)	4,20%	3%

L'enquête montre une forte demande de développement des compétences techniques .L'acquisition de celles-ci, passe, pour le développeur, par de nouvelles missions dans sa société (54%), et par la formation (45,2%), voire le changement d'employeur (41 %) ou...la création d'entreprise (14%) !

Les chefs de projets ou architectes pensent atteindre cet objectif essentiellement par leurs futures missions (68%), ensuite par un changement d'employeur (42%). Des formations adaptées sont citées dans 32% des cas.

Les chefs de projets / architectes estiment majoritairement que leur entreprise les aide à progresser (75%). En revanche, les développeurs ne sont que 52% à répondre par l'affirmative. 40% estiment ne pas être aidés. Employeurs, si vous voulez garder vos informaticiens, et si vous êtes intéressés à leur qualité de travail, vous savez ce qu'il vous reste à faire !

■ Jean **Kaminsky**

L'avis d'un dirigeant

« Les salariés attendent de leur employeur de la proactivité »

« Cette étude met en évidence que les salariés attendent de leurs employeurs une certaine proactivité dans leur évolution de carrière. Chez Ideo Technologies, les contacts mensuels ainsi que les entretiens annuels nous permettent de valider régulièrement les attentes de nos collaborateurs. Nous sommes en mesure de leur proposer un parcours d'évolution personnalisé. Certains profitent par exemple du programme de certification que nous avons mis en place. D'autres participent à nos projets R&D Open Source. Nous gardons une proximité avec l'ensemble de nos collaborateurs en utilisant l'esprit et les outils de l'Entreprise 2.0 sans négliger les moments d'échanges et de partages réels. »

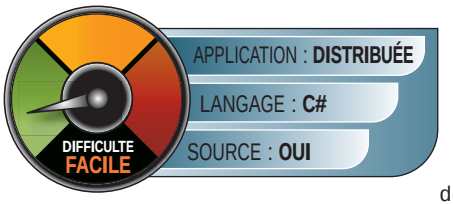
Jean-François Lufeaux, P-DG Ideo Technologies

Juniors et seniors

- **Les Chefs de projet et architectes ont une expérience professionnelle et par conséquent également un âge moyen, beaucoup plus élevés : 87,7% ont une expérience d'au moins 5 ans, et 60% de 10 ans et plus. Ces facteurs influeront sur toutes les réponses.**
- **A l'inverse, les développeurs et ingénieurs de l'échantillon ne sont que 47% à avoir une expérience de 5 ans et plus, et 26,5% de 10 ans et plus.**
- **Alors que le quart des développeurs travaille depuis moins de 3 ans, ils ne sont que 4,1% dans le groupe des chefs de projets et architectes.**
- **Le tiers des développeurs est Bac +2 (27,8%) ou autodidacte (6,8%). Seuls 16% de l'autre groupe ont 2 ans d'études ou moins.**
- **60% des chefs de projets ou architectes gagnent plus de 45 000 euros par an, alors que le salaire annuel des développeurs est dans 70% des cas inférieur à 40 000 euros.**

Découvrir l'environnement DSS et son langage de programmation visuel

DSS (Decentralized Software Services) est un environnement, initialement dédié à la robotique, et qui permet de créer des applications composites et distribuées. Très complexe en lui-même, sa mise en oeuvre et sa programmation peuvent paradoxalement s'avérer très aisées, voire ludiques. Amusons-nous un peu.



DSS est un environnement qui fédère des services orientés état. Il ne s'agit pas de service au sens 'Services Windows' du terme. Il s'agit de threads gérés par CCR, l'API de programmation concurrente pour .Net que nous avons découverte le mois dernier. Cette API a été conçue avec l'intention de minimiser autant que possible les risques de deadlock et elle présente également un mécanisme original de pilotage par messages émis dans des ports. Puisque nous avons lu Programmez! 120 et que nous savons que CCR gère des threads, nous pouvons nous demander en quoi DSS est-il différent ? DSS est une surcouche qui veut apporter de la robustesse à une application, ainsi que la composabilité et l'observabilité des éléments qui la constituent. La robustesse est basée sur le mécanisme de tolérance aux fautes de CCR. Rien de vraiment nouveau.Par contre, la composabilité est le fruit d'un mécanisme de souscription aux services et d'un système de déploiement qui sont le propre de DSS. Enfin, l'observabilité est apportée par une interface Web qui offre une console d'administration d'une part et présente le rendu sous la forme d'un fichier XML des services d'autre part. DSS est le fondement de Microsoft Robotics Studio, un environnement qui, comme son nom l'indique, permet de piloter des robots, ceux-ci étant du point de vue de la programmation des services orchestrés, tel service transmettant les données reçues par un capteur et tel autre actionnant un organe du robot, etc. Tout ceci est évidemment très alléchant, même si l'on n'a pas de robots à la maison. On peut très bien imaginer utiliser DSS pour des applications domotiques par exemples. DSS on s'en doute est très complexe. Il est possible de le programmer totalement en dur pour ceux qui aiment la difficulté. Fort heureusement DSS vient avec un add-in Visual Studio qui fait le plus gros du travail lors de la création d'un service et surtout, il vient accompagné d'un formidable outil, VPL, pour Visual Programming Language, qui permet de créer entièrement des applications composites distribuées sans aucune notion de programmation ou, pour être honnête, quasiment aucune, mais surtout sans la moindre difficulté pour un lecteur de Programmez! Les manipulations sont vraiment ludiques et nous allons nous y adonner. Mais attachons-nous d'abord à bien comprendre la nature d'un service DSS.

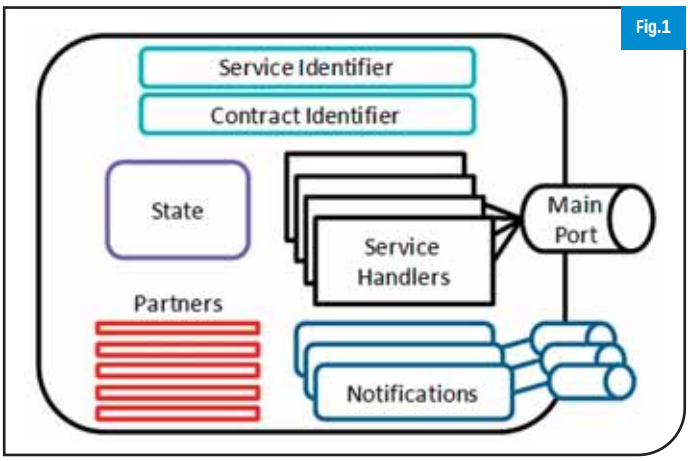


Fig.1

Anatomie d'un service DSS. Diagramme issu de la documentation du Toolkit CCR/DSS.

1 ANATOMIE D'UN SERVICE DSS

L'illustration ci-contre montre l'organisation interne du service DSS [Fig.1]. Nous avons d'abord un identificateur de service (Service Identifier) qui est une ressource assignée dynamiquement au service. Son rôle est bien évidemment de répertorier le service sur le réseau, condition sine qua non pour que d'autres services puissent communiquer avec lui et aussi pour que l'interface Web d'administration puisse y accéder. Vient ensuite l'identificateur de contrat (Contract Identifier) qui d'après la documentation est une description condensée du comportement d'un service. Le contrat est utilisé lors de la compilation pour créer des librairies de proxy. On peut d'une certaine façon dire que le contrat est à un service DSS ce que l'interface est à la programmation objet. Toutefois, le contrat c'est beaucoup plus que cela. La documentation n'est guère bavarde. Si on regarde le code d'un service, on voit que le contrat contient une instance des classes encapsulant: l'état du service, les opérations sur le port, les opérations de souscription au service. Le contrat est donc ce autour de quoi tout s'articule. Vient ensuite l'état du service que nous venons de mentionner. Cet état est une simple classe dont les données membres décrivent l'état lorsqu'on y accède par des propriétés. Viennent ensuite le port et les gestionnaires de requêtes. Ces derniers éléments sont des entités CCR exactement semblables à celles que nous avons découvertes le mois dernier. Enfin, les partenaires sont les services recensés comme devant interagir avec le présent service et les notifications sont un mécanisme de messages envoyés par le service pour notifier du changement de son état aux autres services qui y ont souscrit. Voilà donc la vision qu'il convient d'avoir et qui est suffisante pour nous lancer dans des manipulations.

2 CRÉER UN SERVICE

Vous pourrez travailler de la même façon, que vous ayez installé le Toolkit CCR / DSS ou bien Microsoft Robotics Studio (cf. Programmez! 120). Dans les deux cas vous disposez d'un plugin pour Visual

Studio 2008. Nul besoin de vous ruiner, la version Express gratuite suffit :) En théorie tout langage .Net peut servir pour coder un service, mais l'emploi de C# est tellement naturel que nous n'allons pas nous en priver :) Vous trouverez les sources de tous nos exemples sur notre site. Sous Visual Studio 2008, créez un nouveau projet. Vous trouverez une sous-catégorie 'Microsoft DSS' sous la catégorie C#. Sélectionnez-là, puis sélectionnez 'DSS Service (2.0)' comme type de projet. Admettons que nous baptisons notre projet 'BasicProgrammez'. Validez votre choix, puis validez la configuration par défaut présentée par la boîte qui va suivre.

Ceci fait, vous vous trouvez dans votre projet qui comporte trois fichiers. Parmi eux se trouve un fichier XML qui sert au déploiement et à la gestion du service par le runtime. Nous ne nous y intéresserons pas aujourd'hui. Nous avons encore deux fichiers de code C# automatiquement générés. Le premier, BasicProgrammez.cs contient toute la logique du service. Bonne nouvelle, nous n'avons pas besoin d'y toucher. Le second fichier, BasicProgrammez Types.cs, contient, comme le nom l'indique des types :) Concrètement, ce sont des classes qui sont instanciées par le code du premier fichier lors du démarrage du service. Les services DSS étant orientés état, il nous suffit de nous intéresser à cet état pour personnaliser le service. Dans BasicProgrammezTypes.cs, vous trouvez:

```
[DataContract]
public class BasicProgrammezState
{
}
```

Notez la présence de l'attribut. DSS use et abuse des attributs. Si vous n'êtes pas à l'aise avec eux, c'est peut être une bonne occasion de les réviser :) L'état de notre service est pour l'instant désespérement vide. Nous faisons le choix purement arbitraire que l'état de notre service soit representé par des chaînes de caractères, mais nous aurions pu choisir n'importe quoi d'autre. Notre choix nous conduit à écrire ce code:

```
[DataContract]
public class BasicProgrammezState
{
 string _magazine = "Programmez! ";
 string _conseil = "Abonnez vous! :-)";

 [DataMember]
 [DisplayName("Magazine")]
 public string Magazine
 {
   get { return _magazine; }
   set { _magazine = value; }
 }

 [DataMember]
 [DisplayName("Un bon conseil")]
 public string Conseil
 {
   get { return _conseil; }
   set { _conseil = value; }
 }
}
```


Alter Way

Alter Way, intégrateur Open Source de référence, couvre les besoins de l'ensemble du système d'information avec ses quatre offres

CONSULTING

Conseil en architecture
Choix d'outils
Industrialisation
Best practices
Audits

SOLUTIONS

SOLUTIONS APPLICATIVES
Applications Web, ECM, E-commerce, Intégration de données, Business intelligence, Développements spécifiques

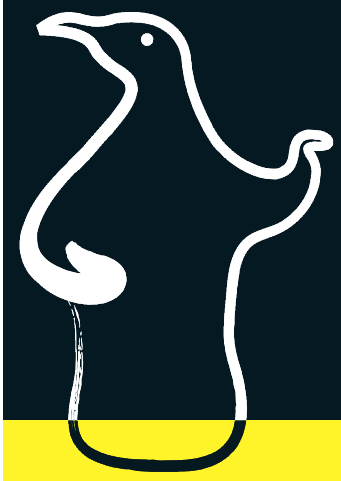
SOLUTIONS D'INFRASTRUCTURE
Supervision, Serveur de messagerie, Serveurs d'infrastructure, Bases de données, Management IP, Filtrage et partage des connexions Internet, Sécurité

FORMATION

Un catalogue de plus de 60 formations
Des programmes de certifications : PHP, MySQL, PostgreSQL, Java
1200 personnes formées par an
Près d'une centaine d'intervenants sur tous les sujets dont quinze internes certifiés
Des partenariats forts : SUN, MySql, Talend, Ingres...

HOSTING

Hébergement à haute disponibilité 24/7
Exploitation et infogérance
Solutions SaaS
Solutions de stockage
Solutions d'e-commerce
Des solutions techniques performantes et stables



TECHNOLOGIES
PHP / Python / Java / ECM
JasperSoft / Talend / Bases de données
Open-Xchange...

www.alterway.fr

OPEN SOURCE

Le libre, autrement

Alter Way

Nous avons au final une classe C# classique. Les deux membres _magazine et _conseil décrivent l'état du service. A côté de cela, nous avons défini deux propriétés pour l'accès aux membres. Ainsi, un autre service peut récupérer ou modifier l'état du nôtre. Si nos propriétés étaient en lecture seule, notre service ne pourrait voir son état modifié par un autre. Remarquez bien la présence des attributs DataMember. Cette présence est requise. Le runtime de DSS fait un intense usage de la réflexion et c'est par ce moyen qu'il verra nos "propriétés d'état".

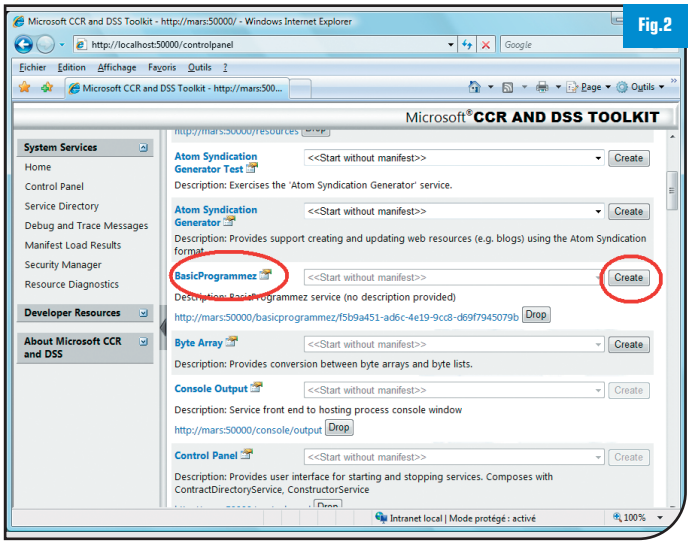
Cependant, ces propriétés doivent pouvoir être distinguées d'autres éventuelles propriétés 'banales' de la classe. C'est pourquoi le runtime recherche cet attribut particulier. Compilez le projet avec la touche F7. **Remarque :** la combinaion Ctrl-F5 aurait pour effet de lancer le runtime DSS et de démarrer le service, mais nous n'en avons pas besoin pour nos manipulations. Fermez simplement la console Windows si vous avez pressé Ctrl-F5 par habitude. Cela arrêtera le service et le runtime. La compilation entraîne le déploiement automatique du service sur la machine locale. Concrètement, cela signifie que les librairies et autres proxy sont copiés dans le sous-répertoire 'bin' du répertoire d'installation de DSS.

3 ADMINISTRER UN SERVICE VIA L'INTERFACE WEB

Démarrons manuellement le runtime DSS. Pour cela, ouvrez une console dans le sous- répertoire bin sus-mentionné et donnez:

```
dsshost /p:50000
```

La valeur 50000 étant le numéro de port au sens socket du terme.



Notre service DSS vu depuis la console d'administration Web.

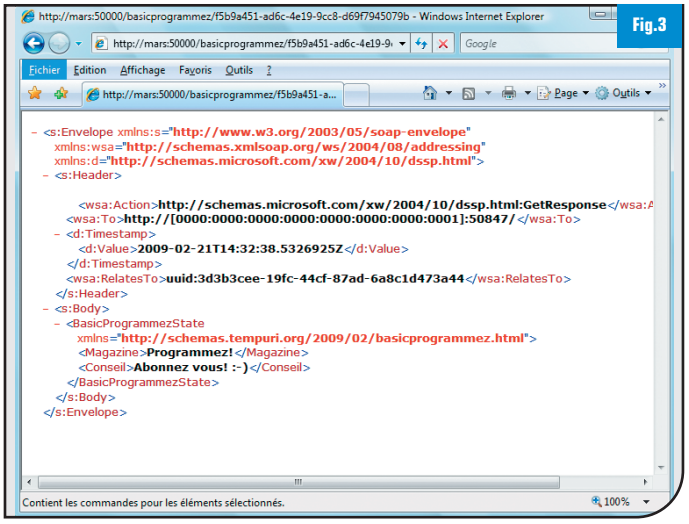
Nous pouvons donc accéder maintenant à l'interface d'administration web de DSS depuis un navigateur, en donnant l'URL

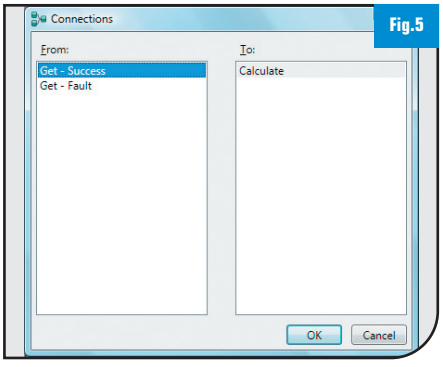
```
http://localhost:50000
```

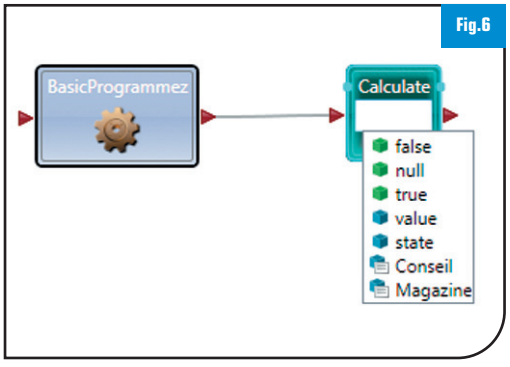
Une fois dans cette interface, cliquez sur 'Control Panel' dans la partie gauche. La partie droite affichera alors tous les services installés sur la machine locale. Comme illustré [Fig.2], vous y trouverez notre service BasicProgrammez. Un clic sur son bouton 'Create' le démarrera. A ce moment, vous devrez peut-être recharger la page dans le navigateur pour constater le démarrage. Si alors vous cliquez sur le lien qui est apparu dans la page sous le nom du service, vous arriverez à la page de rendu par défaut qui est une page XML, comme le montre la capture ci-dessous [Fig.3]. Bien entendu il est possible de personnaliser entièrement ce rendu, ce que nous ne ferons pas aujourd'hui. En revanche, constatons avec satisfaction que nos deux variables d'état apparaissent bien dans le rendu. Cliquez maintenant sur le bouton drop associé au service pour arrêter celui-ci et stoppez également le runtime DSS sur votre machine.

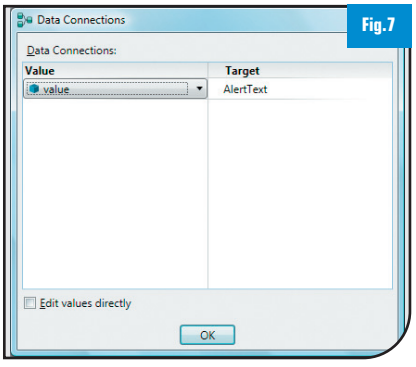
4 PREMIER CONTACT AVEC VISUAL PROGRAMMING LANGUAGE

A ce stade, nous pourrions nous lancer dans l'écriture de code pour récupérer l'état de notre service. Mais il est bien plus amusant de jouer avec Visual Programming Language (VPL) qui permet de faire interagir des services sans écrire une ligne de code! Lancez VPL, vous arrivez dans un IDE. La colonne de gauche présente deux parties. En haut se trouvent des activités. Voyons-les comme des petits bouts de codes cachés derrières des icônes. En bas, ce sont les services déployés sur notre machine (la documentation explique









comment travailler aussi avec d'autres machines, ou nœuds de réseau. Le lecteur intéressé pourra s'y reporter). Dans la liste figure le service BasicProgrammez parmi d'autres proposés par le runtime. La partie centrale attend quant à elle des éléments de la colonne de gauche, éléments qu'on y amène par glisser- déposer.

5 UN EXEMPLE BASIQUE

Sans plus attendre, glissez dans la partie centrale notre service BasicProgrammez, le service SimpleDialog et l'activité Calculate, comme illustré [Fig.4]. Nous sommes ainsi en train de construire une application composite qui interrogera l'état de notre service, fera la somme des deux chaînes constituant cet état et affichera le résultat dans une boîte de dialogue. Les symboles des composants comportent des flèches entrantes et sortantes, destinées à recevoir des connexions qui définiront le flux d'information (workflow) de l'application. Etablissons une première connexion entre le service BasicProgrammez et l'activité Calculate. Pour cela, il suffit de cliquer sur la flèche sortante du service et de glisser le pointeur jusqu'à la flèche entrante de l'activité. Au moment où vous relâcherez le bouton de la souris, un dialogue s'ouvrira pour vous demander de préciser la nature de la connexion. [Fig.5] Notre service n'a qu'une fonctionnalité *Get* qui permet d'obtenir son état. La liste de gauche dans le dialogue nous dit *Get - Success* ou *Get - Fault.* En effet, nous avons évoqué le mois dernier que CCR est doté d'un mécanisme sophistiqué de gestion des erreurs. Il en va de même des services DSS. Nous n'entrerons pas dans les détails aujourd'hui. Nous supposons que tout se passera bien et nous connectons donc simplement *Get - Success* avec *Calculate* en cliquant sur chacun de ces éléments dans le dialogue puis en validant par le bouton Ok. Ensuite, un clic dans la zone d'édition de l'activité montre les valeurs possibles pour le calcul. [Fig.6] Ces valeurs sont issues du workflow. Si nous retrouvons ici nos propriétés Magazine et Conseil, rien n'interdit d'autres valeurs arbitraires supplémentaires. Illustrons cela en saisissant dans la zone d'édition

```
Magazine + Conseil + " pour 2 ans :)"
```

Puis connectez de la même manière l'activité au service SimpleDialog. Dans la boîte de dialogue de connexion, on connectera CalculatedResult à AlertDialog. A ce moment, un deuxième dialogue apparaîtra, vous demandant de spécifier quelle valeur prendre pour l'affichage du texte dans le dialogue. Sélectionnez 'value' comme illustré [Fig.7]. C'est terminé, il ne reste plus qu'à lancer l'application pour admirer le résultat. Rappel : pour que tout fonctionne bien, le runtime DSS ne doit pas tourner sur la machine hôte au moment où on lance l'application avec VPL.

6 UNE APPLICATION PLUS COMPLEXE

Concevons maintenant une application un peu plus complexe, [Fig.8] ce qui nous fera faire connaissance avec des fonctionnalités importantes de VPL. Tout d'abord nous avons éclaté notre service BasicProgrammez en deux services distincts, ServiceProgrammez et ServiceAbonnement. Le lecteur trouvera le code sur notre site. Décrivons maintenant le fonctionnement de notre application: nous injectons une phrase dans un dialogue (service SimpleDialog) au début du workflow. L'utilisateur clique sur ce dialogue, une autre valeur est injectée dans le workflow, ce qui enclenche un timer. Lorsque le timer arrive à échéance et se déclenche, il notifie nos deux services qui délivrent alors leur état dans le workflow. Ces deux états sont collectés dans une activité *Join*. Il est alors possible de les ajouter l'un à l'autre dans une activité *Calculate* et le résultat du calcul est présenté dans un deuxième dialogue.

Globalement, tout est semblable à notre premier exemple, hormis les points particuliers importants suivants: Nous avons deux icônes de timer, mais elles ne représentent qu'un seul et même service. Lorsque l'on glisse deux fois (ou plus) la même icône dans la zone de conception, VPL demande si l'on veut avoir une nouvelle instance du service ou bien travailler avec celle déjà présente. Dans le cas du timer, nous avons utilisé l'instance déjà présente. L'apparition de la deuxième icône n'a qu'un seul but: nous permettre de connecter la notification de déclenchement du timer aux services que nous voulons voir entrer en action à ce moment. On remarque à ce stade que le workflow se dédouble, ce qui s'obtient tout naturellement en établissant les connexions du timer à chacun des services. Ensuite, à la "sortie" des services, le workflow redevient unique grâce à l'activité *Join* dont c'est le rôle, et qui n'a pas comme rôle de synchroniser les threads, comme sont nom pourrait le laisser à penser. Enfin nous retombons sur l'icône du service SimpleDialog. Mais cette fois, il s'agit d'une autre instance du service SimpleDialog. L'essentiel est dit. Laissez libre cours à votre imagination et composez vos applications :)

■ Frédéric **Mazué**
fmazue@programmez.com

Fig.4

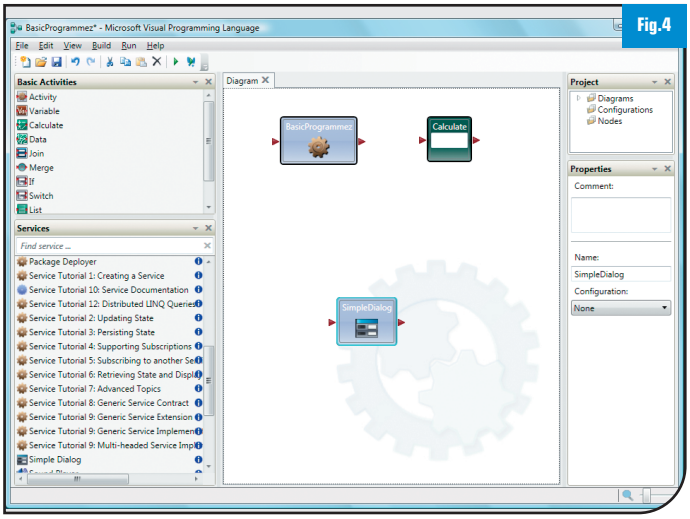
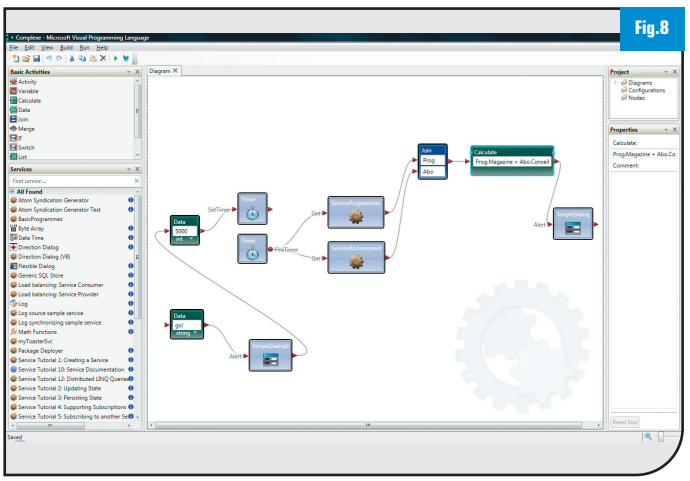
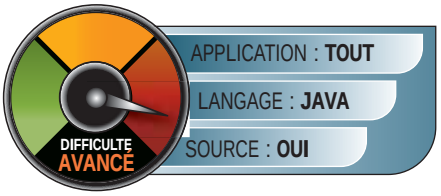


Fig.8



Clojure, un Lisp pour la JVM

Après Jython et Groovy la JVM sert de support à un autre langage de script qui est bien plus qu'un langage de script. C'est un Lisp, il est particulièrement doué pour la programmation concurrente, il s'appelle Clojure et est à découvrir absolument.



Sun a rapidement présenté Java comme le langage de toutes les situations. Incontestablement, Java est un langage polyvalent. Ses créateurs avaient-ils toutefois prévu qu'il servirait à son tour pour créer d'autres langages ? Ce n'est pas certain. Java tout polyvalent qu'il est, ne peut pas tout. Ainsi ce n'est pas un langage de script. Mais il apporte une sécurité indéniable et une richesse de bibliothèques non moins indéniable. Un constat qui a sans doute présidé à la création de Jython, une implémentation du langage Python en Java. Jython est plein de qualités, mais il faut reconnaître qu'il peine à atteindre la popularité qu'il mérite. Plus récemment, c'est le langage de script Groovy qui est apparu sur la scène Java. Contrairement à Jython, Groovy est un langage entièrement nouveau. Programmez! a été parmi les tous premiers à vous le présenter (cf. programmez 91). Groovy, lui, a vu rapidement ses qualités reconnues, ce qui, au passage, montre que Java est une solution viable pour la création de langages de script. Nous allons maintenant nous intéresser à un autre langage construit sur Java. Il est à la fois nouveau et ancien, il est à la fois objet et fonctionnel, et il est à la fois langage de script et langage compilé. Un langage aussi étonnant ne peut être … qu'un Lisp. Il s'appelle Clojure et il est génial. Ses qualités seront-elles reconnues ? L'avenir nous le dira, mais en attendant ne nous privons pas du plaisir de sa découverte.

1 AVANT CLOJURE ÉTAIT LISP

Lisp… cette espèce de dinosaure du monde de l'informatique et qui refuse obstinément de disparaître. Deuxième vrai langage de programmation (après Fortran), il ne laisse personne indifférent. Il y a les inconditionnels et ceux qui le haïssent, notamment en raison de sa notation préfixée et de ses parenthèses. Aux membres de la seconde catégorie, nous suggérons de (re)découvrir Lisp via Clojure, car Lisp a des possibilités et une souplesse que finalement aucun langage n'a jamais égalées. Et si le vieux Lisp a survécu à toutes les modes, s'il est utilisé dans des milieux très exigeants comme la finance ou les télécommunications, ce n'est pas sans raison. Voici qu'avec Clojure nous avons la possibilité de faire interagir du code Lisp avec du code Java et réciproquement, ce qui pourrait donner une nouvelle jeunesse à ce vénérable langage. Cet article étant destiné à Clojure, il ne peut être question d'y apprendre Lisp, toutefois afin que le lecteur intéressé puisse suivre, donnons-en quelques notions très succinctes.

2 NOTIONS DE LISP

En Lisp, terme qui vient de LISt Processing, tout est liste, le code, comme les données. Tout élément de programme est délimité par des parenthèses et s'appelle une form en jargon Lisp. Voici donc une simple liste, constituée de deux entiers et une chaîne.

```
(1 "deux" 3)
```

Cette syntaxe très simple implique que Lisp est un langage dynamiquement typé. La form ci-dessus n'est que donnée. Quand le premier élément d'une liste peut être appliqué au reste de la liste nous avons du code. Exemple:

```
(+ 1 2 3)
```

L'opérateur d'addition s'applique ici et trois entiers de la liste et la form s'évaluent à 6. C'est cette application du premier élément d'une liste à son reste qui implique que Lisp est un langage à notation préfixée. Bien sûr les forms peuvent être imbriquées. Exemple:

```
(+ 1 2 3 (+ 4 5))
```

S'évaluera à 15. Il existe aussi des forms plus complexes dites spéciales, mais fondamentalement le principe reste le même. Les détracteurs de Lisp lui reprochent d'être illisible à cause de ses parenthèses. En fait l'erreur est de vouloir lire les parenthèses. Le code Lisp se lit en fonction de son indentation conventionnelle et dès que l'on est un peu habitué, avec un bon éditeur (il existe des modes Clojure pour Emacs et Vim, forcément… ;) les parenthèses ne se voient même plus :) Abordons maintenant Clojure.

3 INSTALLATION ET PREMIER PAS

On téléchargera l'archive à http://code.google.com/p/clojure/downloads/list Dans celle-ci, on trouvera le fichier clojure.jar, dont on appréciera la légèreté et qui contient Clojure à lui seul. Cependant, notamment pour faciliter la saisie dans la RELP (voir plus loin) on utilisera Clojure avec la librairie Java JLine que l'on téléchargera à http://jline.sourceforge.net/ et qui vient elle aussi sous la forme d'un fichier jar. JLine permet des saisies en console plus évoluées que ce que permet nativement Java. JLine est en quelque sorte l'équivalent de l'outil GNU readline pour Java. Ensuite on lancera Clojure par un petit script. Voici celui que votre serviteur utilise sous Windows et que vous pourrez adapter à vos besoins.

```
@echo off
set CLOJURE_DIR=C:\javalibs\Clojure
set CLOJURE_JAR=%CLOJURE_DIR%\clojure.jar

IF (%1)==() (
        java -cp .;.\classes;%CLOJURE_DIR%\jline-0.9.94.jar;%
CLOJURE_JAR% jline.ConsoleRunner clojure.lang.Repl
    ) ELSE (
      java -cp .;.\classes;%CLOJURE_JAR% clojure.lang.Script %1 -- %*
    )
```

Si ce script ne reçoit pas de paramètre, la classe clojure.lang.script amène dans la REPL dont dispose tout Lisp qui se respecte.

4 LA RELP OU READ EVAL PRINT LOOP

La boucle d'évaluation est à Lisp ce que sont les interpréteurs interactifs d'autres langages de script comme Python ou Ruby. On peut y saisir n'importe quelle form, simple ou complexe pour évaluation:

```
user=> (+ 1 2 3)
6
```

La form (1 "deux" 3) peut-elle être évaluée ? Non, car son premier élément ne peut s'appliquer au reste. Mais on peut la saisir ainsi:

```
'(1 "deux" 3)
```

la quote (') indiquant à l'interpréteur de ne pas évaluer la forme.

```
user=> '(1 "deux" 3)
(1 "deux" 3)
```

Cette saisie n'est pas très utile en elle-même, mais elle nous indique indirectement comment attribuer une liste à une variable sans évaluer la liste:

```
user=> (def ma-liste '(1 "deux" 3))
#'user/ma-liste
user=> (println ma-liste)
(1 deux 3)
```

Bien sûr, il est hors de question de saisir ainsi un programme entier dans l'interpréteur. On demandera plutôt à celui-ci de charger un fichier et d'évaluer son contenu de cette façon:

```
user=> (load-file "demo-addition.clj")
30
70
120
```

Voici le contenu du fichier demo-addition.clj (disponible sur notre site) qui contient deux définitions de fonction d'addition:

```
(defn addition-v1 [x y] (+ x y))

(defn addition-v2
 ([x y] (+ x y))
 ([x y z] (+ x y z)))

(println (addition-v1 10 20))
(println (addition-v2 30 40))
(println (addition-v2 30 40 50))
```

Cet exemple nous apprend comment définir des fonctions en Clojure. Nous voyons que les paramètres sont passés via un tableau (un Vector en fait) et surtout, la fonction addition-v2 nous montre une spécificité de Clojure très intéressante: la surcharge d'arité.

5 CLOJURE ET LES CLASSES JAVA

Pour travailler avec les classes et objet Java, Clojure introduit l'opérateur point (.) ainsi qu'un sucre syntaxique. Supposons que nous voulions faire en Clojure l'équivalent de :

```
frame.setSize( 300, 300);
```

expression dans laquelle frame est une instance de la classe JFrame. La syntaxe en Clojure est

```
(. setSize frame 300 300)
```

ou plus intuitivement avec le sucre syntaxique

```
(.setSize frame 300 300)
```

Sans rentrer plus dans les détails, donnons tout de suite un petit programme Swing (demo-swiwg.clj sur notre site). Nous ouvrons une fenêtre qui contient un label et un bouton. Un clic sur le bouton modifie le texte du label. [Fig.1]

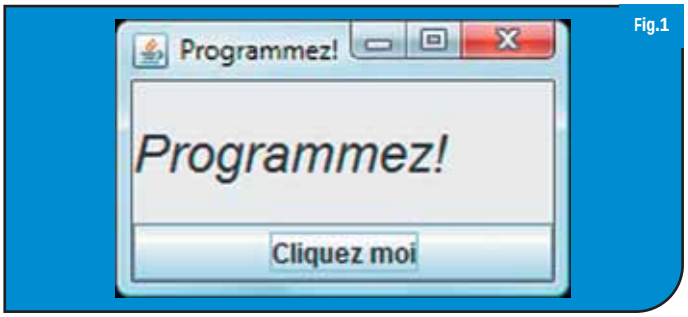


Fig.1

Il est très facile de manipuler les classes Swing avec Clojure.

```
(import '(javax.swing JFrame JButton JLabel)
   '(java.awt.event ActionListener)
    '(java.awt BorderLayout Font))

(defn programmez []
  (let [jframe (JFrame. "Programmez!")
    programmez-jlabel (JLabel. "Programmez!")
    conseil-jbutton (JButton. "Cliquez moi")
  (.addActionListener conseil-jbutton
    (proxy [ActionListener] []
      (actionPerformed [evt]
        (.setText programmez-jlabel "Abonnez vous!! :-)")
        (.setSize jframe 250 80))))
        (.setFont programmez-jlabel (Font. "Times" Font/ITALIC 24))
  (doto jframe
    (.setLayout (BorderLayout.))
    (.add programmez-jlabel BorderLayout/CENTER)
    (.add conseil-jbutton BorderLayout/SOUTH)
    (.pack)
    ;; ATTENTION, ci dessous, termine Clojure :)
    ;;(.setDefaultCloseOperation JFrame/EXIT_ON_CLOSE)
    (.setVisible true))))

;; C'est parti!
(programmez)
```

Saisir (load-file "demo-swing.clj") dans le REPL pour lancer le programme. Finalement, tout cela est assez intuitif. A ceux qui ne seraient pas familiers avec les idiomes Lisp, signalons les forms spéciales *let* et *doto* (Oui les puristes, vous avez raison : ce sont en

fait des macros :). La première, *let,* définit un environnement lexical contenant des variables utilisées par des fonctions invoquées au sein de cet environnement. La seconde, *doto,* passe son argument (ici jframe) en guise de paramètres à toutes les invocations de fonction qu'elle contient.

6 QUAND CLOJURE DEVIENT JAVA

Sans doute, même si on apprécie beaucoup Lisp, n'est-il pas systématiquement pertinent de coder en Clojure. On pourra alors se limiter à profiter de l'expressivité de celui-ci pour traiter des parties ardues auxquelles la programmation fonctionnelle se prête mieux ou pour du code concurrent, car Clojure surclasse très nettement Java pour cela. Une fois le code Clojure écrit, il est possible de le compiler en classes Java pour exécution directe avec la JVM. La procédure est simple si l'on est méticuleux. Clojure respecte les conventions Java qui veulent que les classes soient placées dans des arborescences en correspondances avec les packages. Clojure étend ceci aux sources comme le font d'ailleurs la plupart des IDE Java. Lorsqu'il génère des classes java, Clojure les dépose dans une même arborescence, mais dans un répertoire 'classes' qui doit exister au préalable et être intégré dans le CLASSPATH. De même le répertoire courant étant a priori à la racine de l'arborescence de sources doit lui aussi être intégré au CLASSPATH. Voilà pourquoi vous trouvez -cp .;.\classes; dans notre fichier .bat de lancement de Clojure et ce, pour les deux modes de lancement. Ceci posé, reprenons notre exemple Swing et compilons-le en une classe Java de nom ClojureSwing. D'abord nous modifions un peu le source, rebaptisé pour l'occasion ClojureSwing.clj (fichier disponible sur notre site) et placé dans un répertoire 'fred' puisque nous voulons que la classe appartienne à un package 'fred'. Les imports du début du code sont modifiés pour que tout se passe dans un namespace, pendant Clojure du package Java :

```
(ns fred.ClojureSwing
 (:gen-class)

 (:import (javax.swing JFrame JButton JLabel)
   (java.awt.event ActionListener)
   (java.awt BorderLayout Font)))
```

La fonction programmez reste identique. Au lieu d'appeler celle-ci en fin de fichier comme dans l'exemple Clojure pur, Nous définissons une fonction *main* pour appeler à son tour la fonction programmez. Cette fonction deviendra la méthode statique *main* indispensable à la classe Java générée :

```
(defn -main []
 (programmez))
```

Pour compiler, placez-vous à la racine de l'arborescence des sources. Lancez Clojure et dans la REPL saisissez :

```
(compile 'fred.ClojureSwing)
```

C'est tout :) Pour lancez la démo ClojureSwing, en supposant que le jar de Clojure soit à la racine de l'arborescence des classes, il suffit de faire :

```
java -cp .;./clojure.jar fred.ClojureSwing
```

7 CLOJURE CHAMPION DE LA CONCURRENCE

Faire le tour d'un langage comme Clojure est impossible en un seul article, mais si nous ne devions mentionner qu'un seul de ses apports, ce serait celui de sa remarquable approche de la concurrence. En effet Clojure permet d'écrire du code parallélisé sans manipulation explicite des verrous et autres sémaphores, tout en étant prémuni des *race condition.* A l'instar des langages fonctionnels dernier cri (votre serviteur pense à Haskell), Clojure implémente un mécanisme STM ou Software Transaction Memory qui permet le changement d'état de variables partagées entre différents threads de manière synchrone et coordonnée. Ne nous lançons pas dans la théorie compliquée. Prenons plutôt un exemple qui parlera de lui-même. Supposons que nous ayons deux threads incrémentant chacun 100 fois une valeur partagée et rangeant ensuite celle-ci dans un vecteur. Quand les deux threads ont terminé leur travail, nous avons un vecteur contenant deux cents valeurs consécutives. Ou du moins est-ce ce que nous voulons avoir... :) Codons ceci en Java de manière naturelle, c'est-à-dire sans aucune précaution:

```
package fred;

import java.util.Vector;
import java.util.concurrent.*;

class ThreadForExecutor implements Runnable {
 private Vector v;
 private int id;

 ThreadForExecutor(Vector v, int id) {
   this.v = v;
   this.id = id;
 }

 public void run() {
   for(int i=0; i<100; i++) {
     id++;
     v.add(id);
   }
 }
}

public class Conflit {
 public static void main(String[] args) {
   Vector v = new Vector();
   int id =0;

 ExecutorService es =
   Executors.newFixedThreadPool(2);
 es.execute(new ThreadForExecutor(v, id));
 es.execute(new ThreadForExecutor(v, id));
 es.shutdown();

 try {
   es.awaitTermination(1000,
     TimeUnit.MILLISECONDS);
 }
 catch (InterruptedException ex) {
```

```
  }

  System.out.println("Taille du Vector: " + v.size());
    for(int i=0; i<v.size(); i++) {
      System.out.print("" + v.elementAt(i) + " ")
    }
  }
}
```

Comme nous n'avons pris aucune précaution, le résultat est évidemment catastrophique comme l'illustre la capture ci-dessous. [Fig.2] Notre vecteur semble plus rempli aléatoirement qu'autre chose, et il contient des doublons de surcroît. Bien sûr, avec l'arsenal de classe du package java.util.concurrence, il est possible de résoudre le problème en Java. Mais il faut réfléchir et éventuellement se tromper une fois ou deux avant d'y arriver :) C'est là que Clojure montre sa supériorité. Nous codons la même chose, avec lui, également de manière naturelle (fichier demo-concurrence.clj sur notre site).

```
(import '(java.util.concurrent Executors TimeUnit))

(let [tableau (ref []) ; Une référence sur un tableau vide
  valeur (ref 0)   ; Une référence sur une valeur initialisée à 0
  nb_threads 2      ; 2 threads
  n 100              ; 100 itérations
  pool (Executors/newFixedThreadPool nb_threads)]

  (dotimes [th nb_threads]
    (.submit pool (fn []
      (dotimes [i n]
        (let [nouvelle-valeur (dosync (alter valeur inc))]
          (dosync (commute tableau conj nouvelle-valeur)))))
                    nil))
  (.shutdown pool)
  (.awaitTermination pool 1 TimeUnit/MINUTES)

  (println "Nombre de valeurs :" (count @tableau))
  (println "Contenu du tableau:")
  (println @tableau))
```

apprécions d'abord la concision du code, par rapport au code Java donné plus haut et qui ne contient pourtant pas de synchronisation, rappelons-le. Ensuite, regardons ce que nous avons fait. Les variables tableau et valeur sont définies comme étant des références, ce qui les rend aptes à être partagées. Ensuite les accès à ces variables se limitent à

```
(dosync (alter valeur inc))
```

pour la valeur. La macro *dosync* démarre une transaction dans laquelle la valeur est modifiée (alter) par incrémentation (inc). Ensuite, pour le tableau nous avons

```
(dosync (commute tableau conj nouvelle-valeur))
```

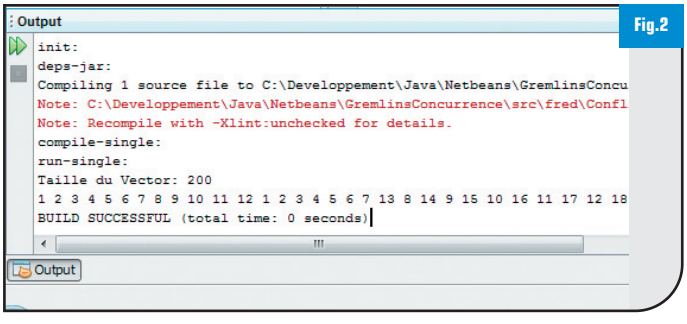
Nous avons là encore *dosync,* donc une transaction dans laquelle le tableau est modifié (commute) par ajout (conj) d'une nouvelle valeur. Et cette fois cela fonctionne, comme l'illustre la capture ci-dessous. [Fig.3]. Bien sûr, nous n'avons pas rien fait pour obtenir ce résultat. Nous avons défini des références et lancé des transactions. Mais ceci est fait de manière aisée et naturelle, sans manipulation directe d'objet de synchronisation. Clojure apporte donc un confort de programmation sans comparaison avec Java (et beaucoup d'autres langages!) dans ce domaine.

8 CONCLUSION PROVISOIRE

Nous n'avons fait qu'effleurer Clojure. Mais nous espérons vous avoir fait ressentir qu'il s'agit là d'un très grand langage. Nous aurions pu parler de l'évaluation paresseuse qu'il supporte à l'instar des plus modernes (encore Haskell, cf. programmez! 104 et 105 ;). Nous aurions pu parler de son approche objet à base de multi-méthodes, comme CLOS (Common Lisp Object System). Nous le ferons peut être une prochaine fois. Mais nous savons d'ores et déjà, parce qu'il prend le meilleur de Java et qu'il est un Lisp, que Clojure mérite un grand succès

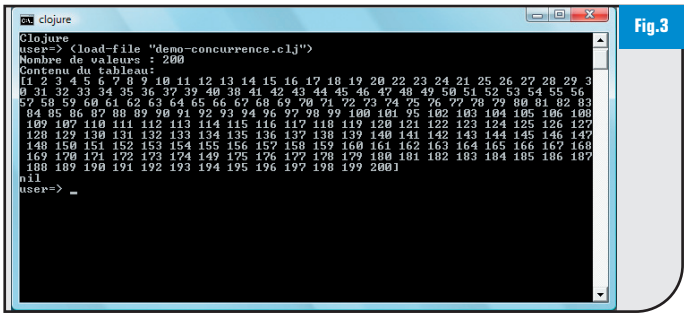
■ Frédéric **Mazué**
fmazue@programmez.com

Fig.2



Mise en évidence d'un accès conflictuel à une variable partagée en Java

Fig.3



Avec Clojure, l'accès concurrent à des variables partagées est sans surprise.

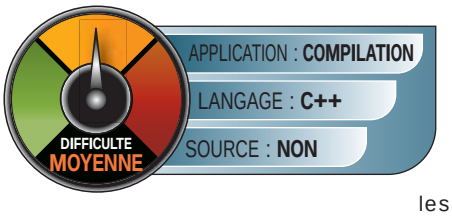
PROCHAIN NUMÉRO N°122
septembre 2009,
parution 1er septembre

✔ **Dossier WEB**
Choisir son **hébergeur.**
Les Techniques de **référencement.**
Monétiser son site.

✔ **.Net 4.0**
Plongée au cœur de la CLR4

Réalisation d'un compilateur de Pascal simplifié en C++

Tout développeur est amené à utiliser un compilateur. Toutefois, peu nombreux sont ceux qui savent réellement comment fonctionne celui-ci. Cet article et les suivants permettront donc de présenter les différentes étapes de compilation, et d'expliquer comment créer un programme pour réaliser l'ensemble de ces étapes.



QU'EST-CE QU'UN COMPILATEUR ?

Avant la venue des langages de programmation, les programmeurs devaient connaître par cœur l'architecture de la machine sur laquelle ils travaillaient, et "parler" directement à celle-ci à l'aide d'un langage binaire. Très rapidement, a été inventé un langage utilisant des raccourcis textuels : l'assembleur. Ce nouveau langage présentait plusieurs avantages: tout d'abord, il était plus facile de lire du texte utilisant des mots sensés, mais surtout, une vérification basique était faite lors d'une erreur de saisie. Toutefois ce langage avait de nombreux défauts. Il imposait peu de contraintes au développeur, ce qui pouvait être avantageux, mais était peu pratique pour la productivité et la robustesse du code. La vérification de type n'existant pas, de nombreux bugs en découlaient, et ceux-ci étaient particulièrement difficiles à détecter. Enfin, le niveau d'abstraction était faible, le langage restant très proche du langage machine. Pour pallier cela, les langages de programmation dits de "haut niveau" ont été inventés. La principale caractéristique de ceux-ci était un niveau d'abstraction élevé. C'est-à-dire que le langage ressemblait plus à un langage humain qu'à des instructions machines.

Ainsi, la programmation devenait moins ésotérique, et il était plus facile pour un néophyte de se lancer dans l'apprentissage de ce domaine. De plus, ces langages imposaient des "contraintes fictives", comme un typage fort des données, qui permettait une meilleure robustesse, ou encore la prise en charge automatique de certains concepts. Par exemple en Pascal, il est possible de concaténer deux chaînes de caractères à l'aide de l'opérateur +. En assembleur, il aurait fallu allouer de la mémoire pour la nouvelle chaîne, et recopier un à un chacun des caractères de ces deux chaînes. L'avantage d'un compilateur est indéniable, et mis à part dans quelques domaines spécifiques, la très grande majorité des développeurs ne programme pas en assembleur. Beaucoup de programmeurs n'en ont même jamais vu !

COMMENT FONCTIONNE UN COMPILATEUR ?

Il existe de nombreux compilateurs. Le fonctionnement de ceux-ci étant propre à un langage, nous n'allons bien évidemment pas tous les passer en revue. Au lieu de cela, nous allons définir à quoi devra ressembler notre langage. A partir de là, nous établirons ce que notre compilateur sera capable de réaliser.

LE LANGAGE

Réaliser un compilateur est extrêmement long et peut être particulièrement difficile. Le but de ce projet étant de réaliser un compilateur basique dans un temps acceptable, nous allons utiliser un langage très simple. Le langage choisi aura une syntaxe proche du Pascal. Définissons maintenant ce que le langage sera capable de faire. Tout d'abord, celui-ci sera typé, c'est-à-dire qu'une variable ne pourra contenir qu'une valeur d'un type compatible. Trois types de base seront suffisants: entier, booléen, et chaînes de caractères. Les fonctions et la récursivité, en plus d'être pratiques, sont pédagogiquement très intéressantes à mettre en place. Incluons-les, ainsi que le mot clé return, qui permettra de quitter une fonction en renvoyant une valeur. Une fonction sera obligée de renvoyer quelque chose. Sa présence implique donc que la portée des variables devient nécessaire. Une variable pourra donc être locale ou globale. Un langage sans structures de contrôle ne serait pas très intéressant, ajoutons donc la possibilité de faire un choix (if), et de réaliser une opération en boucle (while). Enfin, dans un souci de simplification, il n'y aura pas d'ambiguïtés sur l'évaluation d'une expression puisqu'on forcera l'utilisateur à mettre des parenthèses de manière explicite. Nous ne traiterons pas les notions plus compliquées, comme les tableaux, la gestion de la mémoire, les pointeurs ou encore la notion d'objet. Pour finir, incluons trois *builtins,* c'est-à-dire trois fonctions préexistantes, **read, print** et **exit,** qui permettront respectivement de récupérer une information entrée par l'utilisateur, d'afficher une valeur à l'écran et de quitter le programme avant la fin de son exécution.

Un exemple d'un programme qui demande une valeur à l'utilisateur et affiche la factorielle de celle-ci:

```
var a : integer;
var s : string;
function facto (nb : integer) : integer;
var res : integer;
begin
   if nb <= 1 then
   begin
      return 1;
   end
   res = nb * facto(nb - 1);
   return res;
end

begin
```

```
    print("Entrez un nombre : ");
    read(a);
    res = facto(a);
    s = "La factorielle de ";
    print(s);
    print(a);
    print(" est ");
    print(res);
    print("\n");
    exit(0);
    print("Ignoré");
end
```

LA GRAMMAIRE DU LANGAGE

Maintenant que nous sommes fixés sur les fonctionnalités de notre langage, écrivons-en la grammaire. Il faut savoir que tout langage de programmation, comme un véritable langage, possède des règles d'écriture. On ne peut pas écrire n'importe quoi, l'ordre des "mots" du langage à une importance. C'est là qu'intervient cette fameuse grammaire. Elle permet de spécifier de manière claire et exhaustive, comment agencer les mots du langage entre eux. Il est d'usage lors de son écriture, d'utiliser un formalisme universel. Celui que l'on utilisera sera la notation BNF (Backus-Naur Form).

Si vous n'avez pas de connaissance dans le domaine de la théorie des langages, il y a de fortes chances que la lecture de cette grammaire soit un peu compliquée. Voici donc quelques règles basiques pour en comprendre le sens:

Un terminal est un mot du langage existant, et qui comme son nom l'indique, ne renvoie pas vers une autre règle.

Un non-terminal est une règle, qui n'est pas présente explicitement dans le langage, et qui est la composée d'autres règles. On l'entoure des signes "<" et ">" pour la différencier des terminaux.

Une règle est précédée par le signe "::=" qui indique comment est constituée celle-ci.

Le "|" représente un "OU", c'est-à-dire un choix entre plusieurs règles. Une règle ou un terminal entre "[" et "]" veut dire que cet élément est facultatif.

Un petit exemple pour illustrer cela: la règle de grammaire d'un *if*, en C++ serait :

```
<if> ::= if ( <expression> ) { <expression> } [ else { <expression>} ]
que l'on pourrait aussi écrire:
<if> ::= if ( <expression> ) { <expression> }
      | if ( <expression> ) { <expression> } else { <expression>}
```

Voici donc celle de notre langage :

```
<program> ::= [<declarations>] [<funcs>] [<compound_instruction>]
<declarations> := <declaration>
                | <declaration> <declarations>

<declaration> ::= var <declaration_body> ;

<declaration_body> ::= <ids> : <type>

<ids> ::= <id>
        | <id> , <ids>
```

```
<funcs> ::= <func>
          | <func> <funcs>

<func> ::= <header_func> <compound_instruction>
        | <header_func> <declarations> <compound_instruction>

<header_func> ::= function <idfunc> ( [<arguments>] ) : <type> ;

<arguments> ::= <argument>
              | <argument> ; <arguments>

<argument> ::= <declaration_body>
             | <declaration_body> ; <argument>

<compound_instruction> ::= begin end
                         | begin <instructions> <end>

<instructions> ::= <instruction>
                 | <instruction> <instructions>

<instruction> ::= <affect> ;
                | <call_func> ;
                | <compound_instruction>
                | <if>
                | <while>
                | <return> ;
                | <exit> ;
                | <builtin_print> ;
                | <builtin_read> ;

<affect> ::= <id> = <expression>

<return> ::= return <expression>
<exit> ::= return <expression>

<builtin_print> ::= print ( <expression> )

<builtin_read> ::= read ( <id> )

<if> ::= if <expression> then <compound_instruction>
       | if <expression> then <compound_instruction> else <com
pound_instruction>

<while> ::= while <cond> do <compound_instruction>

<call_func> ::= <idfunc> ( )
              | <idfunc> ( <expressions> )

<expressions> ::= <expression>
                | <expression>, <expressions>
```

<expression> ::= <operation> / * Ce noeud peut paraître inutile, mais il est présent pour une future extension du langage, que je ne présenterai pas dans cet article * /

```
<operation> ::= <factor> <symb> <factor>
              | <factor> /* operation sans facteur droit */
       | - <factor> /* facteur gauche = 0 */
```

```
<factor> ::= <id>
           | <call_func>
           | <number>
           | <stringexpr>
           | <boolean>
           | ( <expression> )

<id> ::= [a-zA-Z]([a-zA-Z0-9_]*[a-zA-Z0-9])?
<type> ::= integer | string | boolean
<symb> ::= > | < | >= | <= | != | == | + | - | * | / | %
<number> ::= [0-9]+
<stringexpr> ::= .*
<boolean> ::= true | false
```

LES DIFFÉRENTES ÉTAPES DE LA COMPILATION

Lorsqu'une compilation est lancée, plusieurs étapes se succèdent pour mener à la création d'un exécutable. Tout d'abord, la majorité de ces étapes sert à vérifier la cohérence du code écrit. Voici une présentation non exhaustive des différentes étapes que nous allons réaliser. Les détails techniques seront abordés lors de prochains articles dédiés à chaque fois à une phase précise.

La première étape, appelée "lexing" consiste en la lecture du fichier, et la récupération de "tokens". Un *token* est le plus petit morceau atomique défini dans une grammaire. Si l'on prend une analogie avec la langue française, un *token* représenterait un mot. Donc la phrase suivante: "Bonjour à tous" serait considérée comme valide, tandis que: "Bonjour xvcfbjnb tous" provoquerait une erreur de *lexing.* En effet, le mot "Bonjour" et le mot "tous", sont bien présents dans le dictionnaire français, mais le mot "xvcfbjnb", lui, ne l'est pas. Cette étape va donc vérifier que certains mots impossibles ne sont pas présents dans le code source. Pour provoquer une erreur de lexing dans notre langage, on pourrait écrire : "if @_@ then". Le mot "@_@" n'étant pas un "token" acceptable, le code est considéré comme non valide.

La seconde étape, appelée "parsing", va, à partir des "tokens" précédemment récupérés, vérifier que l'agencement est correct. En effet, ce n'est pas parce que l'on n'a détecté aucun "token" invalide, que ceux-ci sont forcément dans le bon ordre. Reprenons une analogie avec la langue de Molière. Si je vous dis : "Bonjour, ça va ?" vous devriez me comprendre, tandis qu'avec "va ? ça, Bonjour" la communication serait quelque peu compliquée. Et bien avec notre langage, c'est la même chose. Si l'utilisateur entre "if then true if if", alors nous sommes bien en présence d'une erreur de parsing. L'agencement de chacun de ces "tokens", ne respecte pas la grammaire que nous avons définie.

L'étape suivante est appelée "binding". Ici on va vérifier que chacune des variables et fonctions sont correctement reliées entre leurs déclarations et leurs appels. Pour résumer, cette phase se contente de détecter qu'une variable ou une fonction sont bien déclarées une et une seule fois.

Exemple :

```
var a : integer;
begin
  b = 50;
end
```

Ici on voit bien que la variable b, n'est pas déclarée. Donc le compilateur doit prévenir l'utilisateur, et refuser de créer un exécutable.
De même :

```
var b : integer;
var b : integer;
begin
  b = 50;
end
```

Ici la variable b est définie deux fois, donc c'est aussi une erreur de "binding". La même vérification est appliquée sur les fonctions, en prenant aussi en compte que le nombre d'arguments décrits, et le nombre d'arguments réellement passés, correspond.

Enfin la dernière étape de vérification, est celle appelée "type checking". Elle vérifie que le typage des expressions est correct et que chaque variable reçoit une valeur dont le type est compatible.
Par exemple:

```
var b : integer;
begin
  b = 50;
end
```

Ici b, qui est de type entier recevra une valeur entière, donc la compatibilité est assurée, il n'y a pas d'erreur.
Tandis que dans l'exemple suivant:

```
var b : integer;
begin
  b = "coucou";
end
```

La variable b qui est de type entier, reçoit une chaîne de caractères. Ce n'est évidemment pas possible, et c'est ce type de problème qui provoquera une erreur de "type checking".

Une fois toutes ces étapes validées, on considère notre code source comme étant valide. Il ne reste alors qu'à l'interpréter, ou à le convertir en assembleur. Ces deux étapes seront détaillées plus tard. Ici se termine ce premier article. J'espère que cette première approche vous aura donné envie d'aller un peu plus loin dans la compréhension de la compilation. Les articles suivant seront un peu plus techniques.

■ Axel **Berardino** - *axel.berardino@gmail.com*

Étudiant en dernière année à l'EPITA. A souvent dispensé des cours et des formations en entreprise. A l'origine, ce projet de compilateur était dédié à une formation en C++.

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

Migration d'une application asp. net vers Windows Azure

Le socle commun est Windows Azure, le « Cloud OS » de Microsoft. Windows Azure est construit sur Windows Server et Hyper V, pour apporter une infrastructure hautement disponible et adaptée à la montée en charge. On peut considérer que c'est une surcouche de Windows Server, Windows Server étant le système d'une machine et Windows Azure le système dans le nuage.

La plate-forme Azure propose un ensemble de services web (milieu [Fig.1]):

• *Live Services :* ensemble des services fournis par la plate-forme Live, utilisables via un ensemble d'API.

• *.Net Services :* 3 services existent actuellement : Access Control (fédération d'identité), Service Bus (connectivité) et Workflow services.

SQL Services : C'est un service de base de données relationnelle dans le « cloud ». La version actuelle utilise un nouveau paradigme ACE (Authority, Container, Entity) permettant d'exposer les données. L'ancien a été abandonné au profit d'une exposition directe du SQL sur HTTP (TDS), devenant ainsi un véritable SQL Server online. La migration de l'ancien vers le nouveau est prévue pour cet été. Et Microsoft utilise sa plate-forme pour proposer ses SAAS (haut [Fig.1]). En résumé, Windows Azure est un environnement d'exécution élastique, à l'échelle mondiale, hébergé dans les Datacenter Microsoft et facturé à la consommation. Pendant la preview, tout est gratuit pour le moment).

CAS PRATIQUE : MIGRATION D'UNE APPLICATION ASP.NET VERS WINDOWS AZURE

Pour www.Zecloud.fr j'ai porté BlogEngine.Net 1.4.5 sur Windows Azure. En quelques étapes, je vais vous expliquer comment réaliser cette migration vers Windows Azure avec Visual Studio 2008 SP1.

Installation des outils indispensables

Commençons par installer les « Windows Azure Tools for Microsoft Visual Studio », depuis la version de Mars 2009 Le SDK Azure y est inclus. Cela installera les nouveaux modèles de projets dans Visual Studio 2008 et l'environnement qui permet de simuler Windows Azure sur son pc.

Fig.1

Architecture de BlogEngine.Net

C'est une solution vs 2005 composée d'un projet de site web (BlogEngine.Web) et d'une bibliothèque de classes (BlogEngine.Core). BlogEngine alias BE.net a une architecture [Fig.2] où les couches d'accès aux données sont bien séparées du reste du projet, grâce à l'implémentation d'un système de provider similaire à ceux livrés dans le framework ASP.Net comme le MembershipProvider, pour la gestion des utilisateurs.

BlogEngine.Core propose un XmlProvider (stockage par fichiers XML) et un DbBlogProvider (stockage dans une base de données SQL). Enfin, une classe abstraite BlogProvider dont on héritera pour implémenter son propre provider, ainsi qu'une collection qui contient tous les providers disponibles pour un site. Dans la mesure où BlogEngine.Net est un site Web, nous pourrons nous contenter dans un premier temps d'utiliser le template de Web Cloud Service.

Il est composé des éléments de la [Fig.3].

Le premier est un projet de déploiement et de configuration et sert à la simulation de l'environnement Azure en local. Il démarre la Development Fabric et le Development Storage et permet de simuler le Load Balancer et de lancer plusieurs instances de son Web Role en local. Il contient aussi le Service Model. Le second, le « Web Role » est un projet d'application web modifié pour Azure. Voici l'architecture de notre application Web hébergée sur Azure, lui permettant de s'adapter aux pics de charge [Fig.4].

Par conséquent, pour porter BlogEngine.net sur Windows Azure, 3 tâches s'imposent :

- Ecrire un nouveau provider pour stocker les données dans le « cloud »
- Réécrire les fonctions qui permettent d'uploader les images pour les stocker dans le « cloud » pour qu'elles soient accessibles à toutes les instances de mon « Web Role »
- Transformer le projet de site web du BE .net en « Web Role »

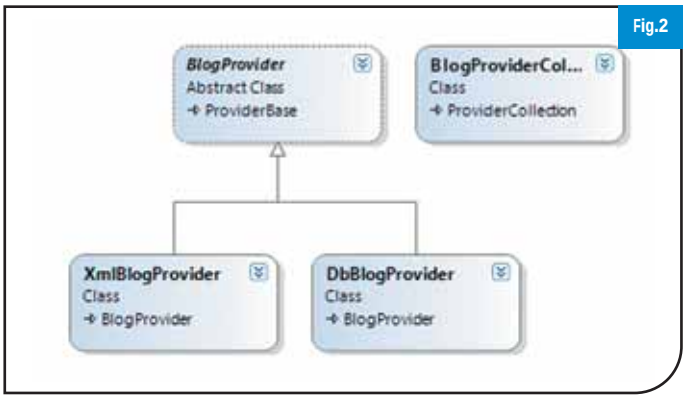


Fig.2

Il existe 3 solutions de développement pour un provider :

• Porter le XmlProvider pour stocker les fichiers Xml dans les blobs de l'Azure Storage.

Cette solution m'a paru la plus simple à développer au premier abord mais s'avérerait la moins efficace, la moins « scalable » et il faudrait gérer l'accès concurrentiel si 2 instances du site essayent d'écrire dans le XML au même moment.

• Faire un provider qui stockera les données dans l'Azure Table Storage.

C'est une solution satisfaisante, C'est « scalable » et un moyen simple existe pour gérer l'accès concurrentiel. Mais ce n'est pas relationnel, donc il n'y a pas d'intégrité structurelle des données avec des clés étrangères.

• Faire un provider qui stockera les données dans les SQL Data Services.

La meilleure solution à mon avis, mais aussi probablement la plus coûteuse quand les services deviendront payants, puisqu'il faudra payer pour Windows Azure et pour SQL Services. L'avantage notable est le stockage dans une base de données relationnelle qui apportera l'intégrité structurelle des données. Et avec la nouvelle version des SQL Data Services, cela devient une solution portable. On pourra rapatrier le blog sur un hébergement « maison » au besoin. Ces 3 solutions fonctionneront, à chacun de choisir celle qui lui convient le mieux. Pour ZeCloud, j'avais choisi de faire mon provider avec SDS (SQL Data Services), mais la version actuelle de SDS disparaîtra cet été. D'ici là, je referai un provider avec l'Azure Table Storage ou avec le nouveau SDS.

Le stockage des images

Comme mon application peut s'exécuter en plusieurs instances, il s'agit de rendre accessibles mes images à toutes les instances. Dans le dossier Samples du SDK, on peut trouver le projet StorageClient qui encapsule l'API REST de l'Azure Storage.

A l'aide de cette bibliothèque de classe, il suffit de réécrire les fonctions qui permettent d'uploader des images dans les pages d'administration de BlogEngine pour les stocker sous forme de Blobs dans l'Azure Storage. Dans BE.net, les images sont rendues à l'internaute via un Handler http.

Il suffit de le modifier un peu pour qu'il aille chercher les images dans les blobs et les images sont à présent accessibles à toutes les instances.

Migration du projet Web

Maintenant passons à l'étape suivante : transformer le projet de « site web » de BE.net en « Web Role ». Un Web Role est basiquement un projet d' « application web » modifié pour Azure. Il y a quelques différences notables entre un projet « site web » et un projet « application web » :

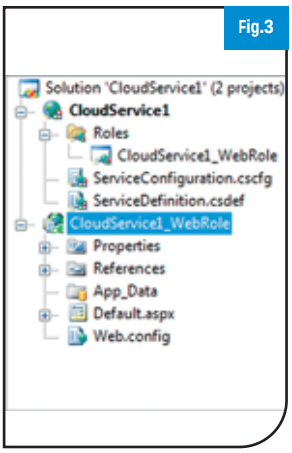


Fig.3

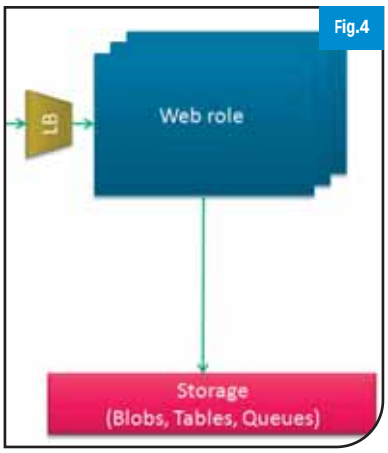


Fig.4

- Un site web pour chaque fichier ASPX contient 2 fichiers : 1 fichier aspx et 1 fichier .apsx.cs contenant le « code Behind » les 2 fichiers pouvant être réunis en 1 seul dans ce type de projet. Une application web contient un fichier de plus (.aspx.designer.cs) comportant la définition de chaque contrôle utilisé dans l'aspx.
- Dans un site web, on ne peut mettre des fichiers de classe que dans le dossier App_Code. Dans une application web, c'est comme dans n'importe quelle application .net : on peut mettre des classes dans tout dossier qui sera associé au namespace du même nom.
- Une application web est plus stricte qu'un site web. Dans un même namespace, on ne peut avoir une page du même nom qu'une seule fois. Ce qui posera problème, on le verra, dans BE .net.

Dans ma solution « webcloudservice » j'ai recopié le contenu de « BlogEngine.web » dans mon « webrole » et ajouté une référence

Infrastructure Windows Azure : lexique de base

Nœud : unité de calcul, cela peut être une machine physique ou une machine virtuelle

Rôle : application. Il peut être instancié en plusieurs exemplaires. 3 types de rôles sont disponibles dans les templates de Visual Studio actuellement :

- Le Web Role, c'est une application web Asp .Net
- Le Cgi Web Role, c'est une application web FastCgi (par exemple Php)
- Le Worker Role, c'est un traitement batch

Le rôle est exécuté sur les nœuds

Service Azure : ensemble de Rôles défini par le Service Model. 4 types de projets sont disponibles actuellement dans Visual Studio : Web Cloud Service, Worker Cloud Service, Web and Worker Cloud Service, Blank Cloud Service.

Service Model : Il est composé de 2 fichiers XML qui contiennent la définition du service, les différents rôles, les ressources nécessitées et la façon dont les rôles interagissent entre eux et avec l'extérieur

Fabrique : C'est l'ensemble logique des nœuds qui permet aux services Azure de fonctionner. En local La *Developer Fabric* simule la fabrique.

Fabric Controller : C'est le véritable cerveau d'Azure, Le *Fabric Controller* gère l'ensemble des ressources (Serveur, Load Balancer). Il alloue les ressources nécessaires lors d'un déploiement qu'il réalise. Il surveille les nœuds et les rôles, maintient un état pour chacun et si un nœud tombe, réalloue les ressources nécessaire à un rôle.

Azure Storage : c'est le système de persistance des données d'azure, 3 types de stockage :

Table storage : Des tables simples non relationnelles.

Blob storage : Permet de stocker des données binaires, pour y stocker des fichiers.

Queues de stockage : « premier entré premier sorti », permet d'échanger des informations entre différents rôles

vers la bibliothèque de classes « Blogengine.Core ». Dans le web.config on peut y trouver cette ligne :

```
<add namespace="Controls" tagPrefix="blog"/>
```

Cela ne marchera plus dans un Web Role, il faut y mettre le chemin complet du namespace. Pour plus de séparation du code. Je crée une nouvelle bibliothèque de classe que j'appelle AzureBlogUI, dans laquelle je vais recopier le dossier se trouvant dans App_Code. Puis je remets toutes les classes dans 1 nouveau namespace AzureBlogUI.UI.Controls pour les contrôles et pour le moment je décide de désactiver les extensions en excluant le code s'y rapportant. Je mets une référence vers cette dll dans mon webrole et je modifie le web.config ainsi.

```
<add namespace="AzureBlogUI.UI.Controls" tagPrefix=
"blog" assembly="AzureBlogUI"/>
```

Autre problème de collision de nom à l'intérieur du dossier thèmes : la « masterpage » de chaque thème à le même nom de classe, et toutes sont dans le même namespace, ce qui posera un problème de compilation. Deux solutions : soit je laisse un seul thème en excluant les autres, soit je renomme les noms des classes de chaque « masterpage » pour qu'il porte aussi le nom du thème. Le même problème se posera avec certains « user control » du dossier widgets, il suffira d'y appliquer la même solution.

A présent, en utilisant « convert to web application » (accessible via le deuxième bouton de la souris sur le projet web) cet assistant va créer les fichiers designer.cs pour tous les fichiers du site.

Enfin le serveur web de Windows Azure qui va héberger mon projet étant un IIS7, je dois faire quelques petites adaptations à mon web.config, les modules et les handlers nécessitant une nouvelle configuration spécifique. Il faut rajouter le tag <system.webServer> y recopier l'ancienne section <httpModules> et <httpHandlers> et adapter la configuration à la nouvelle section.

Après cela, je peux rajouter mon AzureBlogProvider dans le web .config et retirer les autres devenus inutiles.

BE.Net a besoin aussi d'un MembershipProvider et d'un RoleProvider pour la gestion des utilisateurs. Pour ZeCloud, j'ai utilisé un MembershipProvider et un RoleProvider dont les données sont aussi stockées dans SQL Data Services. Il a été fait avec le même code de base que mon AzureBlogProvider. Mais il est aussi possible d'utiliser très simplement l'Azure Table Storage. Dans le SDK, le dossier « Samples » contient un MembershipProvider et un RoleProvider qui stockent le contenu dans le Table Storage. Resteront quelques petits problèmes provoqués par l'hébergement Windows Azure : L'espace dans le "Src" d'un user control est mal interprété par Windows Azure. Si cela fonctionne en local, une fois déployé, une exception se produit. Il faut soit remplacer l'espace par un %20 ou renommer le dossier sans espace et mettre à jour le "Src" des usercontrols.

Attention, l'hébergement Windows Azure change certaines fonctions du framework :

```
context.Request.Url.GetLeftPart(UriPartial.Authority)
```

Normalement, il renvoie la partie gauche de l'url par exemple ici http://www.zecloud.fr

Mais sur Azure, sans doute à cause du load balancer, on récupère http://www.zecloud.fr:20000

C'est une adresse virtuelle utilisée par Azure, le même problème se produit si on utilise un Response.Redirect() avec une url relative en paramètre. Pour cela, il faudra adopter son code en conséquence : faire des redirect sur des adresses absolues et utiliser un code qui vous permettra de savoir si l'exécution se produit dans la Fabric. Une solution simple consiste à avoir une clé dans le web.config qui correspond à l'environnement d'exécution et à exécuter un code spécifique en fonction de la plate-forme (production dans la fabrique ou développement sur votre pc).

Prochaine étape ré-implémenter les extensions de BlogEngine comme les Pingbacks et les Trackbacks qui s'exécutent normalement dans des threads séparés, en créant un « WorkerRole » qui s'occupera de cette gestion.

Retrouvez le code source sur http://zecloud.codeplex.com

■ Aymeric **Weinbach**

MCTS .net 2.0 Web et MCTS SQL Server 2005

Architecte & Chef de projet

Fondateur de Zecloud communauté Open Source autour d'Azure et des Azure Camps, coding party Azure.

Le magazine du développement

PROgrammez !

www.programmez.com

Abonnez-vous : EN CADEAU

Accès illimité de 10 jours à la Bibliothèque Numérique

A choisir parmi les Bouquets de la Bibliothèque Numérique des Editions ENI :

Développement, Microsoft, Open Source

Pour tout abonnement au magazine Programmez un accès de 10 jours à la Bibliothèque Numérique de votre choix vous est offert. Offre Limitée aux 1000 premiers souscripteurs. (voir coupon d'abonnement, cahier Parallel Studio, page XV)

WEB

Ergonomie web 2e édition

*Difficulté : ***
Editeur : Eyrolles
Auteur : Amélie Boucher
Prix : 32 €

De plus en plus l'ergonomie du site web devient un critère sensible pour fidéliser les visiteurs et les transformer en futurs clients. Mais l'ergonomie web ne s'invente pas. Il faut en connaître les ressorts, les impératifs mais aussi les limites à ne pas dépasser. Premièrement : comprendre de quoi on traite exactement avant d'énumérer les règles d'or à respecter. Surtout, le designer, le développeur web doivent se mettre à la place de l'internaute et ne pas coder une interface pour leur seul plaisir. Cela n'a aucun sens. Une fois bien compris ces mécanismes, on passe aux choses sérieuses avec les règles d'ergonomie du web décrites, expliquées et démontrées une par une. Mais surtout, une des règles d'or est de penser ergonomie dès la conception de son site et pas seulement à la fin. Car c'est déjà trop tard ! Abondamment illustré, cet ouvrage s'avère vite indispensable grâce aux références et aux conseils que l'on retrouve un peu partout. A vous de jouer !

WEB

Google Apps

*Difficulté : ***
Editeur : Pearson
Auteur : Philip Lenssen
Prix : 59 €

Google n'est pas uniquement un moteur de recherche. Aujourd'hui, l'éditeur américain possède une grande variété de fonctions et d'applications en ligne pour tous les utilisateurs, les entreprises. Google Apps, en particulier, vous donne accès à des logiciels bureautiques collaboratifs (traitement de texte, tableur, agenda, bloc note), à une adresse électronique et à un espace de stockage de données sécurisé. Vous disposez également d'outils de modélisation, de retouche, de publication, d'analyse, à destination aussi bien des professionnels que des particuliers, de Sketch-Up à Google Analytics, de Picasa à YouTube. L'offre est riche et ces logiciels en ligne cachent souvent des fonctions surpuissantes qui n'attendent que vous. L'auteur propose de se plonger dans Gmail, Docs, Youtube, iGoogle. Très instructif pour celui qui veut aller plus loin.

ADMINISTRATION

Coffret Windows Server 2008 et Powershell

*Difficulté : ***
Editeur : Eni éditions
Auteur : collectif
Prix : 59 €

Voici un coffret indispensable pour tous ceux qui veulent administrer et mieux maîtriser Windows Server 2008 et PowerShell. Le livre sur Windows Server 2008 s´adresse à un public d´informaticiens débutants à intermédiaires, désireux d´acquérir des bases solides pour installer et administrer ce système d´exploitation. Chaque chapitre débute par une introduction sur le sujet puis propose une approche pas à pas pour sa maîtrise. Sur Powershell, tout au long des chapitres, le lecteur découvrira PowerShell sous toutes ses facettes : de la simple utilisation de l'interpréteur de commandes, jusqu'aux techniques de scripting les plus avancées. Les chapitres suivants lui apporteront une ouverture sur le monde de l'administration système ; ainsi que de nombreux cas concrets d'utilisation en situation réelle. Enfin le dernier chapitre donnera au lecteur un avant-goût des fonctionnalités à venir de la version 2.0 de PowerShell.

SCIENCE

L'intelligence artificielle par la pratique

*Difficulté : *****

Editeur : Presses polytechniques et universitaires romandes
Auteur : collectif
Prix : N.C.

L'informatique actuelle est synonyme d'intelligence artificielle. Elle constitue la base de nombreuses applications et technologies. Mais qu'est-ce que l'on entend réellement par intelligence artificielle ? Les auteurs se proposent de vous ouvrir les portes de cette science un peu particulière pour comprendre son fonctionnement comme les représentations de connaissances, l'inférence logique ou encore le traitement d'informations incertaines. L'austérité du propos est à la hauteur de l'ambition de l'ouvrage. C'est précis, avec de nombreux exercices pour tenter de mieux approcher le fonctionnement de cette IA.

LIVRE DU MOIS

Conception objet en Java avec BlueJ

4e édition

*Difficulté : ****
Editeur : Pearson
Auteur : collectif
Prix : 39 €

Comment apprendre le plus simplement possible, la programmation objet ? La réponse, dans le monde Java du moins, est BlueJ. Cet outil possède une approche ergonomique et simplifiée de la programmation objet en Java. Il reprend notamment une approche UML (sans que cela en soit). Pour bien appréhender chaque étape, le lecteur travaille à partir de cas concrets, d'applications à développer (plus de 23 projets expliqués et 400 questions et exercices). Cette édition prend en compte les dernières évolutions de Java 6. Accompagné d'un CD-Rom. Une bonne lecture sur la plage !

LANGAGE

Tout sur le C++

*Difficulté : ***
Editeur : Dunod
Auteur : Christine Eberhardt
Prix : N.C.

Voici un mini guide fort sympathique sur le C++. L'auteur en quelques dizaines de pages explique et présente les conceptions objet, les notions avancées, les opérateurs, les mécanismes du langage. Rapide à lire, ce « Tout sur le » est avant tout utile en aide-mémoire ou pour découvrir rapidement ce qu'est l'objet et le C++. Une approche originale.

Dans la même collection : **Tout sur les réseaux sans fil,** Tout sur le hardware PC.

Vitesse, mémoire, sécurité.

Les serveurs dédiés PRO offrent d'énormes performances.

À partir de

99 € HT/mois*

Satisfait ou remboursé**

ASSISTANCE TECHNIQUE PAR TELEPHONE 7J/7 (1)

AVEC LES PROCESSEURS AMD OPTERON™ QUADRIPROCESSEURS, VOUS DISPOSEZ D'UNE INCROYABLE PUISSANCE DE TRAITEMENT ET D'UNE ÉNORME CAPACITÉ DE MÉMOIRE. VOUS POUVEZ AINSI HÉBERGER VOS APPLICATIONS MÉTIERS LES PLUS EXIGEANTES OU DES SITES WEB À FORT TRAFIC EN TOUTE SÉCURITÉ.

- AMD Opteron™ 1352, 1354 ou 1356
- De 4x2,1 GHz à 4x2,3 GHz
- De 4 à 8 Go de RAM
- Disques durs de 2x500 Go à 2x1 To
- Raid 1 Hardware
- 2 à 6 adresses IP
- Trafic mensuel illimité
- OS 64 bits : Debian 4, Ubuntu 8.04 LTS, CentOS 5 ou Windows Server 2008
- Interface Plesk de 30 à 300 domaines
- Sauvegarde FTP : de 40 Go à 80 Go
- Services Inclus : Reboot, Restore et Recovery
- Assistance technique par téléphone 7j/7(1)

Amen et Dada : 1,4 million de domaines gérés et plus de 500 000 sites hébergés.

Amen
A DADA COMPANY

0 800 74 09 35 (Numéro gratuit) - www.amen.fr

NOMS DE DOMAINE - EMAIL - HÉBERGEMENT - CRÉATION DE SITE - E-COMMERCE - RÉFÉRENCEMENT

*Pour un engagement annuel. ** Sous 10 jours.
(1) 08 99 70 90 01, coût de la communication pour l'assistance technique : 1,35 € l'appel + 0,34 € la minute.

REFLEXION**FAITE** - Conditions générales de vente sur le site www.amen.fr. AMEN RCS PARIS : B 421 527 797.

Infragistics®

Introduisant

NetAdvantage pour .NET 2009 Volume 1

Des composants interface utilisateurs supérieurs pour Windows Forms, WPF, ASP.NET et Silverlight

Pour de plus amples infos : infragistics.com

NetAdvantage®
.NET ASP.NET, WinForms, WPF, Silverlight

Four Platforms. One Package.

N° Vert 0800 667 307

grids tree menus navigation charts & more!

Copyright 1996-2009 Infragistics, Inc. All rights reserved. Infragistics, the Infragistics logo and NetAdvantage are registered trademarks of Infragistics, Inc. All other trademarks or registered trademarks are the respective property of their owners.