

PROgrammez !

www.programmez.com

Mensuel n°154 - juillet-août 2012

PROGRAMMEZ cet ÉTÉ !

Pilotez votre maison

La domotique facile : la maison connectée et programmable

istockphoto© Henrik Jonsson



Webmaster

Créer son navigateur avec Webkit
JQueryMobile : des interfaces mobiles en quelques minutes

Android

IntelliJ : un IDE Java surpuissant,
Développer des widgets

iPhone / iPad

Le moteur graphique
Cocos2D

BUZZWORD

Nouveau Windows Azure

- Monter en 2 minutes un site web Drupal, WordPress, PHP, ASP !
- La fin des Hébergeurs ?

PHP en 2012

Rasmus nous dit tout !



Hardware

Pourquoi utiliser un disque SSD

Printed in EU - Imprimé en UE - BELGIQUE 6,45 €
SUISSE 12 FS - LUXEMBOURG 6,45 € - DOM. SUR 6,90 €
Canada 8,95 \$ - CAN - TOME 9,40 \$ - MAROC 50 DH



DÉVELOPPEZ 10 FOIS PLUS VITE

Nouveau:
iPhone
iPad

WINDEV®

- Windows 32 & 64 bits
- Linux
- Mac
- Internet
- Intranet
- Windows Mobile & CE
- Windows Phone
- Android 
- et maintenant*
- iPhone et iPad.

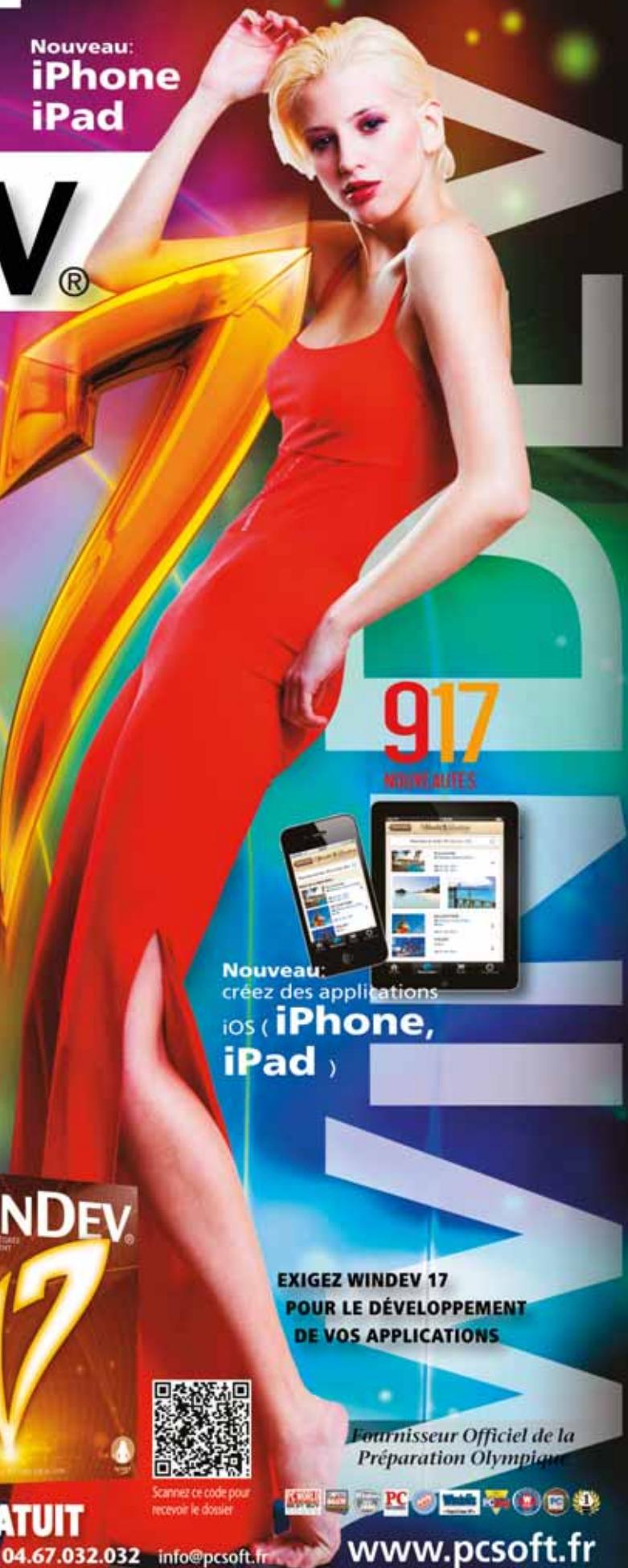
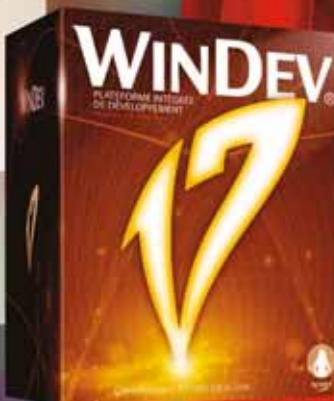
• Développez vos applications une fois pour toutes (les plateformes). Votre code, vos fenêtres, vos données, vos rapports,... sont **compatibles**. Déployez vos applications sur **tous les systèmes** et tous les matériels, dans tous les domaines, pour toutes les volumétries. Vous aussi, développez 10 fois plus vite, pour **toutes les plateformes**.

VERSION
EXPRESS
GRATUITE
Téléchargez-la !

Intégralement en français.
Support Technique inclus.
Ouvert à tous les standards.
à toutes les bases de données

Élu « Langage le plus productif du marché »

WINDEV, WEBDEV et WINDEV Mobile sont compatibles



Nouveau:
créez des applications
iOS (iPhone,
iPad)



EXIGEZ WINDEV 17
POUR LE DÉVELOPPEMENT
DE VOS APPLICATIONS



Scannez ce code pour
recevoir le dossier



Fournisseur Officiel de la
Préparation Olympique

► DEMANDEZ VOTRE DOSSIER GRATUIT

Dossier gratuit 260 pages sur simple demande. Tél: 04.67.032.032 info@pcsoft.fr

www.pcsoft.fr

«Monter en 2 minutes un site web Drupal, WordPress, PHP, ASP»

Hébergeurs : Windows Azure “m'a tuer” !

Avec les nouvelles fonctions dévoilées le 7 juin dernier, Windows Azure s'ouvre à Linux, héberge des sites web et déploie des machines virtuelles.

La force de cette nouvelle version est d'avoir su combiner des fonctionnalités purement plateforme (PaaS) et infrastructure (IaaS).

Windows Azure version dopée juin 2012, c'est du lourd : portail HTML5 accessible partout, outils en ligne de commande pour Linux et OS X (yep, on peut enfin se marquer un peu), machines virtuelles Windows, Linux à gogo, nouveau VPN, nouveau système de load balancing, MySQL, kits Java et PHP revus. Bref, maintenant on peut vraiment s'amuser avec Azure !

> Web site ou comment tuer le business des hébergeurs

Vous voulez monter en 2 minutes un site web Drupal, WordPress, PHP, ASP ? Le service site web de Azure permet en quelques clics de monter son WordPress, parfaitement fonctionnel, avec sa base de données, son interface, sa console d'administration. Sur tout, vous pouvez ajuster les ressources (type d'instances), dire s'il s'agit d'un serveur réservé (donc dédié) ou mutualisé.

Le tout avec le SLA Azure, les réplications automatiques, la migration à chaud sans interruption quand on passe du mutualisé au réservé. Alors pourquoi passer par un hébergeur pour son petit site, son blog ou même pour des sites de e-commerce alors que l'on peut déployer, développer rapidement, et être compatible avec une multitude de technologies ?

Azure vous donne droit jusqu'à 10 sites gratuits (attention : quota restreint du nombre de base de données) sur un même compte Azure. Une galerie de sites pré-configurés est disponible : Drupal, WordPress, Joomla, Asp, etc. Microsoft a beaucoup travaillé avec la communauté pour simplifier l'installation. Vous pouvez même spécifier le niveau de langage : PHP 5.3, .Net 2 ou 4. Franchement, si vous trouvez un hébergeur qui propose une interface de contrôle et d'administration comme celle-ci, je vous paie une quiche (ben oui, faut pas abuser tout de même).

Azure Web Site propose souvent par défaut MySQL (oui vous avez bien lu) mais vous pouvez aussi utiliser SQL Server si vous le souhaitez.

tez. Vous pouvez aussi partager entre plusieurs sites une même base MySQL si besoin.

> Un peu de WebMatrix dans ce monde de brut html

Windows Azure fait le lien entre la partie web site et les outils. Ainsi, en quelques clics vous avez accès à la plateforme WebMatrix. On peut facilement installer l'outil et importer localement le site créé qu'il soit en technologie .Net ou non. Pratique pour un site WordPress, Drupal, Php. Azure s'occupe de tout ou presque. Faites la même chose avec un hébergeur classique ! On peut toujours faire des bidouilles et faire des liens...

Des hébergeurs proposent des packages qu'il faut installer et les outils de sites rapides à faire ne sont pas aussi puissants que là. Sur tout, une grosse partie de l'administration, vous n'y pensez même plus !

Le tableau de bord par défaut fournit les informations de base : occupation mémoire, le stockage libre / utilisé, la CPU. En quelques secondes, vous avez toutes les informations nécessaires pour voir l'activité et ce, même sur un iPad, Android, etc.

Vous pouvez très rapidement monter en charge si votre site explose : mettre 10 instances, faire du réservé et non du mutualisé (là, vous sortez la carte bleue). Et vous pouvez aussi ajuster la taille des instances de la très petite à l'extra-grande. Vous avez même accès à de puissants gestionnaires de code et de projets : TFS, Git ! L'installation WebMatrix prend parfois un peu de temps et surtout, votre machine n'a pas les pré-requis. Avantage : machine à café. Inconvénient : retour à la machine à café (note pour plus tard : penser à remplir une thermos). On peut travailler directement en mode remote sur le répertoire en ligne (Azure donc) ou importer le site entier et travailler en local pour ensuite le redéployer. WebMatrix vérifie les dépendances, la base MySQL. C'est pas top ça ? Et si vous avez Visual Studio, le lien est automatique... Nous reviendrons en profondeur sur Windows Azure June 2012 et le nouveau SDK dans le prochain numéro...

Sur la partie machine virtuelle, j'ai mis en ligne une fiche « premiers pas »

François Tonic



**BUZZWORD**

Hébergeurs : Windows Azure « m'a tuer » ! 3

ACTUS

En bref 6

SudWeb 2012 : la conférence web de l'année 8

PHP en 2012 : la mise au point de Rasmus 12

Forum PHP 2012 :

Zend Framework 2 et 400 personnes ! 14

MATERIEL

SSD, l'arme ultime des performances ? 15

Interview Laurent Sirgy (Kingston Technology) 16

Interview Nicolas Maigne (Crucial) 18

RESSOURCES

Livres et événements du mois 19

Agenda 19

DEVELOPPEUR DU MOIS

Frédéric Pedro, développeur, prof et chef de projet ! 20

**OUTILS**

Bases NoSQL et MongoDB 22

Wakanda, a stack is born ! 24

Concevoir des applications Web

Model-Driven

avec Wakanda 26

Premiers pas avec Wakanda 28

**GAMING**

Cocos2D un moteur de jeu pour les 44

CARRIERE

Epitech : formation, open source, innovation et création d'entreprise 49

JE DEBUTE avec...

Windows 8, partie 4 : la revue de code 53

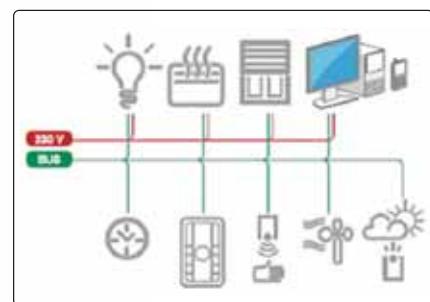
IntelliJ IDEA 11 CE : développer pour Android 57

DOSSIER DOMOTIQUE**La maison connectée : la geekattitude !**

• Piloter sa maison ? C'est facile !

• Les technologies, réseau domestique (1re partie) 31

• La maison connectée 36



Septembre 2012 - **n°155**, parution le 3 septembre 2012
Un numéro à ne pas manquer !



14

CAS D'USAGE

Base de données locale WP7 60

PRATIQUE

Donnez une nouvelle dimension à vos applications

Android grâce aux Widgets 64

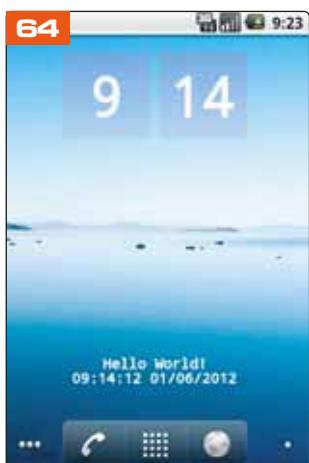
Créez votre propre navigateur avec Webkit et Qt 68

WEBMASTER

jQueryMobile, quelques lignes pour une interface mobile 76

BUGTRACK

MySQL : identifier les tables saturées 82

**Les ennemis de mes ennemis sont mes amis mais aussi mes ennemis**

« Est-ce qu'il y a du sens à participer à un salon dédié aux solutions LINUX alors que son sponsor principal est... Microsoft ? ». Voilà une des justifications de Linagora pour ne pas être présent au salon Solutions Linux de juin. Pourtant, Microsoft était aussi présent sur les gros événements PHP (Forum PHP et Symfony Live) par le truchement de Windows Azure. Et cela n'a choqué personne : pas d'évanouissement (hormis devant Rasmus et votre serviteur ☺), ni de hordes de geeks enragés contre les images impures des logos Microsoft.

Plus que jamais, il n'existe plus de tout blanc ou de tout noir. Tout est nuance. Si vous prenez Windows Azure, Microsoft a travaillé avec plusieurs communautés (Java, PHP, Drupal, Wordpress, des distributions Linux) pour offrir de nouveaux services. Proposer des machines virtuelles Linux directement sur Windows Azure est une évolution significative, qui relève du pragmatisme : aller là où les utilisateurs sont. « *Bon ok, les utilisateurs veulent du Linux pour monter des serveurs Linux en mode IaaS. Nous allons le faire.* »

Non content de batailler frontalement avec Amazon EC2, Azure s'ouvre à des scénarios impossibles il y a encore un mois. Amazon a aussi fait un pas vers .net en supportant officiellement ASP.net sur son Beantalk. Zend, SensioLabs, travaillent aussi activement avec l'éditeur pour être optimisés pour Windows Azure. C'est pour eux un marché important car beaucoup de développeurs travaillent sur Windows.

Enfin, faut-il boycotter Linux ? Car Microsoft contribue (modestement convenons-en) au noyau. Il ne manquerait plus que le Grand Satan travaille avec LibreOffice pour implémenter les formats OpenXML (ce qui ne serait pas forcément une mauvaise idée au passage)...

L'autre annonce que vous n'aurez sans doute pas manquée : Apple veut éradiquer Google Maps de iOS avec la nouvelle version de Plans (Maps serait-il devenu subitement un malware ?). Il faut dire que la Pomme en avait assez de dépendre de Google pour cette fonction et il fallait rentabiliser les différents rachats dans la cartographie 3D. Apple ne veut pas décevoir sur la qualité des cartes et les fonctionnalités, pour Google. iOS demeure un marché important pour le trafic et les revenus, d'où une présentation précipitée de la « nouvelle dimension » de Google Maps quelques jours avant la pomme croquée.

Si vous avez raté des épisodes, ce n'est pas bien grave. L'informatique c'est comme « Plus belle la vie », c'est sans fin.

Rendez-vous le 3 septembre pour le n°155. Ah ! si, avant de vous lâcher pour l'été, mention spéciale à notre Linus et à son célèbre « doigt » envers Nvidia. J'veus l'avais bien dit, on ne s'ennuie jamais dans l'informatique.

François Tonic - Rédacteur en chef
tonic@programmez.com

Editeur : Go-Q2 sarl, 21 rue de Fécamp 75012 Paris - diff@programmez.com

Rédaction : redaction@programmez.com

Directeur de la Rédaction : Jean Kaminsky. **Rédacteur en Chef :** François Tonic - ftonic@programmez.com.

Ont collaboré à ce numéro : F. Mazué, Y. Grandmontagne, S. Sauré. **Experts :** R. Duhem-Verdière, C. Villeneuve, J. Wortel, J.-M. Biraghi, A. Morgaut, S. Warin, S. Amer, S. Cordonnier, P. Didiergeorges, V. Fabing, A. Zanchetta, K. Alnijres.

Photos couverture : © istockphoto Henrik Jonsson
Publicité : Régie publicitaire, KNow sarl.

Pour la publicité uniquement : Tél : 01 41 77 16 03 - diff@programmez.com. Dépôt légal : à parution - Commission paritaire : 0712K78366 ISSN : 1627-0908. Imprimeur : S.A. Corelio Nevada Printing, 30 allée de la recherche, 1070 Bruxelles Belgique.

Directeur de la publication : J-C Vaudecrane

Abonnement : Programmez, 17, Chemin des Boulangers, 78926 Yvelines Cedex 9 - Tél : 01 55 56 70 55 - abonnements.programmez@groupe-gli.com - Fax : 01 55 56 70 91 - 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

» Où en sont les ex-IDE de Borland rachetés par Embarcadero ?

Delphi et C++ Builder confirment, selon l'éditeur, leurs marchés. Delphi possède toujours un fidèle socle de développeurs alors que C++ Builder profite du renouveau de C++ et des applications natives sur mobile. L'approche intégrée de RAD Studio XE2 et le multiplateforme (Windows, OS X, iOS) sont deux points forts évoqués. Embarcadero annonce même +54 % en 2011 sur le nombre de licences vendues. La partie iOS est un axe stratégique fort avec l'explosion de la

plateforme d'Apple. Cependant, Linux n'est toujours pas un marché prioritaire pour l'éditeur, même si une déclinaison Linux arrivera tôt ou tard. Sur Windows 8, il n'y aura pas de nouveaux outils, mais l'intégration avec le nouveau système de Microsoft ne sera pas simple, notamment à cause de l'interface et des modèles de développement. Par contre Prism, outil .Net, est toujours présent dans Visual Studio et permet déjà de créer des applications Windows 8. RadPHP reste lui aussi au catalogue et se renforce sur la partie jQuery Mobile et Phonegap.



C++Builder XE2

Rapid visual C++ development environment



» Une grave faille de sécurité a frappé MySQL et les différents forks.

Cette faille critique se situe dans le module de vérification du mot de passe. Elle permet de se connecter au service par une attaque en force brute, pour peu qu'un nom d'utilisateur soit connu. Et comme la plupart des MySQL ont comme nom d'utilisateur, avec tous les droits, root ou admin, cela devient un jeu d'enfant. Le défaut a son origine dans le fait que le code part du principe que la fonction memcmp() qui est utilisée pour tester la validité du mot de passe retourne toujours une valeur comprise en -127 et 127. Cependant, selon les optimisations du compilateur utilisées pour compiler MySQL, une valeur en dehors de ces bornes peut être retournée, qui fera déduire au code qu'un mot de passe complètement faux est valide. En fait, la faille est telle qu'il y a une chance sur 256 qu'un mot de passe erroné soit accepté. Bref, un utilisateur de votre système cassera MySQL avec un script shell rudimentaire :

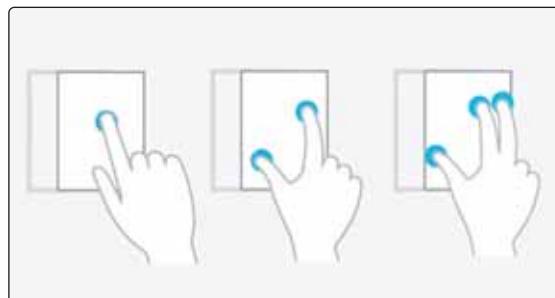
```
for i in `seq 1 1000`; do mysql -u root --password=bad -h 127.0.0.1 2>/dev/null; done
```

Toutes les versions de MariaDB et MySQL jusqu'à 5.1.61, 5.2.11, 5.3.5, 5.5.22 incluses sont vulnérables. Patchez vite votre système !

La faille expliquée : <http://seclists.org/oss-sec/2012/q2/493>

» Visual Studio Express 2012

Microsoft a été vertement critiqué suite à la décision de Microsoft de limiter le modèle de développement aux seules applications Metro. Les développeurs ont crié au scandale, avec raison ! Pour développer des applications de bureau, il fallait soit acquérir un Visual Studio 2012 d'une édition supérieure, ou bien télécharger un Visual Studio Express 2010. Quelques jours plus tard, Microsoft a décidé de proposer une édition Express pour le développement desktop (C++, C#, VB). Ouf ! Au passage, Visual Studio 2012 passe en release candidate.



» Coverity propose une nouvelle méthode pour l'analyse statique pour mieux détecter les défauts de sécurité dans les applications web écrites en Java. Les nouveautés permettent de :

- Renforcer l'analyse statique du code source avec un analyseur de frameworks qui réduit les imprécisions lorsque les données transitent par les frameworks applicatifs, réduisant ainsi les faux positifs.
- Valider automatiquement que les routines de nettoyage des données effectuent une désinfection suffisante des données suspectes et sont utilisées dans le contexte correct, grâce à un « White Box Fuzzer » incorporé à l'analyse statique.
- S'appuyer sur des suggestions de traitement détaillées pour chaque défaut détecté, afin de garantir que les développeurs comprennent comment corriger les défauts de sécurité correctement et efficacement.

Disponibilité de toutes ces nouveautés en septembre !

Et aussi :

- Facebook prépare un SDK spécifique pour les applications iOS
- Mozilla a dévoilé un éditeur de codes HTML 5 en ligne, Thimble. Site : <https://thimble.webmaker.org/en-US/editor>
- Micro Focus travaille activement à Visual Cobol 2 qui est lancé conjointement, en collaboration avec Microsoft, avec Visual Studio 2012 RC. Visual COBOL 2.0 s'intègre à Visual Studio 2012. Il prend également en charge la version la plus récente de .NET Framework (4.5) afin de tirer parti des nouvelles possibilités de développement offertes par la plateforme .NET.
- PostgreSQL met à jour plusieurs versions pour corriger 42 bugs dont deux sérieux. Le premier de ces problèmes réside dans une instabilité pouvant conduire à un crash du serveur lors de l'utilisation de fonctions de type SECURITY DEFINER. Le second concerne une faille de sécurité découverte dans la fonction crypt(). Il est à noter que les projets PHP, OpenBSD et FreeBSD sont aussi affectés par ce problème et ont diffusé leur propres correctifs.

MetroTactual

[me-troh tak-choo-uhl]

SKINS

Compatible avec
Microsoft® Visual Studio® 11 Beta

nom, adjetif

1. Moderne, net, élégant avec design et UX optimisés tactile
2. Feng Shui pour vos applications
3. Inclus dans le NetAdvantage toolsets 12.1

Voir aussi: NetAdvantage for .NET

Obtenez gratuitement votre version
d'essai dès aujourd'hui!
www.infragistics.com/.NET



 **INFRASTICS**
DESIGN / DEVELOP / EXPERIENCE



SudWeb 2012 : la conférence web de l'année



Rassembler des geeks du Web sans connexion wifi ? SudWeb (25-26 mai dernier à Toulouse) relève le défi sans aucune discordance ! Diversité des sujets abordés, qualité d'écoute et échanges constructifs sont au rendez-vous de cette conférence où s'enrichir de l'expérience des autres, permet de construire un Web de qualité. Deux jours pour « faire savoir et savoir faire », comme l'annonce le slogan de l'événement. Le Web ouvert, libre et partagé, était au cœur de cette édition.



Organisée par une poignée de bénévoles passionnés, cette conférence réunissait 150 professionnels du Web français (et quelques invités étrangers), des étudiants, des concepteurs de sites web et des développeurs qualifiés. Au programme : mobilité, responsive webdesign, bonnes pratiques, méthodologie et expérience utilisateur. Le tout dans une alternance bien rythmée de conférences longues et courtes, jusqu'au « lightning talk » de quelques minutes. Comment vendre des prestations agiles ? Qu'est-ce qu'un lead dev ? Comment tester ses idées rapidement ? Quel est l'avenir de CSS ? Comment intégrer les micro-données ? Quels sont les meilleurs outils pour débugger sur mobile ?

La seconde journée permettait d'approfondir et de mettre en pratique les sujets abordés la veille et d'en explorer d'autres, sous la forme d'« élaboratoires », néologisme désignant des ateliers à mi-chemin entre le BarCamp et le Coding Dojo : dans les salles de travail de l'Epitech, école informatique toulousaine reconnue, une grille de programmation éditable par toutes les personnes présentes et une folle envie de partager. Code, rédaction, méthodo, brainstorming,

webdesign ou même business : l'essentiel est de tester, créer et découvrir de nouvelles manières de travailler ! L'agiliste Pablo Pernot en a profité pour hacker l'événement en lançant un « Marshmallow Challenge » et le collectif OpenWeb en a profité pour renaître, dans « un amour partagé pour bien faire le Web et le propulser à d'autres » témoigne Karl Dubost.

Côté organisation, le staff, remarquable d'efficacité et de discrétion, a tout prévu pour favoriser les rencontres et les échanges entre auditeurs et orateurs. Repas en commun, questions à l'ombre du figuier, bienveillance, traduction simultanée... et pas de wifi dans les salles de conférences : tout le monde écoute, oubliant même de live-tweeter et pas un geek ne râle, bravo ! C'est une volonté du staff, qui témoigne : « *Le vrai luxe, maintenant, c'est de savoir déconnecter. Ce n'est pas la machine qui nous domine, mais nous qui la maîtrisons !* ».

Cette conférence se distingue aussi par le soleil, la détente et la bonne humeur : après Nîmes, cette seconde édition avait lieu fin mai à Toulouse, en tongs et pantacourt, par un temps magnifique. En plus des goodies distribués à chacun et chacune, il y avait des

bouquins EYrolles et même des licences Balsamiq Mockups à gagner !

> Les conférences les plus importantes

Mais un lead dev, c'est quoi ? En définissant le « lead dev », la première conférence de la journée pointe le problème de reconnaissance du métier de développeur en France, où la seule perspective est de devenir chef de projet, ce qui n'est pas le même métier. C'est l'occasion du premier troll : « *le chef de projet, c'est celui qui pense qu'avec neuf femmes on peut réussir à faire un bébé en un mois.* » Le lead dev n'est pas un chef, mais un guide. Le chef de projet est confronté au client, qu'il représente au sein de l'équipe, alors que le lead dev met ses compétences techniques au profit du produit. À la fois pompier, archéologue et chef d'orchestre, il guide les développements.

Plus fort que le responsive webdesign, voici l'**open web en tant que pierre angulaire du développement multi objets** ! Ça n'est pas un site web... mais ça utilise beaucoup le Web, d'où l'appellation « openweb »... ce peut être un grille pain, un lapin Nabaztag,



ActiveReports 6

à partir de € 561



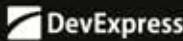
Dernière version du générateur de rapports .NET sans droits le plus vendu.

- Moteur de génération de rapports rapide, souple avec licence sans droits pour le Web/Windows
- Visualisation des données et contrôles de mise en forme: Chart, Barcode et Table Cross Section
- Prise en charge de Medium Trust dans l'environnement ASP.NET
- Vaste gamme de formats d'exportation/aperçus: Windows Forms Viewer, Web Viewer, Flash & PDF
- Contrôle utilisateur de conception avec API souple et personnalisation



DXperience DXv2 12.1

à partir de € 1 203



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

- Abonnement de 12 mois pour tous les produits et mises à jour DevExpress et accès aux versions bêta en développement actif
- Modèles et thèmes d'application intégrés exceptionnels
- Support de nouvelle vue Metro UI et panneaux ancrables tactiles
- Support interface codée pour test environnement utilisateur



TX Text Control .NET for Windows Forms/WPF

à partir de € 911



Composants complets de traitement de texte pour Windows Forms et WPF.

- Le traitement de texte professionnel pour vos applications
- Chargez, modifiez et enregistrez au format MS Word (DOCX, DOC, RTF)
- WYSIWYG, tableaux imbriqués, cadres, en-têtes, pieds de pages, images, puces, listes numérotées, zoom, sauts de section, etc.
- Exploitez la liaison de données XAML pour lier les barres d'outils et de ruban



Janus WinForms Controls Suite V4.0

à partir de € 743



Ajoutez des interfaces de style Outlook à vos applications .NET.

- Vues ruban, grille, calendrier, et barres chronologique/raccourcis
- Nouveau – Style visuel Office 2010 pour tous les contrôles
- Nouveau – Support des profils client Visual Studio 2010 et .NET Framework
- Janus Ribbon ajoute Backstage Menus et la fonctionnalité onglet comme dans Office 2010
- Prend désormais en charge la sélection de cellules multiples

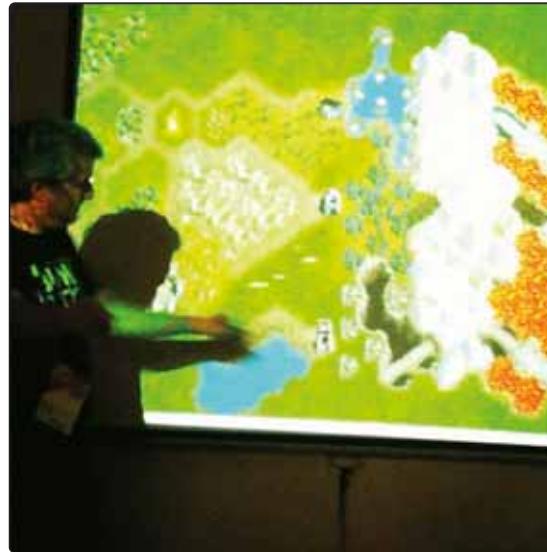
un réfrigérateur intelligent, un miroir tactile... Cette conférence très concrète, bien technique, repose sur le même principe « write once, run everywhere » que le responsive webdesign, car pour les objets connectés, pas besoin de faire du natif : du web encapsulé dans un conteneur natif suffit. Les outils ? media queries, modernizr et d'autres scripts comme formfactor.js, device.js et backbone.js, car les media queries ne suffisent pas à cibler les objets connectés. Finalement, Joshfire a développé son propre framework, open source : <http://framework.joshfire.com>

Comment vendre des prestas agiles ?

En tant que développeur indépendant, Thibault Jouannic a besoin de faire bouillir la marmite. Il nous explique comment vendre des prestations agiles pour répondre à tous ces prospects qui le sollicitent en voulant déjà savoir combien ça va durer et coûter, alors qu'ils ne savent pas encore ce qu'ils veulent... Deux arguments pour séduire le client : pas besoins de specs, on vous en fait cadeau ! Et la première livraison n'est pas dans 6 mois avec 3 mois de retard, mais dans 2 semaines. Ce premier entretien est décisif. Il pose les bases de la gestion de projet réussie.

« Startup weekend addict », Camille Roux nous donne la recette magique pour réussir son projet avec le « lean canvas », un tableau qui permet de mûrir son idée. Ensuite, il faut tester le concept : en consultant des experts, interviewant des clients et en sortant visiter le vrai monde, avant même de prototyper. **Ne perdez plus votre temps, testez vos idées en quelques heures !**

Intéressant retour d'expérience que celui de Rachel Andrew qui est passée de boîte de service à éditeur (**From service provider to software vendor**) : l'agence de développement web edgeofmyseat.com est devenue éditrice d'un petit CMS appelé Perch. Première leçon : on ne doute pas que les développeurs aient du goût, mais ils ne sont pas designers. Or un produit qui marche est avant tout un outil qui répond aux besoins de ses utilisateurs. Dès lors, il faut se demander : quelle est la fonctionnalité qui va faire la différence et qui va apporter le plus, au plus de gens possibles ? S'il faut mettre les



clients au premier plan, il ne faut pas céder à tout et notamment veiller à préserver l'identité du produit. Perch est un petit CMS, basique. Les demandes d'ajout de briques fonctionnelles de toutes sortes sont nombreuses, mais il faut savoir refuser... Mieux vaut proposer une API qui permet l'ajout de telles briques, et préserver la simplicité du scénario initial qui est : éditer du texte sur une page web.

« Oh, IE6, how we loved you » rappelle Bruce Lawson, évangéliste chez Opera Software, nous contant la romance IE6, navigateur que nous avions tant aimé, même s'il semble aujourd'hui difficile d'expliquer pourquoi. Pourtant, le scénario se répète avec webkit, le moteur de rendu de Safari et Chrome. La faute aux concepteurs de sites web ? Il y a ceux qui respectent les standards et les autres, qui contribuent au monopole de webkit, ressuscitant le syndro-

me IE6. Nouvelle guerre des navigateurs à l'horizon ? Il est important de privilégier un web ouvert et d'entretenir une concurrence saine entre navigateurs, dont tout le monde bénéficie. Et c'est à nous, développeurs et intégrateurs, d'en être les garants.

Jérémie Patonnier rappelle l'importance de la documentation, sans laquelle un projet est difficilement utilisable, pas maintenable. Il apporte son témoignage de contributeur **MDN (Mozilla Développer Network)** qui centralise des « ressources pour un meilleur web, quels que soient la marque, le navigateur ou la plateforme. Tout le monde peut contribuer et chaque personne supplémentaire nous renforce. » Son conseil : « *documentez ce que vous savez faire, et ce que vous aimez le plus* ».

La plupart des entreprises produisent du web,

mais pas optimisé pour l'utilisation par un client. À quelle vitesse ma page s'affiche-t-elle ? Il n'y a pas de réponse simple à la question de la performance selon Boris Shapira qui préconise des mesures d'UX plutôt que techniques et statistiques, dans la **mise en place d'une stratégie de performance en entreprise**.

Développeur front-end, Bertrand Keller témoigne de la frustration du métier, mettant en cause l'organisation du travail, notamment en agence où « *les concepteurs conçoivent et les développeurs développent* » sans coopérer, invitant l'intégrateur à sortir de ses gonds pour intervenir en amont : **Intégrateur, lève-toi et conçois !**

La journée s'achève sur l'intervention d'Anthony Ricaud qui nous invite à **l'altruisme, pour notre bénéfice** ou comment participer à des projets Open Source : corriger des bugs, partager ses découvertes, rejoindre un projet, pour progresser ensemble.

Le programme est consultable sur le site officiel de l'événement, Sudweb.fr, et les présentations sont centralisées petit à petit sur Lanyrd : <http://lanyrd.com/2012/sudweb/>

Romy Duhem-Verdière
Consultante web UX, Alter Way

Windows Phone 8 va succéder à Windows Phone 7

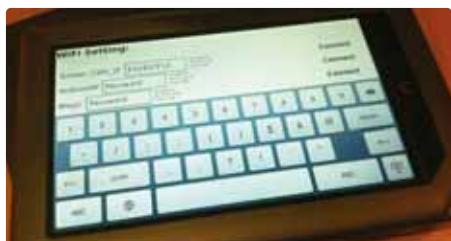
Microsoft a présenté le 20 juin dernier le prochain système mobile pour les téléphones. L'élément majeur à retenir est la convergence entre Windows 8 et Windows Phone 8. En effet, les deux systèmes partageront fonctions, technologies, librairies, API. Ainsi, nous aurons les mêmes piles réseaux, un noyau système identique... Tout cela doit faciliter le développement d'applications et aider le développeur à réduire le portage entre tablette, smartphone et desktop, tout en garantissant une expérience utilisateur la plus homogène possible entre les différents terminaux.

Cette nouvelle génération de mobile va imposer de nouveaux éléments matériels : processeur à cœur multiple, NFC, puces graphiques plus puissantes, multiples résolutions et tailles d'écrans. Les constructeurs pourront développer des pilotes identiques ou quasi. Ainsi sur la partie graphique, on disposera de DirectX, Direct3D. Il est d'ailleurs à noter que le modèle de développement évolue aussi :

- XAML / VB / C# pour les applications Metro
- Développement C, C++ pour les jeux (uniquement ?)
- Html 5 / javascript pour les sites, les applications web



Le retour du code natif était attendu depuis la sortie de Windows Phone 7. Les jeux étant exigeants, ce retour n'est pas donc une surprise, surtout quand il faut manipuler DirectX. On pourra tourner sur desktop et smartphone avec le même jeu (attention tout de même à l'ergonomie et au gameplay). On disposera d'Internet Explorer 10, de la technologie de cartographie de Nokia. Visual Studio 2012 sera l'IDE par excellence. L'entreprise n'a pas été oubliée. Windows Phone 7.x l'avait largement négligée. Windows Phone 8 ne veut pas rater le marché professionnel : administration, cryptage, applications métiers, Office... Le prochain système veut être LE système mobile pour les entreprises. Cependant, Windows Phone 8 ne fonctionnera pas sur les terminaux actuels, des mises à jours 7.x seront disponibles pour rajouter un certain nombre de fonctions. L'agenda n'a pas été précisé. La présentation du 20 juin n'est qu'une première étape.



Et si l'open source créait une tablette totalement ouverte ?

L'objectif de ce concours était de prouver qu'il est possible en moins de 6 mois de proposer une tablette dont l'ensemble des logiciels système et applicatifs est sous licence libre et dont le

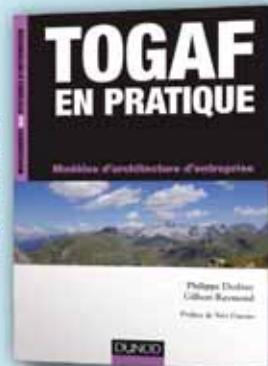
périmètre fonctionnel est compatible avec un usage en entreprise et un usage de loisirs. Les résultats sont tombés le 18 juin dernier. Beaucoup de bons projets : projet Meeto de Ping-Hsun Chen, un projet basé sur NetBSD. Un prix spécial a été accordé à Genymobile pour Android 4.x. Daniel Fages, pour Genymobile nous indique : « Depuis plusieurs mois nous travaillons sur la création d'une distribution Android Open Source sous VirtualBox (www.buildroid.org). C'est grâce à ces travaux que nous avons pu réaliser le projet Android 4.x (Ice Cream Sandwich) pour l'AFULtab. Ce prix est pour nous un encouragement supplémentaire et aussi un moyen pour trouver des contributeurs à notre projet communautaire ». Bravo à toutes les équipes !

Load Tester Pro 5.0 est disponible en France

Le logiciel est distribué par Kapitec Software. Load Tester PROTM scanne automatiquement d'importants volumes de données de performance et analyse les performances d'un site pour déterminer le nombre d'utilisateurs vir-

tuels qu'un site peut supporter et ce, indépendamment du nombre d'utilisateurs ou de serveurs web impliqués. Pour gérer des besoins particuliers de très larges tests, la nouvelle version a été optimisée depuis la base.

DÉVELOPPEZ VOS COMPÉTENCES



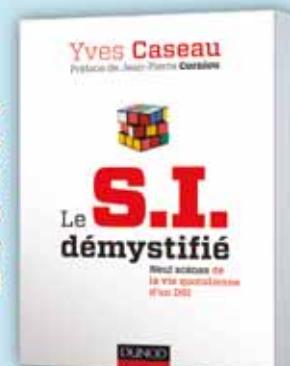
Les modèles d'architecture d'entreprise avec la norme TOGAF

9782100570751 • 35 €



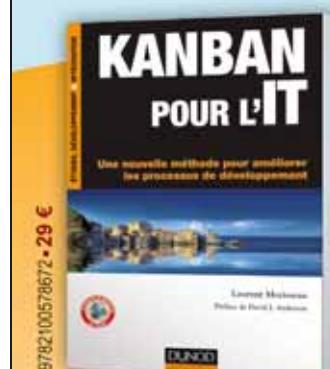
9782101057186 • 39 €

Concevez des applications simples, sûres et adaptables



Les réponses aux questions clés pour ceux qui travaillent avec ou pour un SI

9782100576357 • 24 €



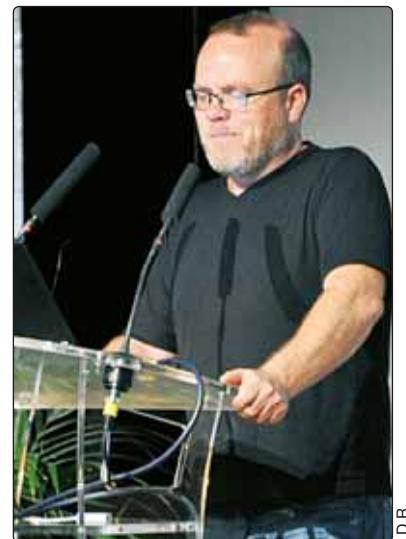
9782100578672 • 29 €

Adaptez les concepts de Kanban dans l'univers du développement logiciel



PHP en 2012 : la mise au point de Rasmus

Après avoir soufflé les bougies de ses 17 années d'existence, le langage PHP se place toujours en première position parmi les langages du web et continue à progresser. Aujourd'hui, le taux d'utilisation serait d'environ 78 %(*) c'est pourquoi il était important de revenir sur la conférence que son créateur, 'Rasmus', a tenu pendant le Forum PHP 2012.



D.R.

Rasmus Lerdorf, créateur du langage PHP, explique que le continent qui utilise le plus ce langage est l'Asie, par l'intermédiaire des différents moteurs de recherches (comme Yahoo) et de son mode de distribution. Cependant il ne faut pas perdre de vue, que le succès est mondial, et qu'il est lié à son utilisation par différents sites internet comme Flickr, Facebook, Ebay... Mais aussi, indirectement, aux différentes couches d'utilisation comme les plateformes de blogging : WordPress, Digg....

Lors de sa présentation, Rasmus Lerdorf a donné les grands repères qu'il faut retenir car ce sont les grosses évolutions qui ont fait date et que vous utilisez tous les jours. Tout d'abord, l'année de lancement du langage en 1993, quand il souhaitait rendre une page web dynamique, avec un résultat loin d'être parfait, et que l'on peut considérer comme la version 1. L'année suivante (1994), la version 2 de PHP fut libérée. Mais le réel démarrage remonte à 1995. Aujourd'hui, ce langage a énormément évolué, et propose aussi bien de la programmation procédurale que de l'objet, mais aussi la possibilité d'utiliser des classes, les exceptions, PDO... De nombreuses évolutions qui sont apparues au fil des ans.

> Pourquoi PHP est-il populaire ?

Rasmus explique que la popularité du langage est liée au grand nombre de fonctionnalités et à sa facilité à résoudre les problèmes du web, que son éco-système est gratuit, et qu'il est le premier à avoir proposé cette solution. En plus, de son succès, il est maintenu par une importante communauté, très active, qui participe à son développement, avec de nombreuses aides en ligne pour en faire.

bénéficier l'ensemble des utilisateurs. Par ailleurs, PHP est une brique et s'intègre sur n'importe quel type de serveur web ou système d'exploitation. Mais grâce à la simplicité d'installation, il peut fonctionner sur un serveur d'hébergement mutualisé, permettant ainsi d'avoir de multi-instances en parallèle. Mais aussi bien sûr un serveur dédié. Même si le succès de PHP va (sans doute) continuer, il ne pourra pas se limiter seulement au Web. La prochaine étape pour PHP sera l'utilisation du langage dans des applications métiers pour devenir un « standard ».

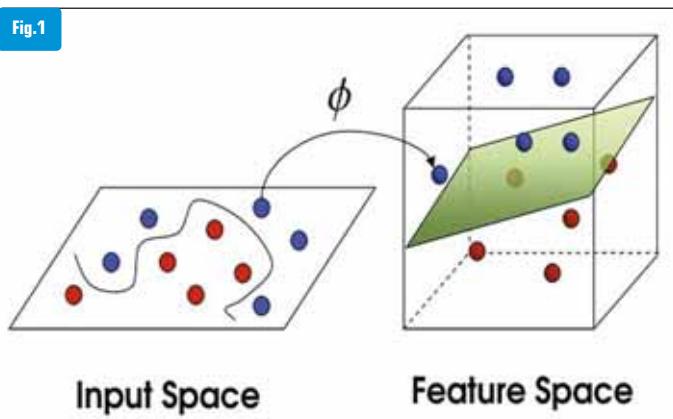
> La diversité

La vision présente et future pour Rasmus Lerdorf, c'est que le langage PHP devienne assez mature pour être utilisé dans des secteurs encore peu « explorés ». C'est pourquoi lors de sa conférence, présentée au Forum PHP 2012, organisé par l'AFUP (Association Française des Utilisateurs de PHP), il s'est attardé sur différents projets qui pointent leur nez et qu'il ne faut pas perdre de vue.

LibEvent + ZMQ

Ces 2 bibliothèques séparées permettent, pour LibEvent, d'être une bibliothèque asynchrone d'avis d'évènements, et pour ZMQ (ZeroMQ) de créer des applications clients-serveurs. Mais si vous associez ces 2 bibliothèques, vous pouvez piloter les événements du serveur, c'est-à-dire directement, vous pouvez interroger 10 ports en même temps comme les ports 8000 à 8009, et vous pouvez envoyer des milliers de requêtes simultanées sans troubler le serveur.

Fig. 1



(*) même s'il est très difficile d'établir un %. ndlr

```
<?php
$loop = ReactEventLoopFactory::create();

for($i=0; $i<10; ++$i) {
    $s=stream_socket_server('tcp://127.0.0.1:'.(8000+$i));
    $loop->addReadStream($s, function ($s) use ($i) {
        $c=stream_socket_accept($s);
        $len=strlen($i)+4;
        fwrite($c,"HTTP/1.1 200 OK\r\n".
                  "Content-Length: $len\r\n\r\n".
                  "Response from server $i\r\n");
        echo "Served on server number $i\r\n";
    });
}

```

```
$loop->run();
?>
```

Ainsi, si vous utilisez cette technique, vous pouvez envoyer des multi fichiers en simultané

Event Abstraction (<http://nodephp.org>)

En s'appuyant sur différentes librairies PHP, vous allez pouvoir obtenir une plateforme développée en PHP pour faciliter la construction d'applications rapides et scalables.

Support Vector Machine (SVM)

Il s'agit d'un autre type de librairie, car il s'agit d'un solveur efficace pour les problèmes de classification et de régression SVM. Il peut se représenter comme sur la [Fig.1].

> Et l'avenir

PHP 5.4

2012 est une année importante pour le langage PHP, car la version PHP 5.4 est disponible, ce que Rasmus n'a pas oublié de mentionner. Les points importants qui ont été insérés dans cette version concernent la performance, l'incorporation des traits, les closures, les fermetures statiques, un serveur de test intégré, des améliorations liées aux messages d'erreurs et au JSON. L'ensemble de ces points et évolutions ont été traités dans les précédents numéros du magazine et il est important de bien les maîtriser.

Le futur

Rasmus explique que même si une nouvelle version vient de sortir, il est important de se projeter sur les années futures. Et c'est bien entendu, dans cette logique que différents chantiers sont en cours et qui touchent :

- Unicode
- La compatibilité avec les tendances du web
- Les améliorations de la traduction
- Le partage de connaissances en améliorant l'existant

> Conclusion

Rasmus Lerdorf conseille très fortement de passer à la version PHP 5.3 si vous utilisez encore une version antérieure. Et si vous le pouvez (ce qui est préférable), de passer à la version PHP 5.4 pour permettre aux équipes qui maintiennent les anciennes versions de se consacrer aux dernières versions du langage.

Par ailleurs, lors de ses différentes rencontres à travers le monde, il rappelle que nous sommes tous des utilisateurs du langage, mais aussi des contributeurs et que si vous souhaitez vous impliquer ou participer dans les différents projets qui sont lancés par l'équipe PHP, il ne faut pas avoir peur de vous investir pour faire évoluer le langage, même en créant des librairies ou extensions pour le langage. Enfin, l'investissement peut aussi se traduire par la mise en place de tests, participer à la documentation, trouver des bugs, tester, etc.

Christophe Villeneuve : consultant pour Alter Way solutions, auteur du livre « PHP & MySQL-MySQLi-PDO, Construisez votre application », aux Éditions ENI. Rédacteur pour nexen.net, membre des Teams DrupalFR, AFUP, LeMug.fr, PHPtv.

Smile

OPEN SOURCE SOLUTIONS

1er

intégrateur européen de solutions open source

Livre blanc

Industrialiser ses développements avec l'ALM



Gestion du Cycle de vie des Applications (ALM)

Le meilleur des solutions open source

Smile

OPEN SOURCE SOLUTIONS

www.smile.fr • +33 (0)1 41 40 11 00 • contact@smile.fr
www.smile.com • blog.smile.fr • smile@GroupedIn.fr

Panorama des des différents concepts et bénéfices de l'ALM open source

A télécharger gratuitement sur www.smile.fr

open

source

www.smile.fr

Forum PHP 2012 : Zend Framework 2 et 400 personnes !

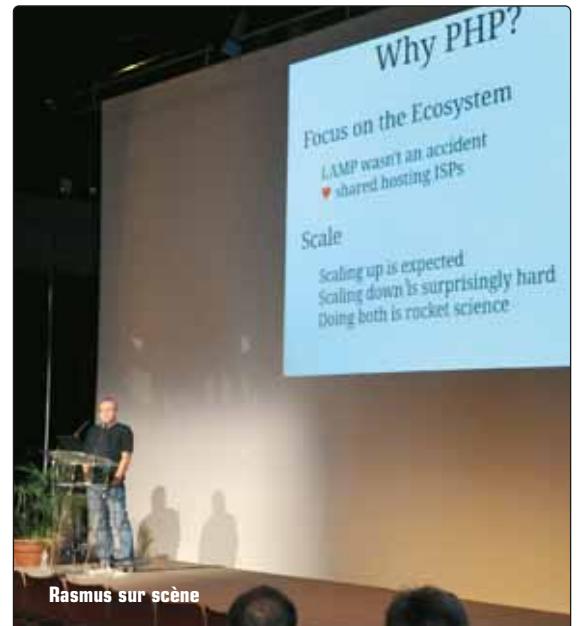
Le Forum PHP est de retour ! En 2011, l'AFUP ne l'avait pas organisé pour différentes raisons et notamment pour pouvoir lancer le PHP Tour, événement organisé en province. A partir de 2012, le Forum et le Tour marqueront la communauté PHP en France. Pour ce cru 2012, le forum signe un retour un peu en baisse avec environ 400 personnes présentes, (Il faut dire que la dernière édition, +500 participants, était un peu spéciale avec les 10 ans de PHP). Et il a été immédiatement suivi d'un autre événement majeur PHP : Symfony Live.

Organisé dans la magnifique Cité Universitaire de Paris, le Forum PHP a mis en avant la communauté et l'écosystème PHP avec plusieurs partenaires historiques mais aussi de nouveaux soutiens comme M6, Microsoft avec Windows Azure. Rasmus Lerdorf a, bien entendu, été une des vedettes de l'événement. Il en a profité pour recadrer l'évolution du langage et la version 5.4. Pour Rasmus, PHP 6 ne verra pas le jour dans son concept d'origine. Les nouveautés seront intégrées dans les itérations de la branche 5.x pour les introduire en douceur et surtout pour éviter les problèmes techniques d'intégration. Mais Rasmus a aussi rappelé qu'il était important de migrer vers les versions récentes du langage car plus les développeurs tardent à migrer, plus la communauté doit supporter des versions anciennes et cela retarde l'ensemble du projet. Pour plus de détails, voir, pages précédentes.

> Qualité, qualité et aussi qualité

Trois thèmes étaient en fil rouge de ce Forum : les nouveaux usages, la qualité / les tests et les performances. Plusieurs sessions techniques abordaient ces thématiques : anatomie du test, monitorer une application, automatiser la qualité, accès concurrent et montée en charge. Mais les organisateurs voulaient ouvrir le Forum PHP 2012 aux entreprises et surtout aux DSIs pour montrer que PHP n'est pas un langage bas de gamme, sans outils d'industrialisation. Plusieurs DSIs (TF1, M6 Web) ont fait le déplacement pour évoquer l'usage de PHP en entreprise et sur des projets importants...

Zend a profité du Forum pour présenter Zend Framework 2.0 qui sortira courant de l'été (la bêta 5 est sortie en juin). Pour cette nouvelle version, l'éditeur a voulu renouveler en profondeur son framework. Pour l'éditeur



Rasmus sur scène

D.R.

c'est un nouveau projet. Les développeurs ont fait remonter de nombreux feedbacks depuis la v1. Zend Framework s'appuie sur PHP 5.3 (cela ne devrait pas changer). La 5.4 n'est pas officiellement supportée. Cette v2 se veut dans la continuité de la v1 et se focalise sur : extensibilité, performances, amélioration de l'apprentissage et la prise en main, réécriture des parties du code qui le nécessitent. Sur la partie extensibilité, on notera la programmation par contrat, la suppression des dépendances codées en dur. Une des grosses nouveautés sera l'autoloading qui doit améliorer les performances globales du framework. On pourra utiliser les espaces de noms de PHP 5.3 et d'une partie MVC entièrement réécrite. Les nouveautés de PHP 5.3 seront présentes et exploitées. Le cloud n'est pas oublié avec ZendCloud, SimplePHP API et le lancement de PHP Cloud. L'objectif de Zend est de faciliter le développement et le déploiement de solutions PHP sur les différents clouds (IaaS ou PaaS), notamment sous Amazon EC2, Rackspace ou encore Windows Azure.

Notons aussi la présence de Theodo qui propose de moderniser les applications PHP anciennes en utilisant les dernières versions du langage et l'agilité Scrum... On oublie que les applications PHP ont elles aussi besoin d'être migrées et modernisées de temps en temps.

Symfony Live : la communauté plus dynamique que jamais !

SensioLabs organisait sa grande conférence technique à Paris : Symfony Live. Un très gros succès : 600 participants ! Si les Français étaient majoritaires, de nombreux Européens ont fait le déplacement jusqu'à Paris. Conséquence : la conférence se déroulait en Anglais (avec traduction). Fabien Potencier a fait l'ouverture et mis surtout en avant la communauté, l'écosystème et le dynamisme des développeurs et partenaires autour de Symfony dont la v2 est sortie l'an dernier. Preuve de ce dynamisme : plus de 1000 demandes de corrections, d'ajouts dans le framework en un an. Fabien n'a pas voulu faire de prospective sur le Symfony. L'objectif est maintenant de faire vivre l'écosystème et d'aider les projets à se monter. Pour appuyer cela, SensioLabs a mis en place une certification pointue, demande récurrente de la communauté et des entreprises. La certification se veut difficile et de haut niveau ! Symfony Live a aussi mis en avant les nouvelles relations entre le framework et Drupal. Les objets Symfony peuvent être implémentés dans Drupal et le code Drupal dans un projet Symfony.

SSD : l'arme ultime des performances

Inutile de rappeler une évidence de l'informatique : le processeur, aussi performant soit-il, n'est rien si l'ensemble des sous-systèmes, de la carte mère aux périphériques, n'est pas optimisé et adapté aux performances. Le disque de stockage SSD est un réel atout au quotidien. Mais ce n'est qu'un élément de la chaîne de production d'un développeur...

Le SSD a plusieurs avantages :

- Aucun mécanisme : assurant ainsi des performances optimales et une consommation moindre
 - Moins sensible au choc : pas de risque de désaligner les têtes de lecture
 - Des performances optimales (en théorie)
- Par contre, le SSD souffre de plusieurs problèmes :
- Une durée de vie incertaine : notamment due aux multiples lectures / écritures des cellules
 - Prix au Go encore élevé (mais en baisse)
 - Support de ces disques par les systèmes

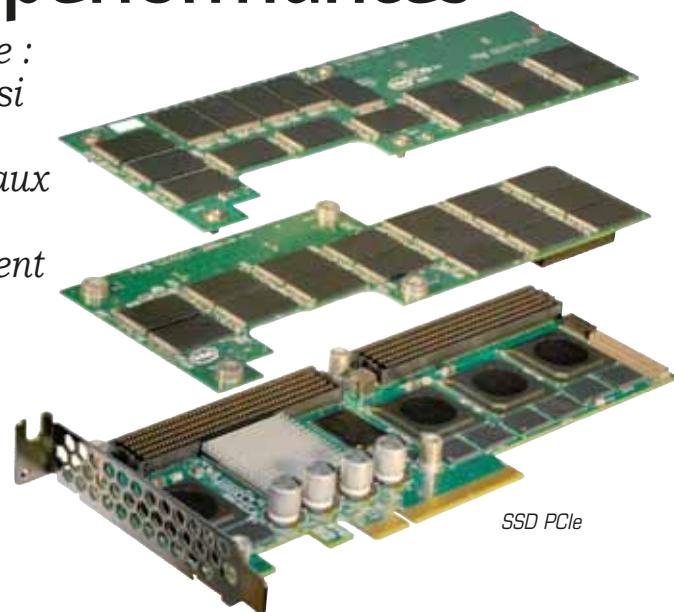
> Des critères trop souvent oubliés

Devant la multiplication et le renouvellement (très rapide) des modèles SSD par les

constructeurs, un utilisateur non averti s'y perd vite. Plusieurs critères sont importants à considérer :

- Débit de données par seconde (ex. : 150 Mo/s) en lecture et en écriture : les deux critères sont importants pour estimer les performances globales du SSD
- Le type d'interface du disque (ex. : SATA)
- Bien surveiller le IOPS : input / output par seconde

Quand vous lisez « jusqu'à 2 fois plus rapide qu'un disque dur SATA 7200 tours par minute », intrinsèquement, cela ne veut pas dire grand-chose. Le tour / minute est un critère



de performance d'un disque dur, mais non comparable avec un SSD qui n'en possède pas. Il faut comparer les performances en lecture / écriture, les IOPS.

Intel conseille de surveiller tout particulièrement deux critères :

1/ Vitesse de transfert : la norme SATA III garantit une vitesse de transfert en écriture et en lecture élevée. Cependant, tous les SSD en SATA III ne proposeront pas les mêmes performances, il faut également prendre en compte le type de puces Flash et le contrôleur (ou son firmware) qui les composent.

2/ Garantie : Certains SSD Intel ont jusqu'à 5 ans de garantie, les rendant très sûrs dans le cadre d'un usage en production.

> TRIM obligatoire

La fonction TRIM est devenue indispensable pour ralentir les pertes de performances et l'usure des cellules du SSD. Car, l'un des problèmes du SSD est le mode de stockage des données et la façon de savoir si la donnée d'une cellule est toujours là ou non. Cela influence la manière dont le disque flash recherche les cellules et efface les données et répercute l'information au système d'exploitation. Plus une cellule subit des écritures et des effacements, plus il faut une gestion performante des cellules car le contrôleur perd ses repères et les performances tombent. Autrement, comment savoir si une cellule est effectivement libre... Le TRIM aide le contrôleur à connaître les cellules libres (surtout quand on efface des fichiers).

Test de l'hybride Iomega StorCenter

Nous avons pu tester quelques jours un StorCenter px6-300d de Iomega. Il était équipé de deux disques et 2 SSD, pour une capacité totale de 4 To. Le boîtier nu revient à 640 € - 700 € (suivant les revendeurs). Ce NAS contient 6 baies, une connectique classique (USB dont 1 USB 3, ethernet), un processeur atom (deux cœurs et 2 Go de ram). La construction est soignée, hormis la porte avant qui aurait mérité un matériau de meilleure qualité. Bon point : l'encoche de sécurité à l'arrière. Il peut monter jusqu'à 18 To et supporte par défaut les formats 2,5" et 3,5". Côté RAID, là aussi, c'est le standard et le bon : raid 1, 10, 5 et 6, avec remplacement à chaud des disques et reconstruction automatique du RAID. Le SSD est prévu par défaut par le constructeur. Nous avons apprécié sa compatibilité Time Machine (idéal sous OS X) et la certification virtualisation. L'écran de contrôle est un plus indéniable avec la disponibilité



des informations essentielles (capacité, remplissage, IP). Côté application, on retrouve le Iomega Storage Manager. Simple et assez complet.

Nous le conseillons en unité de stockage de répertoires de code et de sauvegarde quotidienne, couplé bien entendu à un onduleur.



Indice de performances

Nous avons effectué plusieurs tests pratiques sur différents SSD (internes, externes, NAS). Les résultats sont contrastés mais finalement logiques, surtout en NAS où le SSD ne peut s'exprimer pleinement. On remarque aussi que la connectique sera primordiale pour les performances. Mieux vaut privilégier un USB 3 ou du Thunderbolt (ou un Firewire 800).

SSD (NAS iomega)	8 minutes 18 secondes
Disque dur (NAS iomega)	8 minutes 18 secondes
SSD externe 64 Go (USB 2, iomega)	11 minutes 24 secondes
SSD interne 256 Go Crucial m4	6 minutes 39 secondes

Copie d'une archive de 16 Go, connecté en ethernet sous OS X 10.7. SSD externe connecté en USB 2. Machine de test : MacPro 8 cœurs.

Démarrage système (OS X 10.7 à jour, sur MacPro) :

Disque dur interne 1 to (origine Apple)	39,1 secondes
SSD interne 256 Go Crucial m4	25,5 secondes

> SSD système et applications : un usage classique

Aujourd'hui, avoir un SSD comme disque système et pour les applications est quelque chose de relativement courant, surtout sur les portables. Par rapport à un bon disque dur, le gain de temps de démarrage système peut aller jusqu'à 50-60 %, en moyenne, nous constatons, en usage quotidien, un gain de 30 % (base : OS X Lion sur MacPro). Même amélioration sur les applications (lancement, exécution de certaines fonctions où

les E/S sont essentielles). Pour les applications lourdes et très exigeantes en lecture / écriture et accès disque, le SSD est incontournable mais ne fera pas tout : le reste de la machine doit suivre. Cependant, revenir à un disque dur classique quand on a goûté à

du SSD, tourne vite au cauchemar : « qu'est-ce que ça rame ! » (pour rester poli). Vous pouvez aussi regarder le niveau de SATA et surtout s'il s'agit d'un SATA 6 Gbps. Le SAS est aussi utilisé, dans ce cas privilégié le SAS 6 Gbps.

[Suite page 18](#)



Questions – réponses avec Laurent Sirgy

(directeur France, Europe du Sud, MEA
Kingston Technology)

Programmez ! : Quels critères doit-on considérer pour choisir un SSD ?

L.S. : voici les critères à prendre en considération en allant du plus important au moins important : La performance, quelle performance peut fournir le SSD en fonction du type de données ? (compressibles, incompressibles), la taille (petits fichiers, gros fichiers) de l'interface SATA sur votre ordinateur hôte pour soutenir et profiter de la performance exceptionnelle du SSD. Ensuite, l'endurance - combien

de Go par jour pouvez-vous écrire sur le disque SSD sans user la mémoire flash NAND. Et enfin, la capacité. Quelle quantité de données avez-vous besoin de stocker sur le SSD, et si possible, tous les types de données utilisées dans les applications doivent aussi être stockées sur le SSD pour profiter de la performance sur tous les aspects.

Programmez ! : On parle beaucoup de la fonction TRIM. Est-ce l'unique point à surveiller ?

L.S. : les SSD peuvent gérer les données internes via des processus internes tels que la collecte des ordures et les opérations de fusion de données, même en l'absence du soutien TRIM dans le

système d'exploitation. Il est toujours recommandé d'avoir le soutien TRIM en plus de la collecte des ordures internes du processeur SSD, en maintenant la performance sur le long terme et de la meilleure manière possible.

Programmez ! : Les développeurs utilisent régulièrement des environnements logiciels lourds nécessitant de bonnes performances, sans compter le traitement de nombreuses données. Le SSD est-il là aussi une solution pour les aider au quotidien ?

L.S. : les SSD sont une excellente solution pour traiter tout type de donnée à des vitesses rapides par rapport à un disque dur classique qui

limite les performances globales du système et de l'application qu'utilisent les développeurs.

Programmez ! : Quelle différence fondamentale entre un disque SSD et du SSD en PCI ?

L.S. : les SSD traditionnels se trouvent sous la forme 2.5" avec le support d'interface SATA. Ils sont plus faciles à mettre à jour, à installer et à entretenir que les SSD PCIe qui ne peuvent pas forcément soutenir l'installation et l'hébergement d'un système d'exploitation, et demandent parfois des pilotes particuliers et slots PCIe, avec suffisamment de bande passante pour profiter de leur performance.

VITE, ABONNEZ-VOUS !

jusqu'à **-50%**

Code, gestion de projets, développement web, mobilité...

Programmez ! est à la fois votre outil pratique : articles de code par les meilleurs experts, et aussi votre veille technologique !



1

Abonnement 1 an au magazine

49 € (au lieu de 65,45 €, prix au numéro)

2

Abonnement Intégral : 1 an au magazine + Archives Internet et PDF

59 €

3

Abonnement 2 ans au magazine

78 € (au lieu de 130,90 €, prix au numéro)

4

Abonnement intégral 2 ans au magazine + Archives Internet et PDF

88 €



Toutes les offres en ligne : www.programmez.com

Abonnez-vous à partir de 3,80 € seulement par mois



Oui, je m'abonne

à retourner avec votre règlement à
Groupe GLI, 17 route des Boulanger 78926 Yvelines cedex 9

- Abonnement 1 an au magazine** : 49 € (au lieu de 65,45 €, prix au numéro)
- Abonnement Intégral : 1 an au magazine + archives Internet et PDF** : 59 € (au lieu de 65,45 €, prix au numéro)
- Abonnement 2 ans au magazine** : 78 € (au lieu de 130,90 €, prix au numéro)
- Abonnement intégral 2 ans au magazine + archives Internet et PDF** : 88 €

M. Mme Mlle Entreprise : _____ Fonction : _____

Prénom : _____ Nom : _____

Adresse : _____

Code postal : _____ Ville : _____

(Attention, e-mail indispensable pour les archives sur internet)

E-mail : _____ @ _____

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

Usages types	SSD oui/non
Données froides / archivages	non
Données chaudes	oui
Applications de production au quotidien,	oui
Applications lourdes	oui
Disque système et disque application traitement de données, build, Compilation, etc.	oui

> Quelle topologie de disque ?

La plupart du temps, vous utilisez le SSD en interne à la place ou en plus d'un disque dur classique. Vous pouvez aussi opter pour du SSD en carte PCIe. Ce sont souvent des solutions haut de gamme pour serveur et le stockage d'entreprise mais les performances peuvent exploser.

Cependant, ces solutions sont rarement abordables (ex. : OCZ RevoDrive 3 960 Go : + 2 210 €) mais les capacités dépassent facilement 500 – 600 Go ! Vous pouvez passer par des SSD externes. Aujourd'hui, nous trouvons des solutions de stockage de type NAS utilisant de l'hybride disque dur et SSD ou 100 % SSD. Cependant, ce type d'utilisation des disques Flash concernera le stockage de données chaudes ou régulièrement utilisées. Les caractéristiques du NAS vont limiter les performances (processeur interne, type de connexion).

Astuce : n'oubliez jamais le kit d'installation si la baie disque est en 3,5", le SSD est en 2,5" ! C'est tout bête, mais que de temps perdu sinon !

François Tonic

Un dock SSD chez xilence

Le constructeur de boîtiers et ventilateurs propose un dock deux baies pour SSD : SSD Zub/XPSSD, couplé à un lecteur de cartes. Il s'installe dans tout PC



ayant une baie 5,25". Les disques sont rackables à chaud. Des ports USB 3 sont disponibles en façade avant. Prix agressif : 24,90 €.



Questions – réponses avec Nicolas Maigne (Responsable du Développement Monde Crucial Europe du Sud)

En deux mots quels sont les avantages du

SSD par rapport à un disque dur classique ?

N.M. : la fiabilité et la vitesse. La technologie Solid State a énormément progressé ces dernières années et est désormais considérée comme aussi fiable, voire davantage, que la technologie magnétique. Concernant la durabilité, le fait que les SSD ne comprennent pas de pièces mécaniques les rend beaucoup plus robustes que les disques durs magnétiques traditionnels.

Nous avons récemment fait état d'une étude indépendante, qui plaçait les SSD très favorablement en termes de taux de pannes par rapport aux disques magnétiques traditionnels et qui plaçait Crucial comme la marque avec le taux de pannes le plus bas sur les SSD ! Évidemment, les SSD sont aussi beaucoup plus rapides en écriture, lecture ainsi que temps d'accès que les disques magnétiques.

C'est la raison principale de leur succès, on peut dire, avec la robustesse (il existe des environnements où le disque magnétique ne peut pas être utilisé du fait des vibrations ou des accélérations, pensez aux avions, aux stations pétrolières, etc.). Cela se traduit par une expérience utilisateur PC/portable ou serveur améliorée, notamment pour toutes les applications gourmandes en ressources machines, où le disque magnétique traditionnel pouvait faire figure de goulot d'étranglement. Les jeux vidéo, les applications multimédias, les solutions de CAD/CAM ou de modélisation statistique, entre autres, bénéficient largement de la puissance de la technologie SSD.

Il y a quelques mois, Crucial a corrigé un bug assez étonnant lié à un problème de performances et de bon fonctionnement après x heures. Comment expliquer ce bug, et le SSD sans bon firmware sert-il à quelque chose ?

N.M. : le firmware est en effet un élé-

ment primordial de tout bon SSD. La NAND Flash et les contrôleurs de qualité sont des composants relativement standard et accessibles à de nombreux fabricants de SSD : ce qui fait la différence c'est la capacité d'un Micron (la maison mère de Crucial) de combiner au mieux la NAND et le contrôleur via le firmware. Concernant le "bug" que vous mentionnez, nous l'avons réparé très rapidement via une update de firmware dont ont pu bénéficier tous les possesseurs de m4, et il n'a pas occasionné de pertes de données chez les utilisateurs.

En production, le SSD est soumis à de fortes charges, avec beaucoup de E/S sur les données et en lecture / écriture. Quelles sont les astuces ou les précautions à prendre pour garder de bonnes performances de son disque ? On parle beaucoup du TRIM par exemple.

N.M. : TRIM concerne tous les SSD. Un système d'exploitation supportant TRIM peut identifier un appareil de stockage comme un SSD et sait arrêter des fonctionnalités comme la défragmentation (utilisée pour un disque dur). Il permet aussi l'utilisation de la commande TRIM, celle-ci permet à un système d'exploitation d'informer un SSD des blocs de données qui ne sont plus considérés en utilisation et peuvent être effacés. TRIM permet le "recyclage des déchets" en quelque sorte (fonction "garbage collection"). Fondamentalement, sans cette caractéristique, un SSD ralentirait de manière significative dans les futures opérations d'écriture.

Notre firmware va au-delà de TRIM et fait de la maintenance automatique en arrière-plan pour garantir des performances optimales.

Contrairement aux disques durs qui exigeaient des utilisateurs de lancer des processus tels que la défragmentation, l'utilisateur de nos SSD n'a aucune opération à entreprendre pour garantir la pérennité des performances de son produit dans la durée.

Notre sélection de livres

CHRISTOPHE BLAES



Solutions temps réel sous Linux

Yrolles

Linux et le temps réel est une réalité indiscutable. L'auteur explique en profondeur comment fonctionnent les fondamentaux du temps réel sous Linux et de quelle manière il est possible de créer sa distribution Linux. Pour y arriver, l'ouvrage parle kernel, multitâche, états, les threads et appels, etc. 50 exercices permettent de mieux comprendre le temps réel Linux.

COLLECTIF,



iOS 5

Pearson

En attendant la sortie de iOS 6, iOS 5 couvre des fonctions avancées : créer des tables, implémenter un cache rapide, Core Text, contrôler finalement le multitâche, les nouvelles API, les échanges REST, langage avancé, Grand Central Dispatch. Une excellente base pour aller plus loin dans le langage et le système. Allez-y maintenant !

agenda

» JUILLET

- Du 02 juillet au 04 juillet 2012, Polytech' Nice Sophia, **Sophiaconf 2012** : le cru Open data. On ne change pas une formule qui gagne ! Pour la troisième année consécutive, passionnés et professionnels sophipolitains de l'Open Source se donnent rendez-vous du 2 au 4 juillet pour SophiaConf 2012. Trois conférences ; trois jours d'animation technique, méthodologique et sociétale, et un focus cette année sur un sujet en pleine expansion : l'Open data.

www.sophiaconf.fr

- Le 04 juillet 2012, Roland Garros Club des Loges 8 bd d'Auteuil 92100 - Boulogne Billancourt, **Forum Oracle Fusion Middleware 2012**. Le Forum Oracle Fusion Middleware est de retour sur les courts de Roland Garros ! Venez partager les retours d'expériences de la Direction Générale de l'Aviation Civile, Kiloutou, SNCF Voyages, SI et Société Générale, et découvrez leurs stratégies gagnantes pour transformer leurs SI d'un revers. Inscription : http://storage.dolist.net/2384/www/20120518-jm/Coupon_SILICON.html



Franck EBEL Hacking et Forensic

éditions ENI

Le forensic est une approche de l'investigation cruciale pour remonter les intrusions et comprendre les méthodes, les outils utilisés. L'ouvrage aborde la sécurité, le forensic et la conception d'applications en Python. Il a pour objectif de conduire le lecteur à une bonne compréhension de bibliothèques spécifiques Python pour qu'il puisse ensuite concevoir ses outils personnalisés, adaptés à des situations particulières. L'auteur aborde la programmation réseau, la bibliothèque scapy, le guzzing, PIL, Sulley. Un must de la sécurité informatique à lire et à relire !



LUC VAN LANCKER Les API Javascript du HTML 5

éditions ENI

L'auteur propose une exploration de ces nombreuses API JavaScript, certaines pleinement opérationnelles, d'autres encore en phase de développement. Le HTML5 étant une évolution de portée considérable qui modifie totalement la conception des pages ou applications Web, l'auteur a veillé à adopter une approche pragmatique et explicative, illustrée de nombreux exemples et captures d'écran. Divers sujets sont abordés tout au long du livre : géolocalisation, history, canvas, drag and drop, fichier, offline, etc.

JOELLE DE LACROIX

Linux programmation système et réseau 4e édition



Dunod

Idéal pour maîtriser les arcanes Linux, les piles réseaux. Il présente les principes fondamentaux des systèmes d'exploitation ainsi que la programmation système sous

les systèmes d'exploitation Linux, conformes à la norme POSIX sans oublier tout ce qui est synchronisation, gestion des messages, signaux, etc. Cette 4e édition met à jour les exercices et les corrigés.

ERIC CUBIZOLLE



La Bible Amiga

Pix'n love

Nostalgie quand tu nous tiens ! Voici un livre unique en son genre : 100 % Amiga ! Ce formidable ordinateur a passionné les gamers et les développeurs. Quoi de plus naturel alors, au regard d'un tel panache, que de raconter la fabuleuse histoire de cet ordinateur ludique à travers un historique complet et ponctué d'interviews. Impossible également de passer outre l'abondante logithèque de cette fabuleuse machine qui a su aussi bien susciter l'intérêt qu'attiser la convoitise. L'auteur évoque plus de 1200 jeux ! De quoi se souvenir des heures passées devant son moniteur...

» Communauté

Groupe utilisateurs Java de la ville de Cergy et sa région

L'objectif de cette communauté est de réunir les utilisateurs Java (Java, JEE, JavaFX, Android, Eclipse...), de tous les niveaux, afin de discuter techniques, partager son expérience, en organisant des événements, meetings et formations. Le leader du groupe (Kaesar ALNIJRES) est un expert Java, certifié Sun Microsystems (csjp). Le jug cergy est assez récent mais la communauté ne demande qu'à croître. Il est prévu de mieux structurer l'association et de relancer les événements et réunions à partir de septembre.

Contact: jug.cergy@gmail.com - Site: <http://formations.informatique-cergy.org/2011/08/java-user-group-cergy.html>

» Quelques events en Juillet

3 juillet à Paris

ParisJUG organise une soirée spéciale NoSQL. Les détails sur : <http://www.parisjug.org/xwiki/bin/view/Meeting/Next>

4 juillet à Issy les Moulineaux

La communauté développeurs Windows 8 va être officiellement lancée début juillet au Campus de Microsoft France. Plusieurs sessions techniques seront jouées tout l'après-midi, avec un soupçon de Windows Azure et SignalR. De 14 à 18h.

Pour en savoir plus : <https://www.facebook.com/events/339671249422183/>



Frédéric Pedro, développeur, prof et chef de projet !

Laissons Frédéric se présenter : « je suis président de POHLM Studio, société de développement de jeu vidéo de 25 personnes basée à Angoulême. Diplômé de SUPINFO, j'ai monté cette entreprise dans la continuité de l'Imagine CUP, "Les jeux Olympiques Numériques Etudiants". C'est une compétition qui m'a vraiment apporté beaucoup : 4^e mondial en 2008, Coach Français des 3^e en 2011, Mentor de l'équipe Brainergy qui a gagné la compétition mondiale en 2011.

J'ai été aussi chef de projet sur de gros projets (30 à 80 personnes) et j'enseigne dans les Master 2 de Paris Dauphine, Leonard de Vinci et d'Ingésup. Enfin, vous avez peut-être pu lire mes contributions passées dans Programmez (:D). »

Comment es-tu tombé dans l'informatique et plus spécialement dans le développement ?

Mon oncle m'a transmis petit la passion pour les jeux vidéo. Mes parents ne roulaient pas sur l'or et je n'ai pu accéder à un ordinateur qu'au collège et à Internet en fin de lycée. Mais j'ai très vite eu cette "vocation" de vouloir créer des jeux vidéo ! Une passion si vorace que j'avais finalement "appris" le C++ dans des livres et sur papier avant de toucher un ordinateur. Cela a d'ailleurs été assez drôle lors du passage à la pratique de la programmation, j'imaginais un résultat plus visuel et que des "tableaux" de variable allaient s'afficher ... sous forme de tableau !

Pour toi, qu'est-ce qui fait que l'on aime toujours et encore le développement, la technique ?

C'est une drogue ! Au sens littéral du terme. Imaginons que l'on ait à résoudre un bug compliqué. On se fait mal, on s'énerve, on se dit parfois qu'on n'y arrivera jamais, voire qu'il faudrait tout recommencer de zéro, ... et quand vient la solution, c'est presque orgasmique ! Il y a une amplitude d'émotions qu'on ne trouve pas dans tous les métiers. Etre développeur aujourd'hui, c'est savoir faire face aux difficultés, aux challenges et

avoir tous les jours de nouveaux défis à résoudre. Il faut aimer se remettre en question et continuer à apprendre.

Tu as gardé un regard très geek : gadget, veille techno. C'est important pour ton job et ta passion ?

Le jeu vidéo est un formidable vecteur d'innovation : les tablettes, les smartphones, l'essor du "jeu streaming" / TV connectée, l'essor du jeu digital sur consoles, la Wii/Kinect/PSMove, etc. C'est excitant pour un joueur passionné, c'est excitant comme programmeur, c'est excitant pour l'entreprise et l'équipe car l'on doit sans cesse se réinventer.

Etre développeur n'est pas toujours facile : pression, évolution constante, frustration des projets et des "chefs", c'est quoi pour toi être développeur aujourd'hui ? Le job a-t-il changé depuis tes débuts ?

J'ai co-créé la technologie que nous utilisons en interne et co-programmé *Hasta La Muerte*, un des rares jeux français à avoir eu le label Xbox LIVE sur Windows 7 et prévu pour le 28 juin prochain sur Iphone, Ipad, Android et



autres tablettes (PlayBook, etc.). Vu qu'il s'agit d'une startup que j'ai co-fondée, le contexte est un peu particulier. Je connais et dois faire face à toutes les contraintes. Il y a une équipe de 20 personnes qui compte sur nous tous les mois. Dans ce contexte particulier, on acquiert et comprend mieux des responsabilités et une vision que l'on n'a pas normalement : la gestion des plannings et le choix des projets face aux réalités commerciales et à la trésorerie en banque, etc. De toute manière, on a rarement à se plaindre dans notre secteur : la création artistique est interactive. L'équipe prend toujours beaucoup de plaisir à créer un projet.

Dans ce contexte, être programmeur pour moi, c'est aussi avoir une part "artiste" à sa manière, il faut trouver sans cesse des solutions : on ne sait jamais comment réaliser un projet mais on sait toujours que l'on va y arriver. Tous les chemins mènent à Rome. Plus personnellement, je suis un grand fan de Tim Ferris ("La Semaine des 4H"). Les méthodes agiles, "vraiment" appliquées, peuvent déjà apporter beaucoup. Mais je pense que l'on peut souvent automatiser certains process ou tâches (pas uniquement techniques) de programmation pour se concentrer sur l'essentiel. Chez POHLM Studio, notre assistante de direction, avec un niveau Bac et sans notion IT, gère les droits SVN,

“ on ne sait jamais comment réaliser un projet mais on sait toujours que l'on va y arriver ”



bug tracker, du serveur, du Wiki, etc. et supporte l'équipe au quotidien.

Et en dehors du boulot, qu'est-ce que tu aimes faire ? Comment trouves-tu l'équilibre entre travail, vie privée, passion, famille ?

C'est assez difficile pour moi de trouver du temps, je ne reste jamais sans rien faire et j'ai toujours plein de projets que j'ai envie de réaliser. Je suis un grand fan de développement personnel, de psychologie et notamment du Personal MBA. J'aime beaucoup enseigner aussi.

Peux-tu nous présenter ton quotidien en quelques mots ?

C'est difficile à décrire, cela change tous les jours et j'ai des responsabilités élargies (négociation de contrats, tâches administratives, etc.) auxquelles je ne peux me soustraire. Je viens de passer beaucoup de temps la semaine dernière sur le recrutement d'un nouveau programmeur. La semaine prochaine, ce sera l'envoi de la build du jeu "Votez Pour Moi" et de "Brainergy" (jeu

Le bureau de Pedro

Pour moi, les heures de travail ont une grande importance. C'est difficile d'imposer à un programmeur des heures fixes ou de bureau pour des raisons de concentration, etc. même si on a tous conscience que c'est crucial lorsque l'on travaille en équipe. J'aime bien commencer tardivement le matin pour finir tardivement le soir et 2 matinées par semaine de chez moi. Comme beaucoup, j'apprécie énormément ces moments de "tranquillité", c'est-à-dire n'être dérangé par rien ni personne (les mails ne fusent plus, le téléphone ne sonne plus, etc.) et où la productivité est finalement maximale. Je suis quelqu'un qui aime "s'étaler" pour avoir les informations toujours à vue ; j'ai donc deux écrans et un "loongg" bureau pour tout l'administratif. ☺

vainqueur mondial à l'Imagine Cup 2011, sur lequel nous avons financé et co-produit la moitié du développement).

Comment vois-tu ton job évoluer ?

A ce rythme de croissance chez POHLM Studio, je vais devoir m'éloigner de la technique pour n'avoir plus qu'un impact sur les plannings.

Des conseils aux étudiants et dévs qui nous lisent ?

Si un développeur s'investit dans un projet

personnel (ou qui a poussé ses projets étudiants plus loin que demandé), cela montre une démarche volontaire et une passion forte. Quand on reçoit 100 CV d'étudiants de même qualité technique, de même niveau de formation, etc. cela FAIT LA DIFFERENCE. Il y a d'ailleurs beaucoup de demandes sur des nouvelles technologies et donc tout autant de "places à prendre" pour se démarquer : développement Facebook, intégration d'un système de monétisation type TapJoy, Windows 8, Windows Phone 7, etc.

#



Modelio Java Solution

Boostez l'Agilité de vos développements Java avec Modelio 2.2

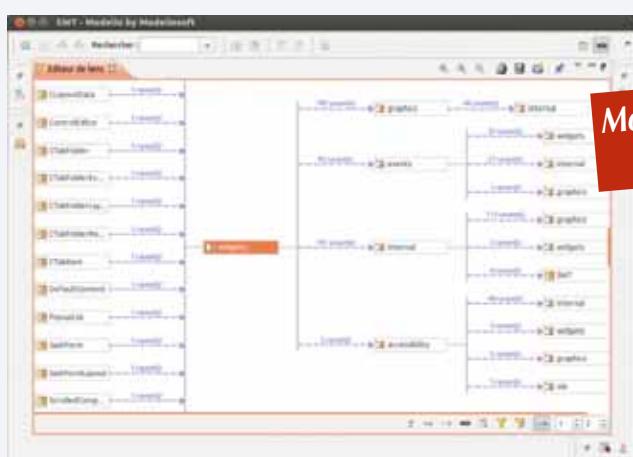
Renforcez votre IDE, Augmentez la productivité, Augmentez la qualité

Développeurs Java

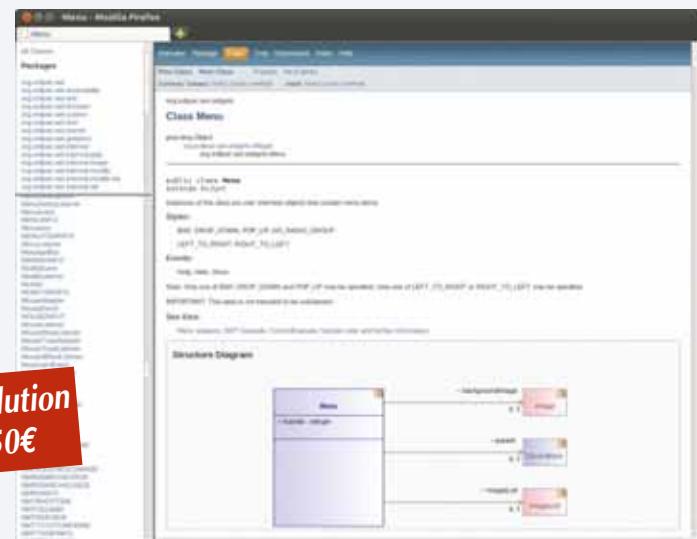
- Mise à jour automatique des méthodes utilitaires (equals, hashCode, toString) sur code existant
- Génération de Javadoc enrichie de diagrammes automatiques UML
- Support Maven pour production code/Java doc et intégration continue

Architectes

- Synchronisation permanente code Java/UML
- Reverse Engineering Java -> UML, modèle et diagrammes
- Support audit/rétro-conception architecture du code
 - Construction automatique des diagrammes d'architecture,
 - Analyse des dépendances inter packages & classes,
 - Exploration graphique des liens du modèle (dépendances, héritage, ...)



**Modelio Java Solution
à partir de 250€**



Agilité accrue pour les développeurs et architectes Java

Modelio augmente les capacités de votre IDE (Eclipse ...) pour automatiquement mettre votre code en cohérence, et en apporter une vue architecturale synchrone. L'architecte peut voir un modèle synthèse du code, explorer et analyser les dépendances, réorganiser le modèle et re-générer. Les Javadocs sont automatiquement enrichies de modèles...

Voir vidéos et exemples sur :

www.modeliosoft.com/fr/a-propos/infos-et-evenements/java-agile.html



sales@modeliosoft.com

Tél. : 01 30 12 18 40

Découvrez et évaluez nos solutions

www.modeliosoft.com

Bases NoSQL et MongoDB

Apparu dans les années 1970, le modèle relationnel des bases de données est aujourd’hui le plus commun. Après l’échec des SGBD objets dans les années 90 - 2000, un autre type de base de données émerge et est de plus en plus utilisé notamment par des sociétés qui doivent gérer de très gros volumes de données (Google, Facebook ...). Il s’agit des bases de données NoSQL.

Cet article commence par décrire le modèle NoSQL pour ensuite le comparer au modèle relationnel. Enfin, le Système de Gestion de Bases de Données (SGBD) NoSQL MongoDB[1] sera présenté.

> Quelques mots sur le NoSQL

Le terme NoSQL (acronyme de Not Only SQL) désigne une famille de SGBD fonctionnant sur un modèle particulier. Ces logiciels sont indépendants du langage SQL et donc fondamentalement différents des logiciels de bases relationnelles. Ce terme a été utilisé pour la première fois en 1998. C'est à ce moment là que de grands acteurs informatiques se sont rendu compte que les SGBD relationnels classiques n'étaient pas adaptés pour un très gros volume de données. Les temps de réponses étaient trop élevés et la distribution des données trop lourde et trop complexe.

Avec l'apparition de nouveaux principes de stockage, de nouveaux modèles ont été développés. On retrouve principalement les quatre modèles suivants:

Le modèle « Clé/valeur »

Le principe de ce modèle est d'associer un identifiant unique à chaque valeur dans la base de données. Une valeur peut aussi bien être de type simple qu'être un objet sérialisé.

Les SGBD qui fonctionnent de cette manière fournissent en général uniquement les quatre opérations basiques : création, lecture, modification, suppression. Toute l'intelligence dans la récupération des données se situe donc dans l'applicatif client.

Le modèle « Base Documentaire »

Dans ce modèle, chaque entrée est appelée « document ». Un document est un ensemble de champs qui possèdent chacun une valeur. Parmi ces champs se trouve un identifiant unique rajouté par le SGBD. Il est intéressant de noter que la valeur d'un champ peut, à son tour, être un document. Un ensemble de documents est appelé une collection (équivalent des tables en relationnel). Voici quelques exemples de documents écrits en JSON (JavaScript Object Notation) :

```
{id : « 1 », nom : « Guitare », matériau : « Bois »}
{id : « 2 », nom : « Batterie », catégorie : « Bruyant », accessoire : « Baguettes »}
```

Ces documents peuvent tout deux appartenir à la même collection. En effet, le modèle de base documentaire n'impose pas aux documents d'une collection d'avoir une structure stricte à respecter. Il s'agit même de l'inverse puisque la collection est créée après insertion du premier document. Cela rend les modifications dites « de structure » plus simples à réaliser.

Le modèle « Base Orientée Colonnes »

Alors que le modèle relationnel stocke les informations sous forme de lignes, ce modèle-ci stocke les informations sous forme de colonnes. Voici quelques tuples que l'on peut retrouver dans une base de modèle relationnel classique :

- 1- Guitare, Flamenco
- 2 - Piano, Classique
- 3 - Flute, Irlandais

Ces tuples seraient agencés de la manière suivante dans une base orientée colonnes :

```
1, 2, 3
Guitare, Piano, Flute
Flamenco, Classique, Irlandais
```

Ce modèle permet donc d'ajouter très facilement des informations et limite les valeurs nulles.

Le modèle « Graphes »

Ce dernier modèle est le moins connu et sans doute le plus complexe des quatre. Il se base sur un modèle mathématique appelé « Théorie des Graphes ».

Dans ce modèle, les données sont modélisées sous forme de « Nœud » dont la structure est sensiblement identique à celle des documents vus précédemment. Ces données sont reliées entre elles par des « Arcs » orientés et nommés. La figure suivante montre un exemple de données stockées dans une base sur le modèle des graphes :

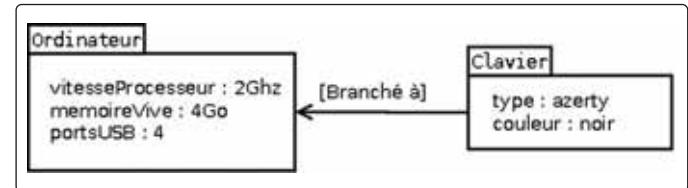


Figure 1 : exemple de stockage avec le modèle Graphes

Après cette présentation des principes de fonctionnement des systèmes NoSQL, nous allons nous intéresser aux différences entre les SGBD NoSQL et les SGBD relationnels.

> Comparaison NoSQL / SGBDR

Pour chaque type de système, une liste de points forts et de points faibles est présentée.

Tout d'abord, intéressons-nous aux SGBD de type relationnel :

Les points forts :

- Beaucoup de fonctionnalités et de règles pour garantir des bases cohérentes et complètes : mécanisme de verrou pour la gestion

- des accès concurrentiels, respect des propriétés ACID (Atomicité, Cohérence, Isolation, Durabilité).
- Utilisation des outils spécialisés comme le modèle entité-relation qui est le Modèle Conceptuel de Données (MCD) utilisé pour modéliser la structure de la base.

Les points faibles :

- Performances qui déclinent avec l'augmentation du volume de données
- Problématique de distribution des données d'une base.

Concernant les SGBD de type NoSQL, les points forts et faibles sont globalement inversés :

Les points forts :

- Bons temps de réponse malgré de très gros volumes de données
- Facilement distribuable
- Plus flexible en cas de panne (disponibilité partielle voire totale)

Les points faibles :

- Moins de propriétés garantissant un état cohérent de la base. Conformément au théorème de Brewer (aussi appelé théorème du CAP), seules deux des trois propriétés suivantes peuvent être respectées par un SGBD NoSQL : Cohérence, Disponibilité et Résistance au morcellement
- Pas de mécanismes de jointures, le côté client doit pallier ce problème.
- Il est rare qu'un SGBD de type NoSQL implémente un mécanisme de verrou pour garantir la cohérence pendant des accès concurrents.

> Zoom sur un SGBD NoSQL : MongoDB

Historique

MongoDB est un SGBD NoSQL open source (licence AGPL V3.0) écrit en C++ qui stocke ses données en BSON (Binary JSON). Son développement a commencé en 2007 et a été initié par la société 10gen.

Ce SGBD a été créé dans le but de fournir un compromis entre les bases de données clé/valeur qui sont très facilement distribuables et les bases relationnelles classiques. Aujourd'hui, 10gen est toujours le principal acteur dans le développement et la maintenance de MongoDB.

Présence sur le marché

En terme de marché, MongoDB est utilisé par des grands comptes comme MTV, Disney, Doodle ou encore le CERN (Organisation Européenne pour la Recherche Nucléaire) pour les données concernant l'accélérateur de particules.

Principes de fonctionnement

MongoDB utilise le modèle de base documentaire pour stocker les informations. Il présente des pilotes pour la plupart des langages communément utilisés aujourd'hui (C, C++, C#, PHP, Java, Python ...). La partie serveur de MongoDB se présente sous la forme d'un démon (ou service sous Windows). Pour le côté client, il existe un mode terminal mais aussi différentes surcouches graphiques (php-MongoAdmin, MongoHub ...).

Le requêtage se fait avec une syntaxe comparable au javascript et MongoDB renvoie les données en JSON.

> Quelques exemples de requêtes

La syntaxe d'une commande est simple et se décompose de la manière suivante :

```
db.nomCollection.nomFonction()
```

Par exemple, pour récupérer les éléments dans la collection « voitures », la ligne suivante est écrite :

```
db.voitures.find()
```

L'exemple de requête suivant permet de récupérer le nom et le prénom des utilisateurs féminins dans une base de type relationnel :

```
SELECT nom, prenom FROM utilisateur WHERE genre = 'F';
```

Un équivalent de cette requête pour le SGBD MongoDB est :

```
db.utilisateur.find({genre : « F »}, {nom : 1, prenom : 1})
```

La seconde liste en paramètre de l'appel de la méthode 'find' permet de définir les attributs qui doivent être affichés ou non.

Le chiffre '0' permet de bloquer l'affichage de l'attribut alors qu'un nombre positif (dans notre exemple '1') permet d'autoriser l'affichage.

La requête suivante récupère les données de la voiture de monsieur « Jacques Dupont » :

```
SELECT V.* FROM utilisateur U, voiture V WHERE U.id = V.id_proprietaire  
AND U.nom = 'Dupont' AND U.prenom = 'Jacques';
```

Un équivalent de cette requête pour le SGBD MongoDB est :

```
utilisateur_temp = db.utilisateur.findOne({nom : « Dupont »,  
prenom : « Jacques »})  
voiture = db.voiture.find({id_proprietaire : utilisateur_temp.  
id})
```

Cette façon d'utiliser les clés étrangères en deux temps est la manière recommandée par le site officiel de MongoDB[2].

> Conclusion

De par ces différences avec les traditionnels modèles relationnel et objet, le modèle NoSQL représente une nouvelle catégorie de SGBD. Il ne s'agit pas de remplacer entièrement les bases relationnelles en bases NoSQL, il s'agit plutôt de fournir une solution alternative là où les modèles en place montrent leurs limites.

Certains projets font d'ailleurs cohabiter les deux types de systèmes au sein d'une même application, permettant ainsi de bénéficier des avantages de chacun.

Sources

[1] Site officiel de MongoDB : <http://www.mongodb.org/>

[2] Page concernant les références sur le site de MongoDB : <http://www.mongodb.org/display/DOCS/Database+References>

Jorrit Wortel

Consultant Osaxis (www.osaxis.fr)



Wakanda : a stack is born !

Fruit de quatre années de R&D, Wakanda, la nouvelle plateforme de l'éditeur français 4D, quitte officiellement son costume Bêta pour endosser ses habits de «Version de production». En se définissant comme la première plateforme de développement «end-to-end» JavaScript pour le développement d'applications Web et mobiles, Wakanda entend jouer son rôle dans la mutation des outils de développement engendrée par l'avènement du Cloud et de la mobilité, tout en se concentrant sur l'écosystème dont son créateur s'est fait la spécialité, celui des applications professionnelles.

La raison d'être de Wakanda part d'un constat simple : la mise en œuvre d'applications Web n'est toujours pas entrée dans sa phase de maturité et s'avère encore chronophage et complexe. La quantité de technologies à interconnecter, de langages à maîtriser, de problèmes de compatibilité et de maintenance à régler, implique d'avoir à sa disposition soit beaucoup de temps, soit beaucoup de ressources. Dès 2008, (voir encadré historique) Wakanda a fait le pari du Server-Side JavaScript et du HTML5, encore balbutiants à l'époque, comme sésame vers un gisement de productivité et de facilité de déploiement sur tout support embarquant un navigateur. La deuxième clé fut l'introduction du Model-Driven Development au cœur de son architecture (voir article pages suivantes). En repensant la création d'applications Web non pas comme un empilement de briques technologiques mais comme un processus de création et d'améliorations perpétuelles effectué par des équipes toujours plus réduites et agiles, Wakanda a l'ambition d'accélérer drastiquement la phase de développement, de simplifier au maximum les problématiques de déploiement, et donc, in fine, de réduire le coût total de chaque projet.

> Eau, gaz et JavaScript à tous les étages

En adoptant un langage familier à des générations de développeurs de tous horizons, Wakanda s'assure d'une courbe d'apprentissage



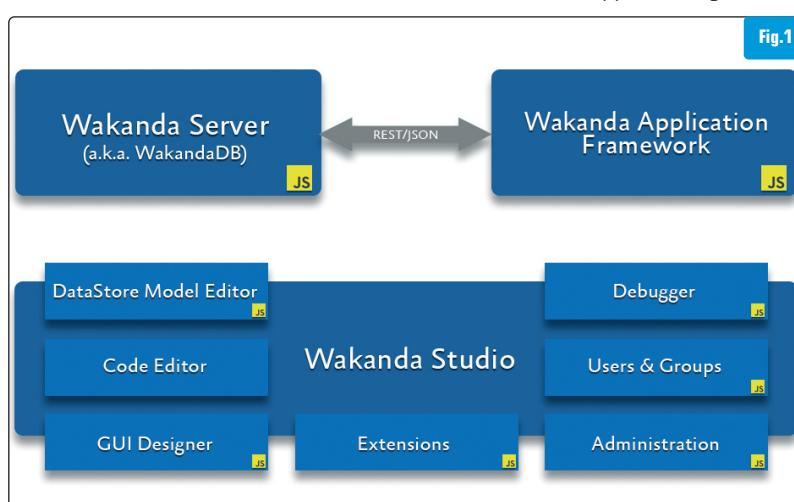
indolore. La plateforme fait partie de la grande famille du Server-Side Javascript, dont l'acteur le plus connu aujourd'hui est node.js. Son originalité c'est que la stack complète, depuis les données jusqu'au framework, repose sur JavaScript. En modélisant son application, le développeur Wakanda va, de façon transparente, instancier des objets JavaScript immédiatement utilisables, tant côté serveur, dans l'application de ses règles métiers et l'exécution de ses méthodes, attributs calculés, et gestionnaire d'événements, que côté client, grâce à la mise à disposition du modèle par le framework Ajax, via son data provider et ses datasources.

> Le tour du non-propriétaire

Wakanda est une plateforme Open Source qui propose une *dual license*, avec des capacités de développement strictement identiques dans les deux cas. La création d'une communauté d'utilisateurs et de contributeurs est clairement l'ambition première affichée

par l'éditeur de Wakanda, dans le but de garantir la pérennité des projets et de s'adapter au foisonnement d'idées, de frameworks, d'appareils et d'usages qui se déroule sous nos yeux [Fig.1].

Au cœur de la plateforme se niche WakandaDB[1], un datastore objet NoSQL ultra-performant, qui, joint au moteur d'exécution JavaScript basé sur Webkit, et à un serveur HTTP communiquant via JSON/REST constitue Wakanda Server, le back-end multi-plateformes (Linux, Mac, Windows) qui permettra d'héberger une solution Wakanda sur un serveur dédié, sur un Cloud privé ou public, en mode SaaS, etc.. Le framework Wakanda est automatiquement déployé vers les clients HTML et embarque un Data Provider qui se comporte comme un proxy des différentes Data-Classes générées côté serveur. Ainsi de façon transparente, tous les objets JavaScript générés à l'aide du modèle de don-





nées sont exploitables via les différents widgets fournis avec le framework, mais également au travers de requêtes REST pour potentiellement connecter tout type d'interface, par ligne de commande ou à l'aide d'adaptateurs que pourront fournir des tierces parties, la communauté, ou l'éditeur 4D lui-même.

Enfin, le studio Wakanda vient compléter la suite de développement en proposant un IDE dédié capable de contrôler, via une application Desktop Mac ou Windows, aussi bien le modèle, le code SSJS et les droits utilisateurs côté serveur que l'interface, les widgets, et l'adaptation cross-device de chaque écran côté client. Le studio fait la part belle à la gestion graphique du projet, et fait en sorte d'économiser à chaque étape l'écriture de code «technique». Le développeur peut dès lors se concentrer sur ses propres scripts et méthodes qui contiendront l'intelligence métier qui fera la valeur de son application. Outre les différents éditeurs (modèles, code, UI, utilisateurs et groupes) [Fig.2 et 3] le studio propose une fenêtre d'administration des solutions et projets et un débuggeur qui permet de tracer le code JavaScript tournant sur le serveur.

L'installation, tout comme le déploiement, s'effectuent par *drag-and-drop* d'un package sans aucune configuration. L'hébergement d'une solution Wakanda peut s'effectuer sur un serveur dédié, sur un VPS (serveur dédié virtuel)[2], ou même sur une plateforme IaaS Cloud comme Amazon EC2 ou Rackspace. Côté mobile, Wakanda permet de réaliser des applications Web ou Hybrides, c'est-à-dire embarquées dans un runtime natif généré par PhoneGap [Fig.4]. Il est également possible de créer des applications mobiles natives et d'accéder au Serveur via ses API REST et JSON-RPC.

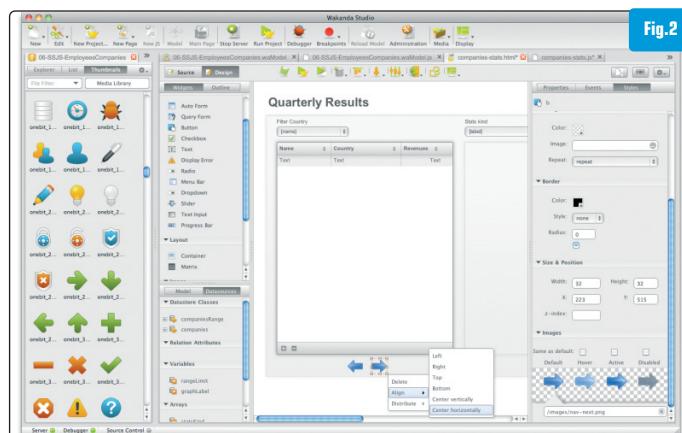


Fig.2

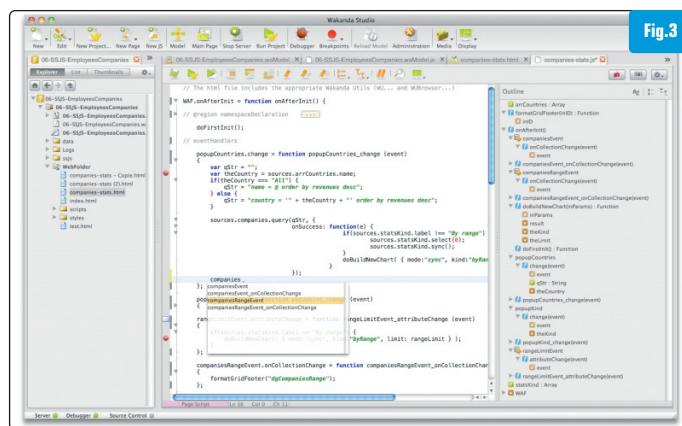


Fig.3

- [1] Plus de détail sur <http://wakandaDB.org>
 [2] <http://techwirenews.com/wakanda-server-on-gandi-vps/>

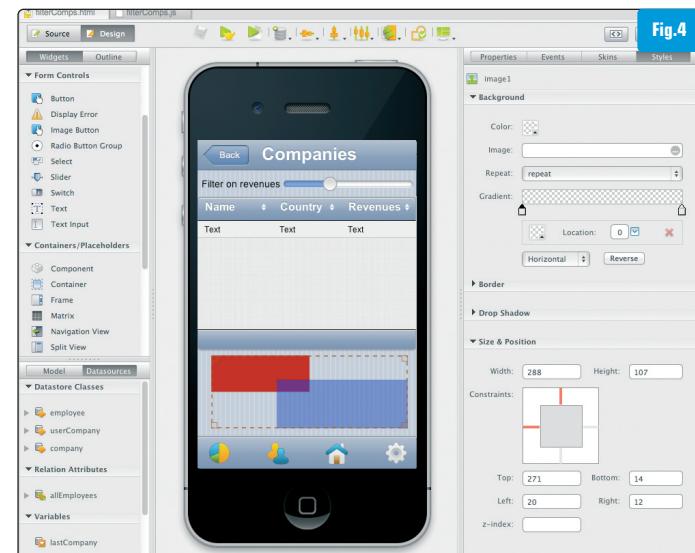
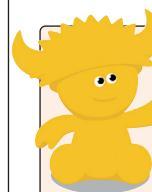


Fig.4

> On the roadmap again

Wakanda est téléchargeable sur www.wakanda.org via 3 canaux distincts : production, stabilisation et développement, les deux derniers offrant l'accès à des *nightly* ou *weekly builds*. Un cycle court de sortie a été mis en place, afin de répondre au plus vite aux expectatives du marché. Dans les prochains mois, l'accent sera mis sur l'optimisation des performances sur les mobiles, des déploiements sur le Cloud, ainsi que sur l'interopérabilité avec d'autres plateformes, bases de données et frameworks populaires. Les retours et attentes de la communauté contribuent largement à son amélioration via un forum très actif comportant déjà plusieurs centaines d'utilisateurs.

Jean-Michel Biraghi
Directeur Marketing Corporate, 4D



Un peu d'histoire

4D possède plus de 25 ans d'expérience dans le domaine des applications métiers et des bases de données, pour lesquelles l'intégration de fonctionnalités Web puissantes a toujours existé. Cependant Laurent Ribardiére, le créateur de la plateforme, souhaitait réaliser une nouvelle gamme d'outils nativement conçue pour les Web Apps. Un choix structurant fut celui du langage à utiliser sur le serveur. Il fallait un langage standard et qui, idéalement, puisse fonctionner aussi dans le navigateur pour éviter tout doublon d'implémentation de la logique métier. JavaScript a été préféré à d'autres options telles que Java, .NET/ActiveX ou ActionScript pour son côté standard et universel. Puis vint la sélection du moteur d'exécution. Le choix s'est porté sur celui du projet Webkit, JavaScriptCore (avec son *Just-in-Time Compiler*), pour des raisons de performances, de support du 64 bits et pour ses capacités *multi-thread* avancées (*thread-safe*, pré-emptif) particulièrement importantes dans le cadre d'une implémentation de bases de données avec des API synchrones facilement maintenables. Le studio Wakanda s'est greffé par la suite au projet et contient lui même une grande proportion de code JavaScript, en plus d'autres parties en C++.



Concevoir des applications Web Model-Driven avec Wakanda

La modélisation a beau avoir le vent en poupe, l'appliquer dans son travail quotidien relève souvent du grand écart pour la plupart des développeurs Web. En effet, si beaucoup s'accordent sur l'intérêt du MDD (Model-Driven Development) : transcription optimale du cahier des charges, structure logique saine sur laquelle se construiront des édifices plus stables, sa mise en pratique est généralement complexe voire malaisée, et même après une implémentation propre, la maintenance du modèle est un deuxième défi encore plus dur à surmonter.

Or, il se trouve que cet aspect du développement est justement la pierre angulaire de la plateforme Wakanda, qui a été conçue autour du postulat suivant : en réalisant une application en ligne directe avec la logique métier, quitte à changer certaines habitudes de développement, on réduit pratiquement à zéro les risques de s'égarer par la suite. Cette approche originale qui ambitionne d'appliquer de façon efficace le MDD au développement Web passe par la création d'une base de données totalement originale et spécialement « entraînée » à cette course d'obstacles et de vitesse.

> Etat des lieux express du MDD

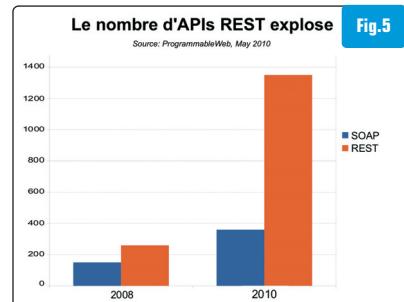
Il serait prétentieux de vouloir résumer l'abondante littérature sur la modélisation en un paragraphe. Pour le propos qui nous intéresse ici, rappelons simplement que les modèles ont vocation à présenter les besoins sous forme de services. La description précise des composants consiste à calquer un diagramme UML sur la couche logique, ce qui rend indispensable la génération de code ayant recours à un ORM pour faire coïncider cette description avec, d'une part un schéma relationnel SQL, et de l'autre les objets du langage de développement serveur. De plus, du code supplémentaire est nécessaire pour exposer le modèle aux clients via des services. Certains écrivent la totalité de ce code manuellement en suivant des Design patterns, d'autres utilisent des outils qui le génèrent. Dans les deux cas, chaque intervention sur le modèle implique des modifications dans plusieurs couches de l'application. Cette organisation sous forme de services, dénommée SOA (*Service Oriented Architecture*), est généralement centrale pour le MDD et a dominé les débats sur le développement dans la décennie écoulée. Pour être utilisable sur le Web, une série de standards ont été créés et regroupés sous le nom « WS-* » (pour Web Services). Les API issues de cette famille utilisent un protocole appelé SOAP (*Simple Object Access Protocol*).

> Les contraintes de l'architecture Web

Le Web est un réseau très large dont les maillons peuvent être temporairement ou définitivement coupés, dont les connexions peuvent s'avérer plus ou moins lentes, et qui fut à l'origine conçu pour diffuser des contenus indépendants auprès d'un très grand nombre d'utilisateurs. Le HTTP de Tim Berners Lee, a été amélioré par Roy

Thomas Fielding grâce à un style d'architecture décrit dans sa thèse de doctorat : REST[3]. REST représente le Web comme une *Resource Oriented Architecture* (ROA) et celle-ci fonctionne très différemment du SOA. Une API Web dite REST est le plus souvent une extension du HTTP (ex : WebDAV, GData, OData...) qui profite de fonctionnalités natives comme le support du cache, des proxies, et de comportements adaptés à des commandes standard (GET, PUT, DELETE...). Les règles de REST permettent une scalabilité et une évolutivité des services optimales. SOAP n'utilisant qu'un sous-ensemble limité du HTTP, une couche complexe et verbeuse vient se rajouter au détriment de ses performances, entre autres.

Rapidement, les applications tournant sur le Web ont commencé à privilégier les API REST par rapport à SOAP. En 2003, Amazon, qui fournissait les 2 types d'API, déclarait que REST représentait plus de 85% de l'utilisation[4]. En 2009, Google prit même la décision de supprimer sa SOAP Search API[5]. Ce graphique extrait de la présentation de John Messer sur les API en 2010 parle de lui-même [6] : [Fig.5].



> Wakanda : le meilleur des deux mondes

Wakanda, avec sa base objet NoSQL, jette un pont entre les deux rives du MDD et du REST. Sa base de données, son data store pour être précis, utilise un schéma UML-like qui ne requiert pas de bêquille ORM pour être exploitable. Aucune génération de code intermédiaire n'est en effet induite après la modélisation. Ce n'est pas donc en SQL, mais directement via une API Server-Side JavaScript que l'on accède aux objets du modèle. Grâce à son serveur HTTP intégré, on accède nativement au data store par l'API REST sans nécessité d'exécuter le moindre code SQL ou JavaScript, ni manuel ni généré automatiquement. En résumé, créer une application Wakanda consiste à modéliser des objets et des règles métiers d'une part, puis à les exploiter instantanément via des commandes REST ou JavaScript d'autre part. Aucune étape intermédiaire n'est nécessaire [Fig.6].

> Le respect ne se perd plus

Les bénéfices de cette approche sont immédiats et puissants. Tout d'abord, en définissant les règles de l'application au plus bas niveau dans le modèle de la base de données, pour tout dire au cœur même des objets, on s'assure d'une mise en œuvre respectueuse de la

[3] <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>

[4] <http://www.oreillynet.com/pub/wlg/3005>

[5] <http://googlecode.blogspot.fr/2009/08/well-earned-retirement-for-soap-search.html>

[6] <http://www.slideshare.net/jmussner/open-apis-state-of-the-market-2011>



logique métier. Aucune distorsion ne risque de parasiter l'une de ces règles lors de la réécriture dans un autre langage, puisqu'elle est naturellement appliquée par l'API REST. Par exemple, les valeurs minimum et maximum, les patterns de formatage, les règles de validation d'un objet ne pourront être ni contournées, ni oubliées, ce qui procure au développeur et au chef de produit une tranquillité d'esprit inédite.

> Tu ne te répéteras point

Contrairement aux idées reçues, un modèle n'a pas simplement vocation à être dessiné, il doit être maintenu presque quotidiennement. En effet, pour faire un rapprochement parlant, on peut dire que l'ensemble des règles métiers qui régissent une application obéit à la seconde loi de la thermodynamique et est par nature destiné à évoluer parfois imperceptiblement mais irréversiblement. Cette notion d'entropie est particulièrement adaptée : lorsque qu'une règle métier évolue, appliquer le changement au niveau du front-end est, à terme, générateur de désordre. Dans Wakanda, en modifiant la règle au niveau du modèle, on se prémunit contre tout effet de bord ; le changement est disponible partout : dans la base de données (attributs calculés, méthodes), dans le code serveur (modules, RPC, services), dans l'API REST et donc sur le client. C'est ici qu'intervient le changement d'habitude nécessaire pour un bon développement : en se forçant à revenir au modèle le plus souvent possible pour y enrichir les règles métiers, le développeur Wakanda gagne en productivité et en cohérence, et respecte le principe du DRY (*Don't Repeat Yourself*), ce qui représente un bénéfice énorme pour un «coût» raisonnable. Cerise sur la gâteau, ces bonnes pratiques ne s'opèrent pas dans une tour d'ivoire. Le support d'API W3C/HTML5 standards, des modules CommonJS, et même de certaines API node.js permettent la réutilisation de librairies JavaScript existantes, la création de modules cross-platform, ou encore de scripts compatibles client et serveur (très pratique en mode offline).

> Sécurité Web améliorée

Tout comme les règles métiers, les droits sur les attributs et les méthodes sont directement définis dans WakandaDB, avec la possibilité d'étendre une classe pour adapter les droits et le scope à un groupe d'utilisateurs spécifiques. Une autre plateforme utiliserait typiquement une connexion unique pour accéder à la base de données, donnant en interne les mêmes droits sur les données à tous les utilisateurs Web, et rendant de ce fait les injections SQL extrêmement dangereuses. Dans Wakanda les droits d'accès aux données sont définis pour les groupes d'utilisateurs Web authentifiés finaux. L'accès à chaque objet stocké peut même être facilement

limité, par design, à un utilisateur en particulier. Dans les cas les plus spécifiques, des gestionnaires d'accès, de validation, ou de suppression seront définis pour rendre l'application définitivement inattaquable [Fig.7].

> Et la programmation devint intuitive

En éliminant toutes les couches d'abstraction technique, Wakanda aboutit à une grande pureté dans l'expression du code. L'API SSJS est ainsi extrêmement simple à manipuler et il devient naturel d'aligner ses lignes de code dans l'ordre exact dans lequel on les pense pour exprimer la contrainte métier. Par exemple, il ne devrait pas être nécessaire de commenter cette double requête N vers M pour en comprendre la finalité, à comparer avec son équivalent (probablement sur plus de quatre lignes) dans d'autres langages :

```
var john = ds.Person.find("firstName eq John");
var conferences = john.conferences;
var JohnJSConferences = conferences.filter("title eq '*JavaScript*'");
var JSAttendeesJohnMet = JohnJSConferences.attendees;
```

> Cours Forrest, Cours !

Le dernier bénéfice, et non des moindres, concerne la performance. Wakanda s'inspire des meilleures pratiques du développement Web, dans le respect des principes de REST, et utilise des mécanismes très efficaces tels que le *Lazy Loading*, le cache d'*EntitySet* ou l'*Optimistic Locking*. Les fichiers HTML sont généralement tous statiques, et donc cachables au même titre que le framework Ajax, tandis que les données dynamiques sont toutes récupérées en pur JSON via l'API REST. L'API du serveur étant native, le client Web a, en toute sécurité et en respectant toutes les règles métiers, un accès immédiat aux données sans devoir passer par un code généré et/ou un ORM envoyant à son tour une ou plusieurs requêtes SQL potentiellement coûteuses.

> Conclusion

En repensant le développement Web dans sa globalité et ses aspects pratiques, Wakanda est conçue pour procurer un *time-to-market* le plus réduit possible, sans faire de compromis ni sur la qualité, ni la sécurité. Quant à la maintenabilité, son modèle évolutif est conçu pour faciliter l'intégration de nouvelles fonctionnalités de manière itérative sans jamais avoir à redémarrer le serveur de production.

Alexandre Morgaut - Architecte Web
Wakanda Community Manager



Premiers pas avec Wakanda

Dans ce numéro, nous nous concentrerons sur le serveur de Wakanda et en particulier sur WakandaDB. Nous allons découvrir comment créer notre premier DataModel Objet avec l'aide du studio, observer l'API REST exposée, et finalement interagir avec celle-ci via le DataBrowser intégré.

> Installation et lancement de Wakanda

Installer Wakanda est très simple. Rendez-vous sur la page de téléchargement [1] et choisissez le package « All-in-one » correspondant à votre plateforme (Mac, Windows 32 ou 64 bits) [2], puis décompressez-le dans le dossier de votre choix. Le plus simple au départ est de garder Wakanda Server et Wakanda Studio dans le même dossier, le studio trouvera et lancera ainsi le serveur de lui-même. Vous n'avez plus qu'à lancer l'application Wakanda Studio. La documentation en ligne décrit en détail son interface [3].

> Création d'un premier projet

Dans Wakanda, le serveur fait tourner une solution, celle-ci pouvant inclure plusieurs projets. Pour aller plus vite, depuis la barre d'outils principale, demandons à créer un nouveau projet en cliquant sur le bouton  , choisissons le modèle « blank » et appelons-le par exemple « Todo », la solution sera alors créée automatiquement et portera le même nom que le projet. Ouvrons le fichier « Todo.waSettings » de la solution, fixons l'authentification à « custom », et sauvons.

> Création d'un utilisateur

Une solution Wakanda comprend un annuaire d'utilisateurs et de groupes qui est partagé pour tous ses projets. Bien sûr, vous pourrez au besoin limiter à un groupe spécifique l'accès aux données d'un projet particulier.

Depuis la barre d'outils, cliquons sur le bouton  . Créons l'utilisateur « admin » associé au groupe « Admin » pré-existant avec un mot de passe privé, puis l'utilisateur « john », avec pour fullName « John Doe » et pour mot de passe de test « pass ». Depuis l'onglet « Groups » créons le groupe « User » et ajoutons-y John [Aide (4)].

> Création d'un modèle

Nous allons maintenant créer un petit modèle pour découvrir quelques particularités du moteur NoSQL WakandaDB. Cliquons sur le bouton  . Une première DataClass est proposée par défaut, nous allons la renommer « Todo ». Dans la liste de ses propriétés vous constaterez qu'un « Collection name » lui est également attribué. Renommez-le « TodoList », ce sera un petit peu plus parlant.

Attributs de stockage

Toute DataClass a par défaut un attribut « ID » de type « long » défini comme clé primaire. Il est recommandé de le garder. Vous pourrez éventuellement le renommer ou changer son type. Ajoutons les attributs « label », « userName » et « userFullName » de type « string », ceux-ci seront considérés comme attributs de stockage [Fig.8].

Attributs calculés

Nous allons transformer « userFullName » en attribut calculé. Dans ses propriétés, fixons son « script kind » à « JavaScript » et ajoutons un « getter » en cliquant sur **On Get**  . Saisissons :

```
var user = directory.user(this.userName);
return (user === null) ? "" : user.fullName;
```

et revenons sur le Model Designer.

Attributs de navigation

Dans Wakanda, toute DataClass créée est immédiatement utilisable comme type d'attribut. Un attribut utilisant un nom de DataClass, ou son nom de collection, est désigné en interne comme attribut de relation car il référence un objet défini par une autre DataClass. A l'utilisation, on parlera également d'attribut de navigation.

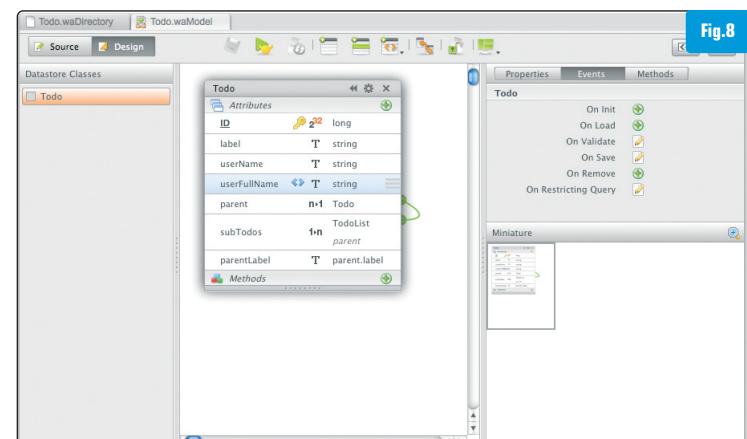
Nous allons établir qu'une tâche peut avoir des sous-tâches. Ajoutons donc un attribut « subTodos » de type « TodoList » (le nom de collection donné à la DataClass), Wakanda propose automatiquement de créer un attribut « todo » (reprenant le nom de la DataClass) utilisant la référence inverse. Appelons-le plutôt « parent ».

Aliases

Créons enfin un attribut « parentLabel » et donnons-lui pour type « parent.label ». Vous constaterez ce faisant que l'auto-complétion a d'abord proposé « parent », qui est un attribut de navigation, puis « label » qui est un autre attribut de la DataClass. Vous venez en réalité de créer un attribut de type « Alias ». Ceux-ci permettent de donner un accès direct à des sous-données liées.

Événements

Sélectionnons la DataClass en cliquant sur son titre, puis sélectionnons l'onglet « Events », nous allons faire en sorte que chaque « User » n'ait accès qu'à ses propres « Todos » en ajoutant le script suivant :



[1] <http://www.wakanda.org/downloads>

[2] Le studio n'est pas disponible sous Linux

[3] <http://doc.wakanda.org/Wakanda-Studio/Solution-Manager/>

[4] <http://doc.wakanda.org/Wakanda-Studio/Directory-Manager/>



On Restricting Query

```
var session = currentSession();
if (session.belongsTo('Admin')) {
    return ds.Todo.all(); //tous les objets présents
} else if (session.belongsTo('User')) {
    return ds.Todo.query('userName eq :1', session.user.name);
} else {
    return ds.Todo.createEntityCollection(); //collection vide
}
```

Il permet aux administrateurs d'accéder à tous les éléments tandis que les utilisateurs n'ont accès qu'à leurs propres « Todos ». Si l'utilisateur n'est pas connecté, rien n'est accessible.

On validate

```
var user = currentSession().user;
if (user.name === 'default guest') {
    return {error: -1, errorMessage: 'Connectez-Vous!!!'};
}
```

Le but est cette fois d'empêcher une personne non connectée d'ajouter un élément.

On Save

```
this.userName = currentUser().name;
```

Tout « Todo » créé par un utilisateur lui sera maintenant attribué de façon certaine. Sauvons ce Modèle et son fichier JavaScript associé.

> Découverte de l'API REST

Astuce : Installez l'extension « JSON View » sur votre navigateur.

Démarrons Wakanda Server en cliquant sur le bouton  de la barre d'outils. Dans votre navigateur aller à l'adresse : [http://127.0.0.1:8081/rest/\\$catalog](http://127.0.0.1:8081/rest/$catalog). On y trouve, en JSON, la liste des DataClasses du Modèle, soit ici, « Todo ». La propriété « dataURI » donne accès aux objets correspondant à la DataClass présents dans le dataStore, et « uri » à la description de cette DataClass. A l'adresse [http://127.0.0.1:8081/rest/\\$catalog/Todo](http://127.0.0.1:8081/rest/$catalog/Todo), nous retrouvons donc tous les attributs que nous avons créés et l'information « kind » doit maintenant vous être familière.

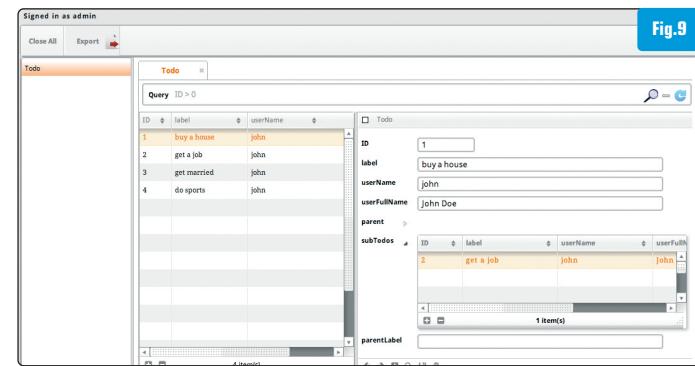
> Création de données via SSJS

Ecrivons un petit script pour générer des données.

Cliquons sur  et appelons le fichier « generateData.js ».

```
loginByPassword('john', 'pass');
var todoArray = ['buy a house', 'get married', 'do sports'];
todoArray.forEach(function importTodo(todoLabel) {
    var newTodo = new ds.Todo({label: todoLabel});
    newTodo.save();
    if (todoLabel === 'buy a house') {
        var getAJob = new ds.Todo({label: 'get a job'});
        getAJob.parent = newTodo;
        getAJob.save();
    }
});
```

« ds » représente le DataStore Wakanda. On y retrouve les DataClasses définies dans le Modèle pour créer ou retrouver des objets. Tout objet créé depuis une DataClass Wakanda hérite d'une méthode



« save() » permettant de stocker l'objet créé dans le DataStore. Cliquez sur le bouton  de l'éditeur de code pour exécuter le script.

Astuce : Poser un point d'arrêt pour suivre l'exécution depuis le débogueur.

> Interagir avec les données

Wakanda Studio fournit un utilitaire pratique: le DataBrowser [Fig.9].

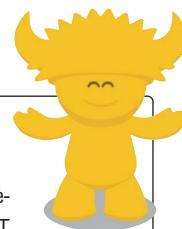
Cliquez sur  pour le lancer.

Tant que vous n'êtes pas connecté, vous ne verrez aucun élément de type « Todo ». Connectons-nous en tant qu'« admin » et cliquons sur  pour rafraîchir la liste des todos. Vous voyez les éléments importés, vous pouvez les sélectionner pour voir leurs détails. Ajoutons des éléments en cliquant sur le bouton . Le userName est automatiquement fixé à « admin ». Reconnectons-nous en tant que « john ». Après un refresh, nous nous rendons compte que nous n'avons plus accès qu'aux éléments de John. Ceux de « admin » sont inaccessibles. Dorénavant, chaque utilisateur ajouté au groupe « User » pourra gérer sa TodoList personnelle sans conflit avec celle des autres utilisateurs.

> Conclusion

WakandaDB est un serveur NoSQL objet puissant grâce auquel on peut décrire toute la logique métier au sein du modèle. Une fois les DataClasses définies, toutes les données liées aux objets sont immédiatement accessibles sans devoir passer par des jointures ou des map-reduce complexes, sans devoir maintenir de code généré supplémentaire. Seul le code lié aux règles et aux besoins métiers est requis, et Wakanda permet de le rendre plus simple à écrire, à comprendre, à maintenir, et à faire évoluer.

Alexandre Morgaut



Dans le prochain numéro...

Nous venons de créer un modèle d'application exécutable et directement accessible via une API REST que vous pouvez maintenant utiliser depuis n'importe quel framework Ajax (Sencha, backbone.js, Kendo UI, ...), serveur (node.js, PHP, Java, ...), ou natif (Objective C, .NET).

Dans le prochain numéro, vous verrez comment le studio de Wakanda vous permet également de profiter d'un framework dédié offrant un accès direct à votre modèle depuis le client.

Vous découvrirez comment créer des interfaces professionnelles, Desktop ou Mobile, avec des widgets riches utilisant la puissance de HTML5 et CSS3.

Fidèle à l'esprit de Wakanda, votre code métier restera, comme toujours, facilement maintenable et évolutif !



Istockphoto© Henrik Jonsson

La maison connectée

La Geek Attitude !



Depuis des années, on nous promet la domotique chez soi. Vous savez, cette technologie « magique » censée simplifier notre vie au quotidien ! Cette belle promesse n'a jamais été tenue : une offre très limitée, trop complexe, trop chère, une utilisation restreinte.

Tout cela est bel et bien derrière nous ! Les nouveaux terminaux mobiles qui servent de plus en plus de télécommandes universelles, des objets divers et variés de plus en plus connectés (machine to machine), des protocoles domotiques reconnus et largement disponibles, font de la maison connectée une réalité.

Oui, c'est une réalité. Nous pouvons installer rapidement des lampes, des prises, des commandes diverses et variées pour « programmer » certains équipements de la maison grâce à des protocoles (ex. : X10, Z-Wave) et des API. Dans notre dossier spécial été, nous allons vous plonger dans l'univers du connecté : où acheter, comment mettre en place une

première domotique, créer les premières applications...

Les premiers kits domotiques X10 / Z-Wave coûtent 150 – 200 euros, mais l'addition s'alourdit si vous poussez l'expérience. L'avantage est de pouvoir modular votre projet domotique, selon vos envies et votre budget. Attention tout de même à l'effet WAF et MAF (eh oui car nous ne sommes pas tous des geeks sans limites). SFR viens d'annoncer une offre grand public, «home by SFR...».

Et ce n'est qu'un début !

François Tonic

Piloter sa maison ? C'est facile ! La preuve ...

1^{re} partie : les technologies et le réseau domestique

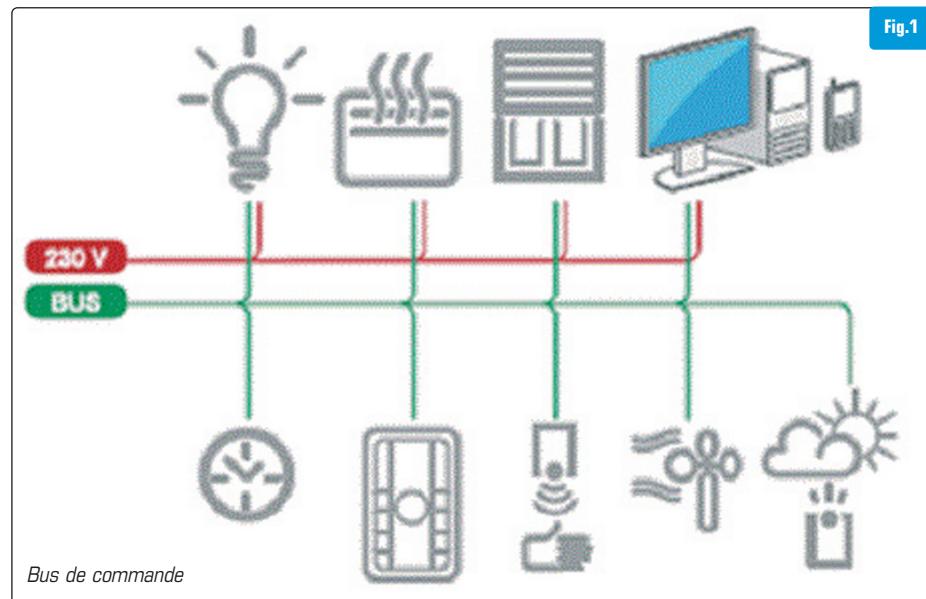
La domotique, du latin « domus » (maison) est l'ensemble des techniques et technologies qui permet de superviser et d'automatiser les services de l'habitat comme l'éclairage, le chauffage, la sécurité mais aussi la consommation de l'habitat, ainsi que les différents services de détente et de loisirs comme la diffusion sonore et vidéo, la téléphonie et l'informatique au sens large.

Les possibilités permettent aujourd'hui de concevoir des habitats confortables, évolutifs, sûrs, autonomes et économies. Mariée à des technologies d'agents intelligents, comme la plateforme IAS d'xBrainSoft, la domotique pose les fondations de la maison intelligente et communicante.

Imaginez une maison à qui on parle et qui nous parle, qui enregistre votre émission favorite quand elle sait que vous êtes absent, qui rallume le chauffage quand vous rentrez le soir d'hiver, qui vous accueille avec votre playlist musicale du moment, toutes lumières allumées, ou tout simplement vous prévient de prendre votre parapluie lorsque vous sortez et que la météo annonce un orage en prenant le soin d'éteindre les lumières, chauffage, volets, musique et TV à votre place.

Pour cela on peut distinguer trois couches permettant de concevoir une maison intelligente : la 1^{re} est la couche « domotique » qui regroupe toutes les techniques et technologies pour piloter votre maison, comme l'éclairage, le chauffage, volets, prises électriques mais aussi les différents capteurs pour la sécurité (déTECTeur de mouvement, d'ouverture, inondation, etc.), la météo et la consommation (électrique, eau, gaz, ...). La 2^e couche est la « domotique multimédia », qui rassemble toutes les techniques et technologies pour diffuser et piloter l'audio, la vidéo, la téléphonie et l'informatique dans la maison. Enfin la dernière couche est « l'intelligence », où il devient possible, grâce à un agent intelligent par exemple, d'humaniser la maison, de la rendre communicante, autonome et intelligente !

Dans cet article nous nous focaliserons uniquement sur les fondations de la maison intelligente, à savoir la couche « domotique » en découvrant les différentes technologies présentes sur le marché et de quelle manière nous pouvons développer avec !



LES TECHNOLOGIES DOMOTIQUES

Les différentes technologies domotiques peuvent être regroupées sous trois catégories en fonction de leur mode de communication :

- Les technologies « filaires »
- Les technologies « sans nouveau fil » (courant porteur)
- Les technologies « sans fil »

Nous pourrions aussi évoquer les technologies infrarouges, comme le SIRCS ou le RC5, mais ces technologies sont très limitées et s'adressent plutôt à l'automatisation des équipements électroniques audiovisuels.

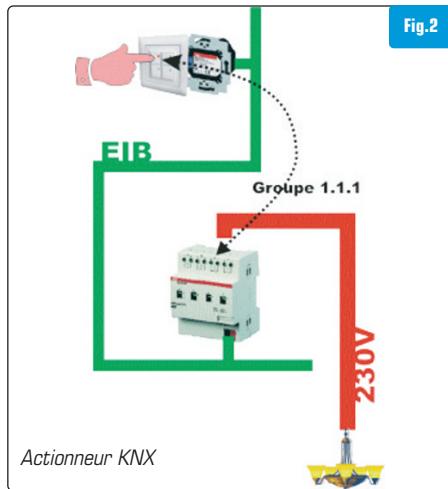
> Les technologies filaires

Dans une maison « classique », l'électricité est relativement simple. Elle est distribuée depuis le tableau électrique qui assure la protection des différents circuits et alimente l'ensemble des équipements au moyen de boîtes de dérivation. Les éclairages sont généralement contrôlés par des interrupteurs, simples dispositifs mécaniques qui

« coupent » la phase de la prise/lampe contrôlée. Cette approche ne permet pas de contrôler les équipements et diffuse le 230V dans toute la maison. De plus elle rend difficile l'évolution dans le temps (les câbles étant ancrés dans les murs, le lien entre le contrôleur et l'action peut être difficilement changé !).

Pour résoudre cela, des technologies par « bus de commande » sont arrivées il y a plus de quinze ans, essentiellement pour les bâtiments industriels et immeubles de bureau. Elles s'implantent aujourd'hui de plus en plus dans les habitats haut de gamme et commencent à apparaître dans les appartements de standing (dernièrement avec le « Smart Building » construit par Bouygues Telecom et Hager à Aubervilliers). L'idée est de séparer le réseau électrique du bus de commande sur lequel seront connectés les différents éléments de contrôle (interrupteur, thermostat, télécommande, écran tactile, etc.) sur des tensions à faible voltage. Grâce à cela, il nous est possible de contrôler l'ensemble des équipements de la maison

depuis n'importe quel type d'interface et nous offre également la possibilité d'associer n'importe quelle commande à une action individuelle, groupée ou générale, et cela à n'importe quel moment (sans modifier le câblage) [Fig.1].



Les technologies les plus répandues étant le KNX et le Lonworks (aux États-Unis). Le KNX est devenu en 2006 un standard normalisé au niveau international et a été choisi par 90% des constructeurs d'équipements électriques en Europe comme Hager, Schneider Electric ou encore Siemens.

En KNX, le bus de commande est un simple câble de type « paire torsadée » qui parcourt la maison et sur lequel sont connectés en série les éléments de contrôle comme les interrupteurs. L'ensemble des circuits d'éclairage, des volets et prises commandées sont ramenés individuellement au tableau électrique et connectés à des actionneurs KNX (module Rail DIN à installer dans le tableau électrique). Ces actionneurs KNX seront ensuite programmés pour répondre aux ordres émis par les interrupteurs/boutons-poussoirs connectés sur le bus de commande ou par des télécommandes sans fil (avec l'utilisation d'une passerelle radio).

Cette architecture évite la prolifération du 230V dans la maison en scindant le réseau électrique du réseau de commande et permet d'adapter facilement le contrôle des différents circuits en fonction de l'utilisation réelle du logement ou en cas de modification interne. Par exemple, en changeant le lit de place dans une chambre, il est possible de reprogrammer individuellement les interrupteurs et les télécommandes afin de les réaffecter aux modules d'éclairage, de volet, ... souhaités. Le protocole KNX supporte aussi le mode de communication sans fil, par courant porteur ou par Ethernet avec l'utilisation

de passerelles. Il devient dès lors possible de compléter son installation par des interrupteurs sans fil par exemple. Il est à noter aussi que dans ce genre d'installation, l'intelligence n'est pas centralisée sur un seul équipement : chaque contrôleur connecté sur le bus (comme les interrupteurs) contient son propre microprogramme des ordres à envoyer sur le bus. Le seul « point of failure » du système réside au niveau de l'alimentation du bus de commande, mais il existe bien évidemment des systèmes auxiliaires de secours permettant le remplacement de l'alimentation principale [Fig.2].

Bien qu'à terme ce type d'installation risque de supplanter peu à peu les installations traditionnelles, il est encore aujourd'hui assez coûteux à mettre en place et très difficile à déployer sur un habitat existant. En effet, ce type d'infrastructure demande à câbler individuellement chaque circuit pour le ramener directement sur le tableau électrique (principal ou secondaire) afin de le connecter sur des actionneurs KNX. Cette condition semble très difficile à mettre en place dans des habitats existants, et n'est souvent possible que pour des projets de construction. Le câblage en étoile du réseau de tension et le câblage du bus de commande en série entraînent une dépense importante (on compte environ 1,2Km de câble électrique pour une habitation de 100/150m²) et demandent des ajustements au niveau des plans de construction pour tenir compte de ce câblage important. De plus, le prix de ce genre d'installation est relativement coûteux : environ 200€ pour l'alimentation du bus, entre 100 et 500 € pour les actionneurs gérant de 1 à 8 contacts, environ 600 euros pour une passerelle TCP/IP, de 70 à 250 € pour les interrupteurs. Il faut compter entre 5 000 et 20 000 € pour une installation « classique » (on estime en général une installation KNX à 10% du budget global). À l'image des réseaux filaire Ethernet, la



Modules KNX

Carte "Web relay" 8 sorties

Fig.4



domotique filaire comme le KNX reste un gage de qualité et de fiabilité. Dans un projet de construction et si le budget le permet, la technologie KNX reste le meilleur choix, par contre pour un habitat existant, en maison comme en appartement, cette technologie reste très difficile à utiliser ! [Fig.3].

Enfin, on pourrait aussi citer les technologies 1-Wire et « web-relay ». La première conçue par Dallas Semiconductor permet de connecter des composants en série, parallèles ou en étoile sur deux fils, souvent utilisée pour des réseaux de capteurs. La seconde, les « cartes web relay » qui fonctionnent par le réseau Ethernet. C'est le cas de l'IPX800 de GCE Electronics (environ 180 euros) qui permet de contrôler 8 sorties sur un module Rail DIN embarquant son propre serveur web pour le contrôle à distance [Fig.4].

> Les technologies « sans nouveaux fils »

Quand il n'est pas possible de passer de nouveaux câbles, une solution consiste à utiliser ceux déjà existants, c'est-à-dire le câblage du réseau de tension. On appelle cela « technologie par courant porteur » où l'on utilise le câblage électrique pour communiquer des informations à une fréquence et tension différentes. Aussi connu sous l'acronyme CPL (courant porteur en ligne), on l'utilise généralement dans les foyers pour mettre en réseau des équipements informatiques facilement sans configuration Wifi ni passage de nouveaux câbles RJ45 : d'où le surnom « sans nouveaux fils » !

Ce genre de technologie demande une bonne qualité de l'installation électrique existante, ce qui dans des vieux bâtiments est rarement le cas. La prolifération de boîtes de dérivation dans le foyer tend à affaiblir le signal. De plus, de nombreux appareils électroniques peuvent créer d'importantes perturbations sur le réseau électrique, ce qui « brouille » la communication entre les équipements. En tête : la machine à laver, le sèche-cheveux, le micro ondes ou encore



certaines alimentations des téléphones et ordinateurs. Il faut avant tout auditer son réseau électrique pour déterminer sa qualité, ajouter des filtres sur les équipements qui pourraient perturber le signal et supprimer les sources parasites, bien que certaines technologies par courant porteur embarquent directement des filtres anti bruit. Parmi les technologies domotiques par courant porteur, on trouve la solution « In One » de Legrand, le X2D de Delta Dore, le PCLBUS, des passerelles pour le protocole KNX et Lonworks ainsi que le protocole X10 [Fig.5]. Le X10 est un excellent moyen de découvrir la domotique très facilement et à moindre frais. Ce protocole a été développé à l'origine par Pico Electronics en 1975. On trouve actuellement différents fabricants de produits X10 comme XDOM, Marmitek, Powerhouse ou Home System. Le protocole X10 supporte aussi le sans fil par transmission radio (433Mhz) grâce à un convertisseur radio/porteuse.

En X10 vous retrouvez trois types d'équipement : les émetteurs, les récepteurs et les contrôleurs.



Les émetteurs permettent d'envoyer des messages X10 sur le réseau comme les télécommandes, certains interrupteurs ou les capteurs (de météo ou de sécurité). En général, les émetteurs fonctionnent par radio.

Les récepteurs correspondent à des actionneurs répondant aux ordres X10 parmi lesquels on retrouve les modules d'Appliance (contact On/Off) et les modules Lampe (qui gèrent en plus la variation de la tension). Chaque module (émetteur ou récepteur), dispose d'une adresse comportant le code « maison », lettre de A à P, et un code « unité », chiffre de 0 à 16 permettant d'adresser jusqu'à 256 adresses différentes.

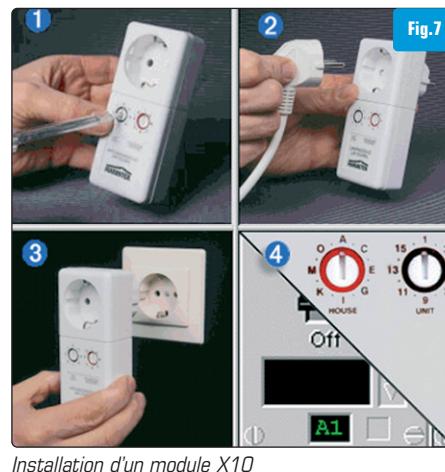
Chaque adresse est définie physiquement sur chacun des modules, et plusieurs modules peuvent avoir la même adresse (et



donc répondre aux mêmes ordres). Les récepteurs peuvent prendre la forme de module « gigogne », c'est-à-dire qu'ils s'intercalent entre la prise électrique et l'équipement à brancher (ou entre la douille et l'ampoule à visser) ou bien sous forme de module Rail DIN à installer dans le tableau électrique (ce qui rend l'installation plus discrète) [Fig.6].

Les contrôleurs permettent d'émettre et de recevoir des messages X10 sur le réseau électrique ou radio. Ils disposent généralement d'une interface USB ou TCP/IP pour être exploités par un PC, une tablette, etc. Bien que le X10 soit relativement peu coûteux et très simple à mettre en œuvre, il souffre de sérieux inconvénients, comme sa fiabilité et sa sécurité !

En effet, tout le monde peut émettre et écouter sur le réseau X10 ! Soit par les ondes soit par le réseau électrique, en branchant un simple contrôleur sur une prise électrique, vous pourrez donner n'importe quel ordre X10 sur le réseau ! Dans un immeuble on recommande d'ailleurs de mettre un filtre juste après l'arrivée EDF pour être sûr qu'aucun ordre émis dans l'appartement à côté ne puisse arriver chez vous ! Hormis la sécurité, c'est son degré de fiabilité



té qui nous pousse à réfléchir quant à son installation à travers toute la maison ! Tout d'abord certaines perturbations générées par les appareils électriques (TV, Hifi, PC, électroménager) brouillent les signaux X10 et l'ordre n'arrive pas à destination ! Ces perturbations électriques peuvent également déclencher des ordres X10 non souhaités ! Il m'est déjà arrivé d'avoir une lampe qui s'éteint ou s'allume toute seule ! Les ordres sont relativement longs à être transmis sur le réseau (50 bits/sec), et comme le protocole ne gère pas le retour d'état (accusé de réception), lorsque l'on donne un ordre (éteindre une lumière par exemple), il faut « croiser les doigts » pour que l'ordre soit bien transmis, bien reçu et bien exécuté par le récepteur. Aucune garantie ne sera donnée si ce n'est la vérification visuelle par soi-même (il existe aussi la possibilité d'envoyer au récepteur X10 un message de type « Query » lui demandant de renvoyer un message indiquant son état) [Fig.7].

Pour obtenir une bonne qualité de service avec le X10, la « best practice » est de privilégier l'usage des modules Rail DIN de manière à centraliser l'ensemble des récepteurs dans le tableau électrique afin de minimiser la distance à parcourir entre le



Module X10 sur Rail DIN

contrôleur et les récepteurs (et éviter l'exposition aux perturbations électriques sur réseau). Mais cela demande comme pour une installation KNX, de câbler en étoile son réseau électrique pour ramener individuellement chaque prise/lampe au tableau. Dans le cas d'un logement existant il est donc impossible à mettre en œuvre sans entamer un lourd chantier de rénovation ! [Fig.8].

> Les technologies « sans fil »

Comme il est difficile d'équiper son habitat avec une technologie filaire (demandant de revoir complètement le câblage) ou que son réseau électrique ne soit pas propice aux technologies par courant portant, la dernière possibilité reste l'utilisation du « sans fil » ! Comme nous l'avons vu, il existe des exten-


Produits DIO de Chacon

sions aux technologies domotiques filaires ou par courant porteur pour fonctionner sans fil. C'est le cas du KNX qui propose une passerelle « KNX radio » permettant d'ajouter des interrupteurs (contrôleurs) sans fil lorsqu'il n'est pas possible de les connecter sur le bus de commande. Il en va de même avec le X10 qui laisse la possibilité de rajouter des équipements sans fil, pratiques pour les détecteurs de sécurité (souvent placés à des endroits où il n'y a pas de prise de courant à proximité). Aussi, les gammes de produits déjà citées, comme celles de Legrand (InOne), Insteon, Lonworks ou encore X2D de Deltadore, proposent des extensions sans fil. Il faut cependant les considérer comme des « extensions » venant étendre un réseau domotique filaire ou CPL, et non comme la possibilité de créer un réseau domotique 100% « sans fil » [Fig.9].

En entrée de gamme, il existe un ensemble de produits domotiques « grand public » que l'on trouve dans les grandes surfaces ou magasins spécialisés. Des packs généralement composés de prises gigognes et télécommandes permettent (pour moins de 30 euros) de télécommander quelques prises, très pratiques pour contrôler les lampes du salon. On y trouve la technologie HomeEasy utilisée notamment par les produits Chacon que l'on retrouve un peu partout en France. Cette technologie monodirectionnelle peut être pilotée « informatiquement parlant » grâce à des émetteurs/récepteurs de trame RF 433Mhz comme avec le RFXCOM (en version USB à 95 euros ou en version LAN pour 200 euros). Le RFXCOM permet


RFXCOM en version USB

de piloter des produits X10, Insteon, sondes Oregon Scientific, etc [Fig.10].

Chez Somfy, Velux, Assa Abloy (serrure électronique) ou encore Honeywell, on utilise la technologie IO Homecontrol, plus haut de gamme, avec l'avantage d'être bidirectionnelle. On peut noter aussi l'arrivée en janvier 2012 de Castorama sur le marché avec la gamme de produit Blyss (projet « liveez Technology »), qui propose une box domotique sans fil (nommée la Blyssbox) et un catalogue assez riche de modules émetteurs et récepteurs sans fil.

Depuis 2010, on a pu voir apparaître en France une technologie prometteuse : EnOcean. Cette société allemande a inventé une technologie de communication sans fil et sans pile ! Celle-ci est généralement utilisée pour les capteurs et émetteurs. Elle capte


Interrupteur EnOcean

l'énergie de l'environnement, soit de la lumière (cellule photovoltaïque), soit d'une différence de température ou encore de la pression mécanique sur un interrupteur. Il n'y a donc plus besoin de câble ni de pile ! Il faut compter environ 70 euros pour des interrupteurs EnOcean et environ 500 euros pour la centrale [Fig.11].

La fiabilité reste le principal défi du « sans fil ». En effet comment pouvons-nous être certains que l'ordre arrive bien à sa destination ? Comment fait-on dans un grand logement, avec différentes contraintes (murs porteurs, canalisations, ...) qui affaiblissent considérablement le signal ?

Lorsqu'il s'agit d'équiper une seule pièce pour piloter quelques lampes grâce à une télécommande, cela ne pose pas de problème (la proximité entre la télécommande et les lampes limite les problèmes de transmission et si la lampe ne reçoit pas l'ordre, elle ne s'allumera pas. J'aurai donc une confirmation visuelle et dans ce cas, j'appuierai à nouveau sur le bouton de la télécommande). Mais lorsqu'il s'agit d'équiper entièrement

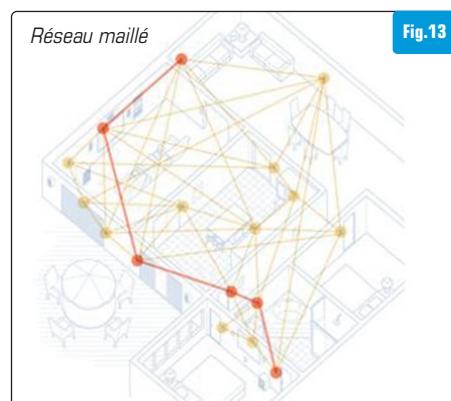
son habitat pour contrôler l'ensemble des équipements, il semble primordial d'avoir confiance sur le fait que l'action émise locale-


Prises télécommandables IDK

ment ou à distance sera quoi qu'il en soit bien exécutée ! Pour cela il est nécessaire d'utiliser une technologie bidirectionnelle à retour d'état, afin de confirmer au contrôleur et/ou à l'utilisateur que l'ordre a bien été réceptionné et exécuté par l'actionneur cible grâce à l'envoi d'un accusé de réception.

La sécurité peut aussi être une problématique. Par exemple, une télécommande de la marque IDK permettant de piloter jusqu'à 16 prises est sécurisée par la programmation d'un « code maison/unité » entre la télécommande et les prises. Mais sur ce produit, le code n'a que 16 combinaisons possibles ! Autant vous dire qu'il paraît très facile de piloter le réseau de votre voisin en essayant toutes les combinaisons de code un à par un (comme avec le X10 d'ailleurs, où il n'y a que 16 codes « maison » possibles, les lettres de A à P !) [Fig.12].

Enfin, pour résoudre les problèmes liés à la distance de transmission et contrer les différents obstacles, il est nécessaire d'utiliser des répéteurs/relais ou routeurs afin de couvrir l'ensemble de l'habitat. Dans cette lignée, il existe deux technologies très intéressantes : le ZigBee et le Z-Wave qui permettent de créer des réseaux maillés [Fig.13]. Dans cette topologie certains éléments du réseau (émetteurs ou récepteurs) peuvent se comporter comme des relais et ainsi


Fig.13



créer un réseau de maillage. Dans le cas où le contrôleur ne parvient pas à joindre directement le récepteur cible, il peut transmettre l'ordre à d'autres équipements, physiquement plus proches, qui se chargeraient de router le message jusqu'à « bon port » en utilisant un ou plusieurs nœuds. À l'image des protocoles de routage sur internet, chaque équipement échange avec ses voisins la liste des équipements qu'ils peuvent joindre directement et ainsi définir une table de routage pour joindre l'ensemble des nœuds du réseau. Lors de l'installation du réseau Z-Wave, il faudra garder à l'esprit cet aspect pour placer judicieusement ces équipements afin de couvrir au mieux son habitat et éviter les « zones d'ombre ».



Nous nous arrêterons sur la technologie Z-Wave que nous mettrons en pratique dans la suite de cet article. Pour la partie théorique, le Z-Wave a été développé par une société danoise Zensys, rachetée en 2008 par Sigma Design qui utilise la bande de fréquence de 868,42Mhz pour l'Europe. Bien que propriétaire, une alliance a été créée en 2008 et réunit aujourd'hui plus de 160 fabricants commercialisant des produits Z-Wave. Cela permet de garantir une certaine pérennité du protocole mais aussi un large choix de produits (on compte aujourd'hui plus de 600 produits Z-Wave sur le marché).

Le Z-Wave se définit comme une technologie bidirectionnelle à retour d'état permettant de garantir la bonne exécution des ordres. Très rapide (environ 50ms pour émettre un ordre et obtenir l'accusé de réception) et beaucoup plus sûre par l'utilisation d'un code sur 32 bits (soit plus de 4 milliards de combinaisons possibles) qui bloque tout équipement ne disposant pas du bon code. Enfin, comme décrit précédemment, sa topologie en réseau de maillage permet de couvrir aisément l'ensemble de l'habitat en prenant en compte les différentes contraintes physiques.

Un réseau Z-Wave se compose d'un ou plusieurs contrôleurs (clé USB, box Wifi/Ether-

net, interrupteurs ou télécommande) « maître » gérant la sécurité ainsi que le routage. Il est possible de gérer jusqu'à 232 périphériques Z-Wave (au-delà il faudra créer un autre réseau Z-Wave et établir un pont entre les réseaux). Les périphériques peuvent être des actionneurs (prises gigognes, relais encastrables, sirène, module pour volets, ou actionneur de chaudière ...) ou des émetteurs (de consommation, de sécurité, météo, interrupteurs, ou télécommandes). Pour lier les émetteurs/récepteurs au réseau Z-Wave (géré par un contrôleur), il faut les appairer en entrant dans le mode « association » sur le contrôleur, et en activant le périphérique à ajouter au réseau (appui sur un bouton ou une séquence de bouton en fonction du produit), dans un rayon de moins de 2 mètres afin de garantir une certaine sécurité quant à l'inclusion de périphériques étrangers ! C'est à ce moment-là que seront recalculées les tables de routage en tenant compte des nouveaux périphériques. C'est pourquoi il est préférable d'associer les périphériques à leurs emplacements définitifs [Fig.14].

Contrairement au ZygBee, à l'heure actuelle, le Z-Wave offre un large catalogue de produits et permet d'équiper sa maison pour un budget bien inférieur à certaines technolo-

gies filaires tout en garantissant une bonne fiabilité et sécurité (même si rien ne pourrait égaler les technologies filaires, l'utilisation d'un brouilleur à proximité bloquerait l'ensemble du réseau Z-Wave !).

USB. Le module dispose d'une carte SIM intégrée et donc d'un numéro de téléphone associé, permettant de lui envoyer des SMS pour allumer ou éteindre la prise commandée ou de lui demander son état et sa consommation électrique (le module vous répondra par SMS). Vous pourrez définir jusqu'à 7 numéros de téléphone pour définir les personnes ayant le droit de piloter chaque « MyPlug ». De plus, elle dispose d'une batterie intégrée pour vous avertir par SMS lors de coupure et du rétablissement du courant. Enfin, il existe une fonction localisant un porte-clés ZigBee : en envoyant « qui », le MyPlug peut vous répondre s'il « voit » ou non le porte-clés dans son champ ! Cette solution est disponible pour 79,99 € avec un module « MyPlug », un porte-clés et 3 ans de SMS illimités [Fig.15].

> Solutions retenues

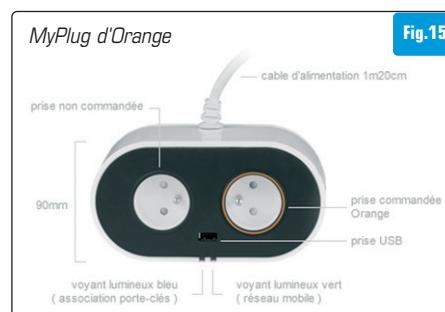
Le choix de la technologie doit s'étudier avec sérieux car il est difficile d'en changer ! Le plus important consiste à choisir en fonction de ses besoins, en privilégiant la facilité d'utilisation : peu importe les différents moyens de contrôle (télécommande, Smartphone, tablette ou même agent intelligent), il est important de conserver les traditionnels interrupteurs qui ne manqueront jamais de piles et seront facilement localisables et utilisables par tous de manière intuitive !

Il faut surtout prendre en compte les contraintes techniques et budgétaires pour choisir sa technologie et envisager l'utilisation de plusieurs technologies pour répondre à des besoins différents bien que cela ait tendance à complexifier l'installation.

L'idéal serait l'utilisation du KNX pour son standard et sa fiabilité, mais d'un point de vue technique (impossible de modifier le câblage sans engendrer de lourds travaux) et budgétaire (environ 10% du prix de l'habitat), cela paraît difficile à mettre en œuvre. La deuxième solution retenue peut être aisément déployée à l'ensemble de l'habitat : la technologie Z-Wave que nous mettrons en pratique dans la 3e partie de cet article.

Néanmoins, pour démarrer à moindre coût dans la domotique, intéressons-nous au développement avec la technologie X10.

Dans la seconde partie, nous mettrons les mains dans le code !!! Rendez-vous dans le n°155 de Programmez !



gies filaires tout en garantissant une bonne fiabilité et sécurité (même si rien ne pourrait égaler les technologies filaires, l'utilisation d'un brouilleur à proximité bloquerait l'ensemble du réseau Z-Wave !).

Pour finir, nous citerons les récentes percées d'Orange dans la domotique : il y a quelques mois, avec l'annonce du support du protocole ZigBee dans la dernière LiveBox Pro qui offrira à terme la possibilité de contrôler des équipements (et arrivera très certainement dans la prochaine LiveBox grand public). Mais aussi avec la dernière solution d'Orange nommée « My Plug », sortie en Mai 2012, qui se compose d'un module avec 2 prises de courants (dont une commandée) et d'une prise d'alimentation

Sébastien Warin
CTO xBrainSoft
[Http://sebastien.warin.fr](http://sebastien.warin.fr)

La maison connectée : puissance x 10

Commençons par une petite définition. Une maison connectée, c'est avant tout une maison équipée de capteurs pour lui permettre de comprendre son environnement et également une maison équipée d'actionneurs lui permettant d'agir sur ce qui l'entoure. Enfin, c'est une maison reliée à un ordinateur pour lui permettre de gérer l'intelligence des comportements programmés.

A l'heure actuelle, seuls 1% des logements sont équipés d'un système domotique. Si on laisse de côté l'aspect « Geek » d'une maison connectée, elle est censée apporter à la famille qui l'occupe confort, sécurité et économie. Ce sont toutes ces raisons que vous pourrez largement exploiter pour convaincre votre entourage lorsque la facture, qui résultera de la lecture de ce document, arrivera chez vous.

Lors de la session « Geek in da house » aux Techdays 2012, nous avions présenté, avec mes compagnons de « Coding » plusieurs équipements pour vous permettre de créer des appareils connectés. Nous allons évoquer toutes les étapes qui vont vous permettre de connecter les appareils électriques de votre maison. Nous aborderons les équipements, le montage, la programmation et les problèmes que vous pourriez rencontrer pendant le montage des modules X10. Nous parlerons également des nouveaux appareils existant sur le marché et qui sont là pour s'occuper de votre maison à votre place, que vous les programmiez ou pas. Cet article ne se veut pas exhaustif, il n'est clairement pas possible d'aborder la totalité des produits X10 disponibles, en revanche je vous propose de détailler ceux que j'utilise actuellement et qui équipent mon domicile. Vous pourrez retrouver les sections de codes illustrant ce document dans les sources que je mets à votre disposition sur un post que j'ai créé à l'adresse ci-dessous. Vous pourrez également y trouver les différentes bibliothèques, fichiers d'installation et autres tutoriels dont vous aurez besoin pour approfondir le sujet : <http://www.proteus-groupe.com/MessagesLanguages/Details?id=Connected-House-Evolution>

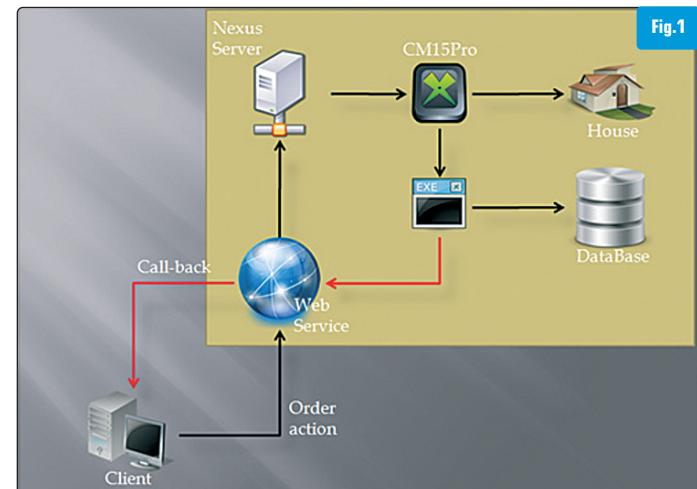
Pour avoir expérimenté les châtaignes de courant électrique, je vais vous demander de couper l'électricité des lignes sur lesquelles vous allez travailler. La coupe retro punk, c'est rigolo, mais le but principal est de rester en vie, évitez à « Programmez » de perdre des lecteurs alors qu'ils étaient en train d'expérimenter ce sujet 😊

> Les fournisseurs

Il en existe pléthore sur le Web. Coté domotique : EuroX10, La maison du numérique, IntelliHome, etc. Coté robotique : GHI Electronics, RoboShop, Génération Robot, etc. Pour ma part, les deux fournisseurs que j'utilise pour des raisons de disponibilité, rapidité et conseils sont Génération Robot et La maison du numérique. Je n'ai jamais eu de problème de livraison avec mes produits.

> Comment ça marche ?

Nous allons examiner en détail l'installation que j'ai mise en place chez moi, cela vous permettra d'adapter la configuration à vos besoins. N'étant pas le fils caché de Crésus je n'aborderai pas toutes les solutions de domotique disponibles sur le marché. En revanche je vais vous parler d'une solution de domotique X10 portée par Marmitek que je commence à plutôt bien connaître. L'avantage



de cette solution est qu'elle s'ajoute à votre installation électrique actuelle sans pour autant vous forcer à prendre un électricien sous CDI. Pour effectuer ce travail, nous aurons besoin d'un serveur (n'importe quelle machine ayant une prise USB et pouvant stocker un service WCF fera l'affaire), d'une base de données et pour finir d'un client (n'importe quel client, qu'il soit Windows Phone, WPF ou une application WEB sans distinction) [Fig.1].

LE CM15 PRO

Pour commencer à jouer avec l'électricité de la maison il nous faut une centrale. Le CM15PRO est l'appareil qui va servir d'interface entre les différents modules X10 et votre PC. Le CM15PRO est relié à votre PC par le port USB de votre ordinateur. Ce premier appareil vous reviendra à 78,90 € ; dans le package de ce produit vous aurez également un CD d'installation plus ou moins utile qui contient le logiciel « Active-Home ».

> Installation

Ici rien de bien compliqué, il suffit juste de connecter votre CM15 à votre ordinateur via un port USB. Le vrai problème est de bien choisir la prise électrique sur laquelle vous allez brancher l'appareil. Selon la configuration électrique de votre maison, toutes les prises ne permettront pas à votre CM15 de communiquer avec toutes les pièces. Typiquement, les prises dont l'alimentation se fait via un interrupteur mural sont sur une ligne qui peut bloquer la communication des signaux X10. Il faut aussi garder à l'esprit que certains appareils peuvent générer des interférences et empêcher la communication avec les modules à travers le réseau électrique, par exemple certaines télévisions, aspirateurs ou halogènes. Pour pallier ce problème, il vous suffit de vous procurer des Filtres « FM10 » [Fig.2].



Fig.2



> Configuration

Ici je vous propose deux solutions, une solution sans code pour permettre à un utilisateur final de ne pas avoir à se tracasser l'esprit avec le Framework .Net. Il n'aura qu'à brancher ces modules, installer l'application et jouer avec ces derniers. La deuxième solution que je vous propose est de s'abstraire de la couche applicative de « Active Home Pro », elle permettra à l'obsessionnel du code de se lâcher sur l'interface utilisateur.

Le code ce n'est vraiment pas mon truc !

Prenez le CD, installez le logiciel « Active-Home », lorsque l'installation est effectuée, ouvrez la console « Active-Home Pro ». Ici, vous pourrez créer des pièces et pour chacune d'entre elles, vous allez pouvoir affecter des modules X10. Lorsque vous allez créer ces différents boutons il faut que vous gardiez sous le coude l'adresse électrique de l'appareil [Fig.3].



The diagram shows a circular layout for a 15-unit house. The main circle is divided into 15 equal segments, each representing a unit. A smaller circle to the right represents an additional 1-unit extension, labeled 'UNIT' at the bottom. The segments are labeled with numbers: 15, 1, 3, 13, 11, 9, and 7. The segments are color-coded: the main circle has alternating segments in light blue and light green, while the extension circle has alternating segments in light blue and light yellow.

Pour information, lorsque je parle d'adresse électrique je parle de l'adresse affichée sur la partie configurable de votre module à savoir ceci : [Fig.4].

Je fais tout moi-même !

Vous allez quand même devoir installer le CD, et oui, pas le choix. Une fois l'installation effectuée, installez le SDK de « Active Home ». Pour notre installation, nous allons faire de l'ordinateur sur lequel est connecté le CM15Pro notre « Home Server », nous allons créer un service WCF afin de l'exposer à travers le WEB. Notre service sera chargé de recevoir les ordres des différents clients et de les router via la ligne de commande « SendPlc ». Si vous ne souhaitez pas le réécrire dans son intégralité, utilisez le code source disponible sur le site.

> **Communication**

Le CM15Pro est un Emetteur / Récepteur, il va convertir les signaux reçus par votre ordinateur pour les envoyer à travers le réseau électrique. Le CM15 bénéficie de deux canaux : le premier via le courant électrique, le second via les ondes RF. Si on prend l'exemple du MS13E, il informe le CM15 via les ondes RF de la présence de quelqu'un et/ou de lumière. L'appareil fonctionnant sur batterie, il s'agit de son seul moyen de communication. Maintenant que nous savons comment fonctionne l'appareil côté matériel, nous allons mettre en place notre service. Nous allons également installer une application qui va se charger d'écouter l'appareil afin de nous transmettre des informations qui transitent entre les différents appareils X10.

LE SERVICE WCF

Le choix d'un service WCF pour gerer notre CM15Pro s'impose de lui-même, une fois que vous avez mis votre service en place, il ne

vous reste plus qu'a rediriger un port de votre routeur vers le poste qui héberge votre service et il est accessible de partout. Dès lors, vous pouvez tout imaginer pour le client : un site web, une application Windows Phone, une application WPF, ou tout autre client concurrent 😊. Nous parlerons des clients un peu plus bas, aussi, pour l'instant restons concentrés sur le Service. Ce qui va nous intéresser c'est la signature de l'interface et les différentes énumérations pour ne pas etre perdu.

L'interface

```
using System.ServiceModel;
using Proteus.Services.HouseController.Contracts;

namespace Proteus.Services.HouseController
{
    [ServiceContract]
    public interface IX10Manager
    {
        [OperationContract]
        void SendOrder(HouseCodeEnum houseCode, int portCode, Order
Enum order, int percent);
    }
}
```

HouseCodeEnum

Les lettres correspondent aux différents codes maison que vous allez voir sur les modules X10.

```
using System.Runtime.Serialization;

namespace Proteus.Services.HouseController.Contracts
{
    [DataContract(Name = «HouseCode»)]
    public enum HouseCodeEnum
    {
        [EnumMember]
        None,
        [EnumMember]
        A,
        [EnumMember]
        B,
        [EnumMember]
        C,
        [EnumMember]
        D,
        [EnumMember]
        E,
        [EnumMember]
        F,
        [EnumMember]
        G,
    }
}
```

```

    [EnumMember]
    H,
}
}

```

OrderEnum

Vous utiliserez les ordres On/Off pour allumer et éteindre les lumières. Les ordres Dim/Bright sont utilisés pour vos volets.

```

using System.Runtime.Serialization;

namespace Proteus.Services.HouseController.Contracts
{
    [DataContract(Name = «Order»)]
    public enum OrderEnum
    {
        [EnumMember]
        None,
        [EnumMember]
        On,
        [EnumMember]
        Off,
        [EnumMember]
        Dim,
        [EnumMember]
        Bright,
    }
}

```

La base de données

Un peu de persistance pour l'état de vos modules. Le vrai problème dans le X10 est qu'on n'a pas la possibilité de faire des requêtes d'état. Cela veut dire que, si vous avez une lumière allumée, vous n'avez pas la possibilité de savoir si elle est allumée ou non. Vous serez confrontés au même problème avec vos volets : si vous les baissez, vous ne saurez pas si ils sont complètement baissés ou pas. Nous allons donc créer une base de données pour faire persister les ordres que nous aurons envoyés. Les scripts de création de tables sont également disponibles sur le post [Fig.5].



Fig.5

- > **Rooms** [Fig.6]
- > **Types** [Fig.7]
- > **Equipements** [Fig.8]

LA MAISON ET SES MODULES X10

Les lumières – LM15ES, LM12, LM12W [Fig.9]

Chez Marmitek, on aime les lumières : il y a une quantité astronomique de modules qui permettent de les connecter à des modules X10. La douille LM15ES et le LM12 sont vraiment les plus simples à poser. Dans le premier cas, vous dévissez votre ampoule, vous vissez le module, puis vous revissez votre ampoule dessus. Dans le deuxième cas, il vous suffit de connecter l'éclairage à cette prise. Pour ce qui est du tarif, il vous en coûtera respectivement, 18 €, 24 € et 30 €.



Fig.9

> Installation

Les installations du LM15ES et du LM12 ne sont pas d'une grande complexité. En revanche, je vous recommande de prendre du temps pour l'installation du LM12W. La douille LM15ES s'installe entre votre douille électrique et votre ampoule. La prise LM12 s'interpose entre votre prise électrique et votre appareil.

> Configuration

Pas de surprise, la configuration des modules est plutôt simple sauf pour la douille : je vous invite à vous rendre sur la fiche *descriptif du produit* pour effectuer la chorégraphie, euh... la séquence de configuration.

> Communication

Partons du principe que notre module X10 est configuré avec l'adresse « A2 ». On crée une instance de notre service et on envoie l'ordre d'allumage.

Résultats					Messages				
GuidValue	Name	Code	Position	Icone	GuidValue	Name	Code	Position	Icone
1 F0CE11B4...	Toilette	F	8	NULL					
2 458F20C5...	Bureau	G	3	NULL					
3 D6E24D10...	Salon	C	1	NULL					
4 48531902...	Chambre parent	D	5	NULL					
5 91FD65F5...	Cuisine	A	2	NULL					
6 95E23198...	Salle de bain	H	7	NULL					
7 CD3919EE...	Chambre d'enfant	E	4	NULL					
8 DCBE5D69...	Couloir	B	6	NULL					

Fig.6

Résultats					Messages				
GuidValue	Name	Position	Icone	GuidValue	Name	Code	Position	Icone	
1 46DAC9DF...	Sensor	3	NULL						
2 54DD5272...	Store	1	NULL						
3 C3992FAE...	Light	2	NULL						

Fig.7

Résultats					Messages				
GuidValue	GuidRoomValue	GuidTypeValue	Name	Code	State	StoreLevel	GuidValue	Name	Code
1 A149368D...	48531902-E46E...	C3992FAE-1C9...	Lumières	2	NULL	NULL			
2 A9FA991C...	91FD65F5-A868...	46DAC9DF-22...	Détecteur de mouvement	6	NULL	NULL			
3 0EF29B30...	458F2C05-A59F...	C3992FAE-1C9...	Lumières	2	1	NULL			
4 2050B741...	F0CE11B4-0E5...	46DAC9DF-22...	Détecteur de mouvement	6	NULL	NULL			
5 F13F43B3...	458F2C05-A59F...	54DD5272-9A...	Volets électriques	1	NULL	4			
6 D8D9A10B...	D6E24D10-433...	46DAC9DF-22...	Détecteur de mouvement	6	NULL	NULL			
7 D58EBCB3...	95E23138-B7D...	46DAC9DF-22...	Détecteur de mouvement	6	NULL	NULL			
8 0864D404...	91FD65F5-A868...	C3992FAE-1C9...	Lumières	2	1	NULL			
9 203D4CB7...	458F2C05-A59F...	46DAC9DF-22...	Détecteur de mouvement	6	NULL	NULL			
10 B9293135...	FC0E11B4-0E5...	C3992FAE-1C9...	Lumières	2	NULL	NULL			
11 D6945E25...	95E23138-B7D...	C3992FAE-1C9...	Lumières	2	NULL	NULL			
12 65B43887...	CD3991EE-D99...	C3992FAE-1C9...	Lumières	1	0	NULL			
13 CC8CA33E...	D6E24D10-433...	54DD5272-9A...	Volets électriques	1	NULL	-1			
14 71CB5380...	CD3991EE-D99...	46DAC9DF-22...	Détecteur de mouvement	6	NULL	NULL			
15 659529F...	DCBE5D69-835...	46DAC9DF-22...	Détecteur de mouvement	6	NULL	NULL			
16 D6158E1...	48531902-E46E...	46DAC9DF-22...	Détecteur de mouvement	6	NULL	NULL			
17 28405DC7...	D6E24D10-433...	C3992FAE-1C9...	Lumières	2	NULL	NULL			
18 6F153485...	DCBE5D69-835...	C3992FAE-1C9...	Lumières	2	NULL	NULL			
19 7F096569...	91FD65F5-A868...	54DD5272-9A...	Volets électriques	1	NULL	0			
20 52842F37...	48531902-E46E...	54DD5272-9A...	Volets électriques	1	NULL	0			
21 C2A63449...	CD3991EE-D99...	54DD5272-9A...	Volets électriques	2	NULL	1			

Fig.8



```
ServiceReference1.X10ManagerClient client = new ServiceReference1.X10ManagerClient();
client.SendOrder(ServiceReference1.HouseCode.A, 2, ServiceReference1.Order.On, 0);
```

Pour l'éteindre, on modifie la ligne de code « SendOrder » pour qu'il prenne en paramètre l'ordre d'extinction.

```
client.SendOrder(ServiceReference1.HouseCode.A, 2, ServiceReference1.Order.Off, 0);
```

Les volets - SW10G [Fig.10]

Lorsque l'on parle de domotique, la seconde priorité après les lumières est le contrôle des volets, pour plusieurs raisons : gérer l'ouverture et la fermeture automatique à une certaine heure et surtout pouvoir commander les volets sans avoir à utiliser l'interrupteur d'origine. Pour l'acquisition de ce module il vous en coûtera 55 € (j'admet que la facture dudit appareil reste un peu salée).



Fig. 10

> Installation

L'installation de l'interrupteur est des plus simples. Cependant, pour cette opération, vous allez devoir couper le disjoncteur. Le commutateur est encastrable. Si votre mur ne vous permet pas de l'encastrer, un petit aller-retour rapide dans le magasin de bricolage le plus proche de chez vous pour acheter un support à 9 € sera obligatoire.

> Configuration

Ici, pas de magie noire, il suffit de retirer la façade de l'interrupteur et vous tomberez nez à nez avec les commutateurs de configuration pour paramétriser l'adresse X10 de l'appareil.

> Communication

Sur les précédents modules vous avez vu qu'on utilisait un ordre On/Off pour allumer ou éteindre la lumière. Même s'il est possible d'envoyer le même ordre au SW10, il convient d'utiliser la commande Dim/Bright pour monter ou descendre les volets. A ceci, il faut ajouter un paramètre supplémentaire qui correspond à la valeur de l'action. Si on part du principe que notre SW10G est sous l'adresse électrique « H1 », alors l'ordre envoyé sera :

```
ServiceReference1.X10ManagerClient client = new ServiceReference1.X10ManagerClient();
client.SendOrder(ServiceReference1.HouseCode.H, 1, ServiceReference1.Order.Bright, 50);
```

Pour effectuer l'ordre inverse, il ne vous reste plus qu'à changer l'ordre en passant de "Bright" à "Dim"

```
client.SendOrder(ServiceReference1.HouseCode.H, 1, ServiceReference1.Order.Dim, 50);
```

La sécurité - DéTECTEURS MS13E. [Fig.11]

Ce module est le détecteur de présence de chez Marmitek, il est capable d'informer le réseau d'une présence mais également du changement de lumière dans l'endroit où il est installé. Pour l'acquisition de cet appareil il vous en coûtera 28 €.



> Installation

Ce module fonctionne sur pile, il s'installe donc où vous en avez besoin. Gardez cependant à l'esprit que les piles ne sont pas éternelles et qu'il faudra tôt ou tard les changer. Lorsque vous serez avertis, faites en sorte qu'il soit possible de le décrocher de son emplacement. Avant d'installer l'appareil pensez à bien configurer l'adresse X10 du module.

> Configuration

Par défaut, le code X10 de l'appareil est A1, mais vous pouvez le changer facilement pour qu'il corresponde à la disposition des pièces dont vous avez besoin. Référez-vous à la documentation en ligne pour effectuer l'opération.

> Communication

Les signaux du MS13E transitent via les ondes RF. L'application d'origine « ActiveHome Pro » est capable de vous informer que ce module a détecté quelque chose. En revanche pour les acharnés du code, il vous faudra utiliser la bibliothèque « Interop.ActiveHomeScriptLib » et écouter les informations qui transitent entre votre CM15 et le MS13. Nous en reparlerons juste après quand nous allons aborder l'exécutable d'écoute de votre réseau X10.

L'EXÉCUTABLE DE SURVEILLANCE

Comme je vous l'ai expliqué, nous n'avons pas de gestion d'état sur les modules X10, en revanche le MS13E (le détecteur de mouvement), lui, est capable de renseigner sur la présence ou non de quelqu'un dans une zone et également d'informer de la présence de lumière dans cette même zone. Cependant, pour avoir accès à ces données, il faut pouvoir écouter les messages qui transitent entre le CM15 et le MS13. Pour ce faire, nous allons créer un petit exécutable rapide qui aura pour travail de mettre en base de données l'état des modules X10.

Commencez par ajouter une référence à l'interop. Créez ensuite une instance de l'objet « ActiveHome ». Tous les ordres passés au CM15 déclencheront l'événement « RecvAction ». Une petite explication s'impose quand même : vous trouverez dans le constructeur de votre méthode des paramètres « Bsz ».

```
ActiveHome _home = new ActiveHome();
_home.RecvAction += new _DIActiveHomeEvents_RecvActionEventHandler(HomeRecvAction);
```

- bszParam1 : Contient le code X10 de l'appareil.
- bszParam2 : Contient l'ordre qui a été transmis entre ledit appareil et votre CM15.
- bszParam3 : Contient la valeur du paramètre qui a été transmis à l'appareil X10.

Pour modifier l'état des modules dans votre base de données, nous allons utiliser « Entity Framework », ajouter un nouvel élément à votre projet (ADO.Net Entity Data Model) et faire pointer la configuration de votre EDMX vers la base de données que nous avons créée plus tôt. Il ne vous reste plus qu'à vous positionner dans la méthode déclenchée par l'événement « RecvAction ».

```
private void HomeRecvAction(object bszAction, object bszParam1,
object bszParam2, object bszParam3, object bszParam4, object bszParam5,
```

```

object bszReserved)
{
    Models.HouseControllerEntities entity = new Models.HouseControllerEntities();
    string houseCode = bszParm1.ToString().Substring(0, 1).ToUpper();
    int moduleCode = Convert.ToInt16(bszParm1.ToString().Substring(1, 1));
    Models.Equipements equipement = entity.Equipements.Where(o => o.Rooms.Code == houseCode && o.Code == moduleCode).FirstOrDefault();
    if (bszParm2.ToString() == «On»)
        equipement.State = true;
    else if (bszParm2.ToString() == «Off»)
        equipement.State = false;
    entity.SaveChanges();
}

```

Pour finir, demandez à votre exécutable de notifier votre service qui se chargera de notifier à son tour tous les clients qui sont connectés via la fonction de « Call-back ».

LES CLIENTS

Nous avons parlé du serveur, nous avons parlé des différents modules X10, maintenant nous allons attaquer la partie cliente.

Une application ASP.Net MVC

Intégrer la gestion de la maison sur mon blog n'avait rien de compliqué. Après avoir ajouté une icône pour accéder à cette nouvelle page « House Controller », il ne me restait plus qu'à sécuriser l'accès à cette page afin que seules les personnes ayant les droits puissent y accéder [Fig.12]. Pour récupérer la liste des équipements une simple requête via « Entity » :

```

[AuthorizationCustom(RoleList = new[] { Role.Administrateur })]
public ActionResult Index()
{
    var rooms = _entity.Rooms.Include(«Equipements»).OrderBy(o => o.Position);
    return View(rooms.ToList());
}

```

[Fig.13] Pour finir, on attache à chaque équipement un « Action Link » spécifique qui viendra remplir les variables nécessaires à l'exécution de cette méthode avant de faire appel au service WCF, qui lui, transmettra l'ordre à notre CM15PRO.



Fig.12



Fig.13

```

[AuthorizationCustom(RoleList = new[] { Role.Administrateur })]
public ActionResult SendOrder(string HouseCodeValue, string ModuleCodeValue, string OrderTypeValue, string ParameterValue)
{
    X10ManagerClient client = new X10ManagerClient();
    client.SendOrder(GetHouseCode(HouseCodeValue), Convert.ToInt16(ModuleCodeValue), GetOrder(OrderTypeValue), Convert.ToInt16(ParameterValue));
    return RedirectToAction(«Index»);
}

```

Une application Windows Phone 7

Que ce soit pour une application Windows Phone ou une application WEB, l'appel au service WCF reste quasiment le même si ce n'est que sous Windows Phone l'appel à la méthode sera : «SendOrderAsync». Commençons par créer une instance notre service WCF :

```

_client = new X10ManagerClient();
_client.SendOrderCompleted += new EventHandler<SendOrderCompletedEventArgs>(ClientSendOrderCompleted);

```

Appel à la méthode du service WCF :

```

_client.SendOrderAsync(...);

```

> Gestion des luminaires [Fig.14]

Appuyez-vous sur le « Microsoft.Phone.Controls.Toolkit ». Dans cette librairie vous trouverez un contrôle qui devrait coller parfaitement à la gestion des lumières dans votre application, le « ToggleSwitch »

> Gestion des stores [Fig.15]

N'étant pas un adepte du design, je vous prie d'excuser le peu d'intérêt de cette interface qui permet de contrôler les volets. Ici les boutons envoient simplement l'ordre aux volets de s'ouvrir ou de se fermer en utilisant le palier fourni en paramètre.

Une application WPF

Et maintenant, du lourd pour notre application de domotique ! Pour notre application WPF nous allons éviter de mettre des boutons. Nous l'avons déjà fait avec Windows Phone et notre application WEB. Nous allons mettre en place un peu de reconnaissance vocale. Commençons par créer notre dictionnaire et instancier notre objet de reconnaissance vocale :

```

private SpeechRecognizer _speechReco;
public Listener()

```



Fig.14



```

{
    List<string> _grammerList = new List<string>();
    _grammerList.Add("Allume la lumière du salon");
    _speechReco = new SpeechRecognizer();
    _speechReco.SpeechRecognized += new EventHandler<SpeechRecognizedEventArgs>(EventSpeechRecognized);
    _speechReco.LoadGrammar(CreateGrammarObject(_grammerList.ToArray()));
    _speechReco.Enabled = true;
}

private Grammar CreateGrammarObject(string[] grammerList)
{
    Choices choices = new Choices(grammerList.ToArray());
    GrammarBuilder builder = new GrammarBuilder(choices);
    Grammar gram = new Grammar(builder);
    return gram;
}

```

Quand une commande vocale est interceptée par notre objet, nous ferons appel à notre service WCF afin qu'il effectue l'ordre demandé.

```

private void EventSpeechRecognized(object sender, SpeechRecognizedEventArgs e)
{
    if (e.Result.Text == "Allume la lumière du salon")
    {
        X10ManagerClient client = new X10ManagerClient();
        client.SendOrder(HouseCode.C, 3, Order.On, 0);
    }
}

```

PROTEUS, UNE ARMOIRE 42U COMPATIBLE MINDSTORM ! [Fig.16]

Dans un article paru au mois de Mars, je vous ai un peu parlé de ce qu'était Proteus, mais sans trop entrer dans les détails. Je vais vous expliquer ce que j'ai mis en place de mon armoire 42U avec l'aide d'un peu de Lego. Ce n'est pas tant l'armoire qui doit attirer votre attention mais plutôt l'intérieur. Quand on parle de domotique, on parle de gestion automatique de l'alimentation, allumage, extinction. Avec un peu d'imagination vous pourrez envisager des solutions que vous n'auriez jamais abordées. Quand des serveurs sont dans une armoire et qu'on veut les allumer ou les éteindre sans avoir à ouvrir la porte de ladite armoire, on peut faire appel au WOL (Wake On Lan) cela n'est pas toujours 100 % fiable. C'est ici qu'entre en jeu le « Lego MindStorm ».



Fig.15



Fig.16



Fig.17

Dans la boîte !

Dans la boîte du Lego MindStorm vous trouverez les éléments suivants :

- La brique NXT
- 3 Servomoteurs
- 4 Capteurs
 - 2 capteurs de contacts
 - 1 capteur de lumière qui distingue également les couleurs
 - 1 capteur ultrasonique

Encore plus de MindStorm !

Vous n'en avez pas assez avec ce qu'il y a dans la boîte ? Je vous invite à vous rendre sur le site de Génération Robot, là vous y trouverez des capteurs supplémentaires pour permettre à votre brique de faire davantage de choses.

Du Mindstorm, oui mais en C# !

Dans la fenêtre de contrôle pour la Brick NXT, vous pouvez télécharger le code source de cette application et l'adapter à vos besoins. La capture d'écran ci-dessous représente l'affichage des Capteurs [Fig.17]. Cet écran, quant à lui, représente l'écran de contrôle des Moteurs de la brique NXT [Fig.18].

Vous retrouverez également en téléchargement la bibliothèque d'AForgé pour communiquer avec votre brique NXT. Vous y trouverez également le code source de l'application que j'utilise pour contrôler les moteurs et tirer les informations nécessaires des différents capteurs. Dans cette armoire 42U chaque serveur est associé à un code couleur et grâce à deux servomoteurs, le plateau est capable de monter et descendre dans l'armoire. Une fois que le capteur de couleur a repéré la couleur correspondant au serveur pour lequel il a été sollicité, le dernier servomoteur oriente le capteur de contact vers l'interrupteur ON/OFF du serveur. Dès lors, vous n'êtes plus dépendant du Wake On Lan pour allumer vos serveurs. Plus encore, en cas de bug ou de défaillance de l'un d'entre eux, « MindStorm » vous permettra donc d'effectuer un « Hard Reset ». Toute cette opération est effectuée en C# faisant ainsi de la brique une extension du serveur de domaine [Fig.19].

JE SUIS DÉSOLÉ MAIS ÇA NE ME SUFFIT PAS...

Une fois que vous avez déballé le tapis d'arguments à votre famille pour justifier ce que vous venez de lire, il est possible que par le plus grand des hasards vous ayez encore du budget. Pour dépenser ce qui

vous reste, je vous propose de voir quels sont les produits existant sur le marché pour s'occuper de votre maison à votre place.

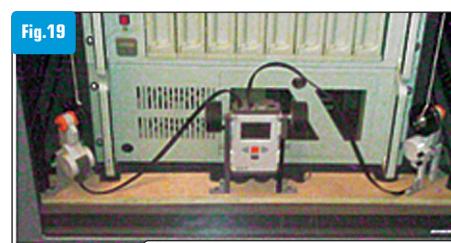


Fig.19

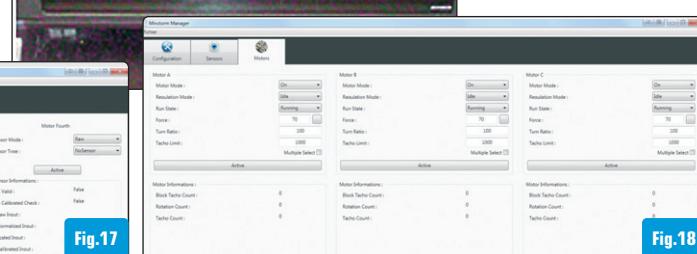


Fig.18


Fig.20
• Roomba (i-Robot) : 430 € [Fig.20]

Passer l'aspirateur, voilà bien une tâche obligatoire, heureusement reléguée aux tâches ménagères effectuées par des robots. i-Robot possède une gamme d'aspirateurs automatiques impressionnante pouvant convenir à différentes bourses, elle va de 300 € à 800 €. Le comble du nettoyeur automatique vous conduira quand même à une dernière tâche : le nettoyage de l'appareil une fois par semaine !


Fig.21
• Scooba (i-Robot) : 460 € [Fig.21]

Une fois que vous avez passé l'aspirateur, il faut encore laver le sol et là aussi c'est une tâche que vous pouvez confier à un robot. Scooba passe derrière le Roomba et lave votre sol pour vous. La gamme ne se compose que de 3 appareils pour l'instant dont les tarifs vont de 320 € à 500 €


Fig.22
• Windorobo : NC [Fig.22]

Rien avoir avec Windows, il s'agit bien d'un robot qui va vous délester de votre tâche de nettoyeur de vitre. En revanche, je vous suggère de ne pas tenter l'opération si vous habitez un bâtiment et que vous n'êtes pas au rez-de-chaussée. Actuellement à l'étude, le département de recherche de la firme travaille sur une version du robot capable de laver les deux côtés de la fenêtre, « Stay Tuned » !

Create (iRobot, plateforme de développement) [Fig.23]

Il s'agit ici de la plateforme de développement d'iRobot, la base des Roomba. Vous allez pouvoir créer votre propre version de l'aspirateur ou en faire complètement autre chose. Pour information, lors de la session 2011 d'Innorobo, Pierre Cauchois avait déjà montré comment utiliser cette plateforme pour en faire un appareil de télé-présence. A ce propos je vous invite à consulter l'article suivant : « <http://www.proteus-groupe.com/MessagesLangages/Details?id=ICreate> » vous y trouverez des renseignements sur la plateforme iCreate et sur sa programmation.


Fig.23
• Vortex (Zodiac) : 750 € [Fig.24]

Préoccupation principale de cet appareil, votre piscine. Equipé d'une brosse à lamelle pour soulever les débris situés au fond de votre bassin, il sera le compagnon idéal pour cette tâche de nettoyage !


Fig.24
• Eddie (Microsoft) : 1 245 € [Fig.25 et 26]

J'ai commandé cet appareil chez Génération Robot. Lorsque vous le recevez, vous vous retrouvez devant une boîte qui ressemble beaucoup à un puzzle à assembler. Une heure et quelques coups de tournevis plus tard, l'objet est assemblé, il ne reste plus qu'à laisser libre cours à votre imagination. Pour ne citer personne, certains ont trouvé le moyen de transformer cet appareil en arme de dissuasion pour empêcher les gens de squatter la machine à café : c'est équipé d'un fusil nerf, le robot traque ainsi les discussions ennuyeuses pour les


Fig.25

Fig.26

transformer en cibles vivantes. Si le développement autour de cette plateforme vous intéresse, je vous invite à vous rendre à cette adresse « <http://www.proteus-groupe.com/MessagesLangages/Details?id=Eddie> » afin d'approfondir le sujet.


Fig.27
• R40Li (Gardena) : 1 389 € [Fig.27]

A l'approche de l'été, encore plus avec la météo que nous avons eue jusqu'au mois de Mai, la pelouse pousse à une vitesse démesurée. La corvée de tondeuse est ultra chronophage, mais là aussi on peut la confier à un robot. Capable de s'occuper de surfaces allant jusqu'à 400 m², capable de se recharger automatiquement, il est équipé d'un écran LCD et d'un clavier pour vous permettre de configurer l'appareil.


Fig.28
• Nao (Aldebaran) : 3 600 € (si vous êtes « développeur ») [Fig.28]

Il y a quelques semaines, Aldebaran a réuni une petite armée de développeurs passionnés dans le seul but de permettre à Nao d'offrir plus d'interactions avec son environnement. Si le sujet vous intéresse, je vous invite à consulter le billet sur le « Nao Contest 2012 », cela devrait vous permettre de vous faire une idée plus précise sur les capacités de ces 60 centimètres d'électronique ! Pour ma part, en plus d'être un assistant qui gère mon compte Facebook, il est aussi une solution plus intéressante qu'une simple caméra pour effectuer de la télésurveillance. Une caméra oui, mais pas que cela. La mobilité de Nao la rend plus intéressante qu'une simple caméra sur IP. Il représente aussi une solution de télé présence efficace. En effet, avec un peu de .Net et d'huile de coude, peu importe où vous vous trouverez, vous pourrez voir le flux vidéo du robot et lui faire lire des messages que vous lui transmettrez. Pour terminer, il représente également le point d'entrée des ordres de domotique puisque les ordres, via la commande vocale passent par le robot avant d'être transférés au CM15.

Le mot de la fin !

La domotique n'en est qu'à ses débuts, il faudra du temps et de l'imagination pour voir la chose se généraliser et apparaître dans toutes les maisons. X10 est un moyen simple de connecter des appareils mais ce n'est pas le seul. Gadgeteer & Netduino représentent à mon avis des solutions efficaces pour connecter des appareils, nous aurons bientôt l'occasion d'en reparler. J'espère que cet article vous permettra d'aller jusqu'au bout de votre projet de domotique. Si jamais vous rencontrez des problèmes ou que vous avez des idées que vous souhaitez partager avec tout le monde, n'hésitez pas à venir laisser des messages sur le billet créé à cet effet.

Salah Amer - www.proteus-groupe.com

Les outils des Décideurs Informatiques

*Vous avez besoin d'info
sur des sujets
d'administration,
de sécurité, de progiciel,
de projets ?*

*Accédez directement
à l'information ciblée.*

Cas clients

Actu triée par secteur

Avis d'Experts



Actus / Evénements / Newsletter / Vidéos



www.solutions-logiciels.com

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

Envoyer par la poste à : Solutions Logiciels, service Diffusion, GLIE - 17 chemin des Boulangers 78926 Yvelines cedex 9 - ou par fax : 01 55 56 70 20

1 an : 50€ au lieu de 60€, prix au numéro (Tarif France métropolitaine) - Autres destinations : CEE et Suisse : 60€ - Algérie, Maroc, Tunisie : 65€ , Canada : 80€ - Dom : 75€ Tom : 100€

10 numéros par an.

M. Mme Mlle Société

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

NOM Prénom

N° rue

Complément

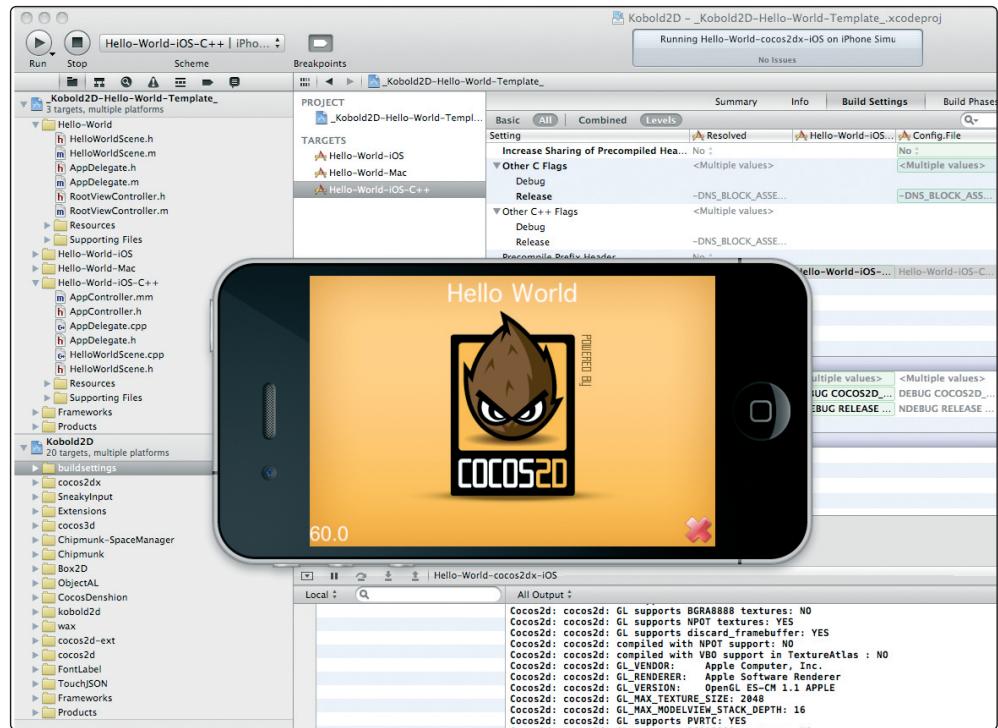
Code postal : | | | | | Ville

Adresse mail

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

Cocos2D, un moteur de jeu pour iOS

Depuis la sortie de l'iPhone puis de l'iPad, il est incontestable que ce qui a fait le succès de la plateforme repose en très grande partie sur la boutique d'applications mise à disposition par Apple pour sa plateforme mobile.



Parmi la multitude de catégories d'applications disponibles, celle concernant les jeux vidéo possède un attrait de plus en plus fort puisque de grands noms de l'industrie tels que Gameloft, Electronic Arts, Epic Games ou encore Ubisoft (pour n'en citer que quelques-uns) se mettent à réaliser des jeux pour iOS depuis quelque temps déjà. En tant que développeur, il est probable que l'idée vous ait traversé un jour l'esprit, de vouloir réaliser votre propre jeu en rêvant que celui-ci rencontre le même succès que les Flight Control, Angry Birds ou FarmVille.

> Quelle solution technique choisir ?

La première question à laquelle vous avez donc dû réfléchir est : « avec quelle technologie réaliser mon jeu ? » iOS propose de nombreuses API destinées à cet usage, parmi lesquelles nous pouvons citer CoreGraphics, CoreAnimation, OpenGL ES ou bien encore GLKit (qui a fait son apparition dans iOS 5). Ces API, bien que permettant de développer quasiment tous les types de jeux que vous pouvez trouver actuellement sur l'AppStore, ont un inconvénient majeur : la quantité de code devant être écrite sera extrêmement importante.

En effet, tout jeu digne de ce nom va avoir besoin de briques techniques qui sont à peu près toujours les mêmes quel que soit le type de jeu, à savoir : la gestion des scènes, des animations, des sprites, des effets graphiques, des interactions avec l'utilisateur, etc.

Deux solutions s'offrent donc à vous pour débuter le développement de votre jeu : partir d'une feuille blanche et tout réaliser par vous-même ou bien avoir recours à un moteur de jeu qui vous fera gagner un temps précieux en vous offrant un ensemble de briques techniques clé en main, que vous avez juste besoin d'assembler pour réaliser votre jeu.

Encore faut-il choisir le bon moteur de jeu car il en existe de très

nombreux sur le marché iOS : Cocos2D, Unity, UDK, Sparrow, SiO2, iTorque2D pour n'en citer que quelques-uns parmi les plus connus. La plupart d'entre eux sont payants et sont plutôt destinés à des publics avertis ayant de bonnes connaissances techniques, notamment dans le domaine de la 3D.

Si vous cherchez une solution OpenSource et que vous envisagez de réaliser votre jeu en 2D ou 3D isométrique (également appelée 2.5D), il existe toutefois une solution très intéressante, à savoir Cocos2D (<http://www.cocos2d-iphone.org/>).

> Présentation de Cocos2D

Cocos2D est comme son nom l'indique, un moteur destiné à la réalisation de jeux 2D (mais également 3D isométrique) et dont les premières versions sont sorties il y a maintenant de cela plus de 4 ans. La particularité de Cocos2D est qu'il est entièrement gratuit et OpenSource, statut qui a été conservé malgré l'intégration en Mai 2011 des deux développeurs à l'origine du projet au sein des équipes de Zynga (éditeur bien connu des jeux sociaux sur Facebook et mobiles).

A ce jour, deux versions du projet évoluent en parallèle à savoir une version dédiée à iOS (Cocos2D-iPhone) et une version nommée **Cocos2D-X** (prononcée Cocos2D-Cross) et destinée à être utilisable sur tous types de plateformes (iOS, Android, Windows Phone, HTML5...).

La version sur laquelle nous allons nous attarder dans cet article est celle destinée exclusivement à iOS (Cocos2D-iPhone) qui est actuellement disponible en **version 2.0 RC1**.

Comme évoqué précédemment, ce moteur de jeu va vous fournir de très nombreuses briques techniques, éprouvées depuis de nombreuses années sur de très nombreux jeux, vous permettant de gagner un temps précieux dans la réalisation d'opérations basiques

telles que l'affichage d'une scène, le remplacement d'une scène par une autre avec une transition (fondu, flip, scroll...), l'affichage et l'animation de sprites à l'écran, etc.

Outre les briques techniques fournies par le moteur, celui-ci est propulsé en arrière-plan par **OpenGL ES**, ce qui lui permet d'obtenir d'excellentes performances, notamment en nombre de « Frame per Second » (FPS), avec de très nombreux éléments affichés à l'écran comme cela est souvent le cas dans les jeux d'action (ex : de nombreux ennemis et tirs sur un Shoot'em Up).

A titre d'information, le site web de Cocos2D possède un annuaire des jeux utilisant ce moteur, et à ce jour quelques 3 686 jeux sont recensés à l'intérieur, sachant que l'AppStore en revendique pour sa part 34 106. Cela fait de Cocos2D l'un des moteurs de jeux les plus utilisés (si ce n'est le plus) de la plateforme iOS.

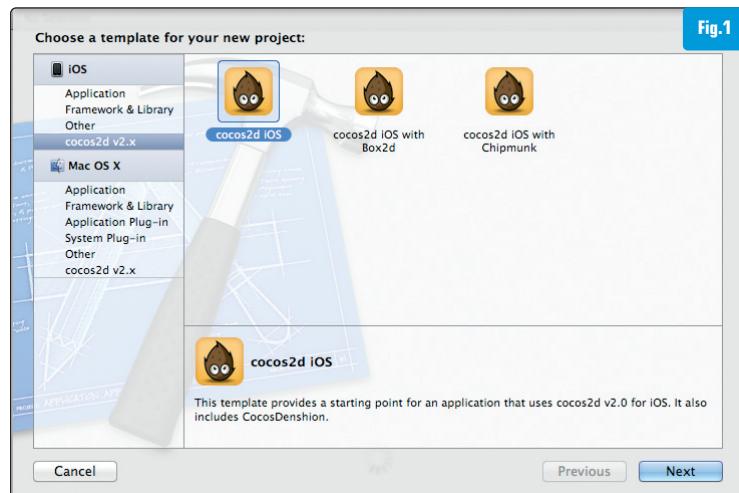
Un autre point fort de Cocos2D est la communauté qui s'est montée autour de ce moteur. En effet, de nombreux auteurs et dont le plus actif est probablement **Steffen Itterheim**, mettent à disposition de manière très régulière, des articles techniques sur leurs blogs, des ouvrages vendus dans les librairies, des extensions aux briques de bases, etc. Si une fonctionnalité n'est pas dans le moteur, nul doute que vous trouverez de l'aide pour la mettre en œuvre, si quelqu'un ne l'a pas déjà fait et publié sur son site Internet.

Après cette introduction, passons aux choses sérieuses et regardons de plus près ce que propose Cocos2D afin de nous permettre de réaliser un jeu. Nous ne réaliserons pas ici un jeu clé en main, mais nous passerons en revue tous les concepts importants vous permettant de créer votre premier jeu en quelques heures / jours selon vos compétences techniques.

> Concepts et briques techniques fournis en standard

La première chose qu'il est bon de noter après avoir téléchargé le fichier ZIP contenant tous les éléments nécessaires à l'utilisateur de Cocos2D, est que celui-ci contient des modèles de projets intégrés à Xcode et permettant de créer le squelette d'une application en quelques secondes [Fig.1].

Comme vous pouvez le voir sur la capture d'écran, trois types de projets existent. Nous nous servirons du premier pour créer une application afin d'expliquer les concepts de base. Les deux autres modèles de projets permettent de créer des applications utilisant un moteur physique (nous en reparlerons ultérieurement).



Après avoir créé une application en utilisant le modèle précédemment évoqué, ouvrons le fichier nommé « **AppDelegate.m** » et passons en revue les éléments importants de celui-ci.

Initialisation du moteur

Le premier concept important concerne la classe « **CCDirector** » qui définit le chef d'orchestre de tout ce qui va se dérouler ensuite dans l'utilisation du moteur. Quand vous aurez besoin de faire une opération simple (ex : afficher une scène), vous demanderez au directeur d'effectuer cette opération pour vous et de se charger de gérer tous les effets que vous pourrez avoir souhaités (ex : faire apparaître la scène avec un fondu durant 2 secondes).

Sur le code ci-dessous, vous voyez comment récupérer l'instance du « **Director** » qui est gérée sous forme d'un singleton, définir que votre application doit s'exécuter en plein écran, afficher les statistiques du nombre de FPS, définir que vous souhaitez avec un taux 60 FPS au niveau de vos animations et enfin que vous voulez activer la gestion de l'écran Retina.

```
director_ = (CCDirectorIOS*) [CCDirector sharedDirector];
director_.wantsFullScreenLayout = YES;
[director_ setDisplayStats:YES];
[director_ setAnimationInterval:1.0/60];
[director_ enableRetinaDisplay:YES];
```

Viens ensuite la création de la vue OpenGL avec différents paramètres tels que le format des pixels ou la profondeur de couleur souhaitée. Cette vue est ensuite associée au « **Director** » afin d'effectuer le rendu graphique des éléments que vous manipulerez dans votre jeu. L'intérêt principal de Cocos2D ici est que vous n'avez pas besoin de connaître OpenGL pour commencer à développer votre jeu. Des wrappers sont fournis afin de cacher toute la complexité qui se situe derrière l'utilisation d'OpenGL ES qui est faite par Cocos2D.

```
CCGLView *glView = [CCGLView viewWithFrame:[window_ bounds]
pixelFormat:kEAGLColorFormatRGB565
depthFormat:0
preserveBackbuffer:NO
sharegroup:nil
multiSampling:NO
numberOfSamples:0];

[director_ setView:glView];
```

Puisque la gestion de l'écran Retina a été activée, il est nécessaire à Cocos2D de savoir comment trouver les images haute et basse résolution, aussi bien pour l'iPhone que pour l'iPad. Pour cela, le moteur se base sur une convention de nommage à base de suffixes que vous donnez à vos fichiers.

Si par exemple vous souhaitez afficher une image d'arrière-plan nommée « *Background.png* », la convention par défaut (mise en œuvre avec le code ci-dessous) fait que le moteur va chercher des images nommées « *Background.png* » pour le mode iPhone basse définition, « *Background-hd.png* » pour le mode iPhone Retina, « *Background-ipad.png* » pour le mode iPad basse définition et enfin « *Background-ipadhd.png* » pour le mode iPad Retina.

Si jamais une des images n'existe pas, par défaut, le système essaie de se rabattre sur une autre image sauf si, vous avez indiqué de ne

pas utiliser le système de secours « `setEnableFallbackSuffixes :NO` ». Si vous respectez la convention mise en œuvre (vous pouvez définir la vôtre en changeant les suffixes ci-dessous), Cocos2D gérera pour vous l'affectation de la bonne image au bon contenu.

```
CCFileUtils *sharedFileUtils = [CCFileUtils sharedFileUtils];
[sharedFileUtils setEnableFallbackSuffixes:NO];
[sharedFileUtils setiPhoneRetinaDisplaySuffix:@"-hd"];
[sharedFileUtils setiPadSuffix:@"-ipad"];
[sharedFileUtils setiPadRetinaDisplaySuffix:@"-ipadhd"];
```

Une fois toutes les initialisations du moteur effectuées, il ne vous reste plus qu'à demander d'afficher la 1re scène de votre application. Pour cela, il suffit de demander au « `Director` » de la faire apparaître en appelant la méthode « `pushScene` » avec en paramètre, l'instance de la scène que vous souhaitez.

```
[director_ pushScene: [HelloWorldLayer scene]];
```

Quand vos utilisateurs quitteront votre jeu, il est important que toutes les actions qui pouvaient être en cours à ce moment-là et notamment les animations, soient mises en pause pour pouvoir être reprises ultérieurement dans l'état dans lequel elles étaient. C'est entre autres le rôle du « `Director` » de coordonner tout cela pour vous et il met à disposition les méthodes nécessaires pour mettre votre jeu en pause et le reprendre suivant les actions effectuées par l'utilisateur.

```
-(void) applicationWillResignActive:(UIApplication *)application
{
    [director_ pause];
}

-(void) applicationDidBecomeActive:(UIApplication *)application
{
    [director_ resume];
}

-(void) applicationDidEnterBackground:(UIApplication*)application
{
    [director_ stopAnimation];
}

-(void) applicationWillEnterForeground:(UIApplication*)application
{
    [director_ startAnimation];
}
```

Scènes et sprites

Comme évoqué précédemment, lors de son exécution un jeu présente une scène (un écran) à l'utilisateur pour que celui-ci interagisse avec elle, et il est possible d'enchaîner ensuite les scènes afin de définir un scénario à votre jeu (ex : passer d'un niveau à un autre, afficher un écran de paramétrage, etc.).

Pour cela, Cocos2D met à votre disposition les éléments nécessaires avec de nombreuses classes permettant de créer une scène en lui ajoutant différents éléments (layers, sprites, etc.).

Si nous reprenons notre application précédente et que nous ouvrons

le fichier « `HelloWorldLayer.m` », nous voyons la méthode permettant d'instancier une scène et de la récupérer afin de la faire afficher par le « `Director` ».

Le code ci-dessous utilise la classe « `CCScene` » et appelle la méthode « `node` » pour récupérer une instance de cette classe, puis ajoute un enfant à la scène, cet enfant étant lui-même instancié depuis la classe « `HelloWorldLayer` ».

```
+ (CCScene *) scene
{
    CCScene *scene = [CCScene node];
    HelloWorldLayer *layer = [HelloWorldLayer node];
    [scene addChild: layer];
    return scene;
}
```

Si nous regardons la méthode d'initialisation de la classe « `HelloWorldLayer` », nous voyons qu'un label utilisant une police TrueType est créé et ajouté à la scène, ce label étant positionné au milieu de l'écran. Les dimensions de l'écran sont fournies une nouvelle fois grâce au « `Director` ».

```
- (id) init
{
    if( (self=[super init]) )
    {

        CCLabelTTF *label = [CCLabelTTF labelWithString:@"Hello
World" fontName:@"Marker Felt" fontSize:64];

        CGSize size = [[CCDirector sharedDirector] winSize];
        label.position = CGPointMake( size.width /2 , size.height/2 );
        [self addChild: label];

        ... // Initialisation d'autres éléments dans la scène
    }
    return self;
}
```

Plutôt que d'ajouter un simple label, vous voudriez peut être probablement ajouter des images, comme par exemple une image d'arrière-plan et une image qui serait animée au-dessus de celle-ci.

Rien de plus simple, car Cocos2D nous fournit une nouvelle fois tous les éléments nécessaires pour réaliser cette opération grâce à la classe « `CCSprite` ».

```
CCSprite *backgroundSprite = [CCSprite spriteWithFile:@"Back
ground.png"];
[self addChild:backgroundSprite] ;

CCSprite *cloudSprite = [CCSprite spriteWithFile:@"Cloud.png"];
[self addChild:cloudSprite] ;
```

Grâce à la convention de nommage dont nous avons parlé précédemment, si vous ajoutez des images basse et haute résolution pour iPhone et iPad, le code écrit ci-dessus permettra à votre application de fonctionner sur les deux plateformes sans avoir à faire quoi que ce soit de plus, la méthode « `spriteWithFile` » allant automatique-



Fig.2

ment récupérer la bonne image en fonction du périphérique et de la résolution de son écran.

Animations des sprites

Maintenant que les sprites sont ajoutés à la scène, peut-être souhaitez-vous animer un minimum votre scène. Pour cela, un concept est proposé par Cocos2D vous proposant d'associer à vos objets (des sprites dans notre exemple mais ce n'est pas une obligation) des actions portant sur différents aspects de ceux-ci (déplacement, rotation, grossissement, réduction...).

C'est ce que nous allons mettre en place ci-dessous en animant de droite à gauche un de nos deux sprites, cette animation s'exécutant de manière perpétuelle. Pour cela nous avons à notre disposition de nouvelles classes « `CCMoveBy` », « `CCSequence` » et « `CCRepeatForever` » qui vont nous permettre de gérer les déplacements, d'enchaîner ceux-ci les uns par rapport aux autres, puis de demander leur exécution de manière perpétuelle.

```
CCMoveBy *moveRight = [CCMoveBy actionWithDuration:2 position:
CGPointMake(100, 0)];
CCMoveBy *moveLeft = [CCMoveBy actionWithDuration:2 position:
CGPointMake(-100, 0)];
CCSequence *sequence = [CCSequence actions:moveRight, moveLeft, nil];
CCRepeatForever *repeat = [CCRepeatForever actionWithAction:
sequence];
[cloudSprite runAction:repeat];
```

Libre à vous de manipuler les classes mises à disposition par Cocos2D, pour mener à bien les animations que vous souhaitez. Par exemple déplacer un objet jusqu'à un point précis sur l'écran, en lui faisant faire une rotation et en diminuant sa taille, vous pouvez manipuler les classes « `CCMoveTo` », « `CCRotateBy` » et « `CCScaleBy` ».

> Système de particules

Les éléments vus précédemment ne sont que quelques-unes des fonctionnalités basiques de Cocos2D, mais sachez qu'il existe aussi des fonctionnalités plus avancées permettant de mettre en œuvre des effets graphiques aboutis.

Dans de nombreux jeux, vous avez certainement remarqué des explosions, des flammes, des effets de fumée, etc. La mise en œuvre de telles fonctionnalités repose sur ce qu'on appelle les sys-

tèmes de particules. Cocos2D fournit en standard un tel système qui vous permet donc d'intégrer dans votre jeu des effets plus professionnels en quelques minutes seulement. Pour cela, de nombreux effets prédéfinis sont fournis avec le moteur et peuvent être ajoutés à vos jeux en seulement 2 lignes de code.

```
// Particules prédéfinies
// CCParticleExplosion, CCParticleFire, CCParticleFireworks,
CCParticleFlower, CCParticleGalaxy, CCParticleMeteor, CCParticleRain,
CCParticleSmoke, CCParticleSnow, CCParticleSpiral

CCParticleSun *sun = [CCParticleSun node];
[self addChild:sun];
```

Si vous souhaitez réaliser des effets personnalisés, vous devez tout d'abord commencer par regarder l'outil « **ParticleDesigner** » qui offre un éditeur graphique pour différents systèmes de particules, dont celui de Cocos2D. Cet outil est fourni en deux versions, l'une gratuite permettant de réaliser toutes les opérations nécessaires pour créer des particules, et l'autre payante possédant des fonctionnalités avancées [Fig.2].

Après avoir créé votre particule avec ParticleDesigner, il suffit de la sauvegarder au format Cocos2D, d'intégrer le fichier dans votre projet Xcode, puis d'ajouter le code suivant pour ajouter la particule à votre scène (comme fait précédemment pour les sprites).

```
CCParticleSystem *system = [CCParticleSystemQuad particleWith
File:@"CustomParticle.plist"];
[self addChild:system];
```

Vous pouvez également très bien vous passer de l'utilisation de ParticleDesigner pour réaliser vos particules et les écrire entièrement via du code mais au regard de la complexité et du grand nombre de paramètres entrant en ligne de compte pour arriver à l'effet désiré, ce n'est pas forcément la meilleure idée pour gagner du temps.

> Conclusion

Nous avons vu précédemment quelques-unes des fonctionnalités offertes par Cocos2D pour vous aider à créer vos jeux 2D en affichant des scènes, en ajoutant des éléments à celles-ci (labels et sprites), en animant les objets à l'écran et en ajoutant des effets permettant de simuler le feu, la neige, la pluie, les explosions...

Ce ne sont que quelques-unes des très nombreuses possibilités offertes par Cocos2D qui mériterait une série entière d'articles pour aborder tous les sujets, tellement les fonctionnalités offertes par ce moteur sont grandes. Nous n'avons par exemple pas abordé l'intégration des moteurs physiques (Box2D et Chipmunk), l'affichage de plusieurs centaines de sprites à l'écran à l'aide des « `SpriteBatchNode` », la gestion de la 3D isométrique, la gestion du son ou bien encore les extensions disponibles dans la communauté et notamment le projet Kobold2D qui ajoute et simplifie de nombreuses fonctionnalités à Cocos2D. Si vous souhaitez découvrir les possibilités de moteur plus en détail, n'hésitez pas à vous rendre sur le site de référence de la communauté Cocos2D qui contient de nombreux tutoriaux et exemples d'applications : <http://www.learn-cocos2d.com/>

Stéphane Cordonnier

Directeur - iTouch - <http://www.itouchconsulting.com>



Rassembler des passionnés des technologies Microsoft !

Entretien

Hubert de Charnacé, PDG de la SSII MCNEXT, leader Microsoft

➤ Pouvez-vous nous présenter MCNEXT ?

Nous avons créé MCNEXT en 2007. L'idée était de rassembler des passionnés des technologies Microsoft côté développement (.NET, SharePoint, BizTalk et la suite décisionnelle). Nous avons pris le risque très tôt de recruter des experts reconnus afin de garantir un encadrement technique de bon niveau tout autant pour nos clients que pour les futurs collaborateurs qui allaient rejoindre MCNEXT. Nous avons ainsi atteint le statut de partenaire Gold Microsoft dès la première année d'activité !

Notre partenariat avec Microsoft, renforcé par les nouvelles certifications Gold Partner sur nos 5 domaines d'expertises, nous permet ainsi de participer aux programmes de tests des nouvelles versions des produits avant leurs sorties (Office 365, Azure, SQL Server 2012, SharePoint 2010...).

5 ans après, MCNEXT regroupe plus de 140 collaborateurs, est présente à Paris, Lyon et Genève et maintient une progression de son chiffre d'affaires de 40 % par an sans avoir changé de feuille de route.

Elle accompagne une majorité des grands comptes français ainsi que des PME innovantes.

MCNEXT a mis en place pour les 3 prochaines années un programme de développement intensif dans lequel la compétence et l'innovation autour des solutions Microsoft sont au cœur de sa stratégie.

➤ Qu'est ce qui caractérise le plus MCNEXT ?

Le mouvement, le respect des promesses et l'innovation !

Pourquoi ?

- Une croissance constante, en plus de pérenniser l'entreprise, ouvre des possibilités d'évolutions pour tous.
- Le positionnement d'expert et de leader Microsoft est la meilleure garantie pour tous ceux qui nous rejoignent d'intervenir sur des projets utilisant la technologie qui les intéresse et de développer fortement leurs connaissances.
- Donner une place à l'innovation permet à tous ceux qui veulent s'investir sur une nouvelle version ou une nouvelle solution

d'en avoir les moyens et de travailler en équipe.

Nous travaillons à ce que ces trois enjeux soient au cœur du développement de l'entreprise. Chacun a la possibilité d'écrire des articles, d'animer des événements (plus de 2000 personnes sont venues à nos événements et nous intervenons chaque année aux Techdays), d'enregistrer des Webcasts et de participer aux grandes conférences Microsoft organisées aux Etats-Unis.

“ Pour continuer de grandir et de renforcer notre position d'experts Microsoft, MCNEXT recrute 60 collaborateurs ”

Nous organisons également des groupes de travail sous forme de labs. Ils étudient notamment Windows 8 (Metro), la table Surface V2 ou le maquettage de prototypes innovants en BI et SharePoint. Au travers de ces mini-projets internes, nos collaborateurs peuvent monter en compétences.

Cela débouche, pour Windows 8 Metro par exemple, sur des applications originales que nous avons proposées dans le nouveau « store » et par des projets chez des clients stratégiques ! Nous sommes ainsi prêts à

investir de l'énergie et des équipes pour développer toute bonne idée et la faire vivre dans le temps. Mais la cohésion et les moments de détente sont aussi essentiels.

➤ Comment accompagnez-vous les collaborateurs ?

Pour développer les compétences professionnelles de nos collaborateurs et préserver l'état d'esprit, nous fonctionnons sur un management de proximité organisé par les directeurs de pôles et non les commerciaux. Chaque collaborateur est considéré à part entière et ses perspectives d'évolutions sont définies avec le directeur de pôle. Nous ne parlons pas de plan de carrières mais d'évolutions individuelles au travers de formations ciblées, d'un parcours de certifications, de veille technologique partagée ou encore d'entraide au quotidien. Ce dernier point facilite la cohésion des équipes.

➤ Pourquoi choisir les technologies Microsoft ?

Les choix de spécialisations sont variés pour tous ceux qui se passionnent dans les technologies de l'information. L'intérêt de Microsoft pour les développeurs professionnels réside dans la variété des solutions et des sujets que .NET aborde (application native, web et mobile) et la qualité des outils de développement proposés. Windows 8 avec Metro est également une opportunité à saisir par tous.

➤ Allez-vous continuer à recruter en 2012 ?

Pour continuer de grandir et de renforcer notre position d'experts Microsoft, MCNEXT recrute 60 collaborateurs. Développeurs, consultants, architectes, formateurs. Contactez-nous et intégrez une société 100% Microsoft !



Quelques photos de nos soirées



MCNEXT
SKILLS FOR .NET

2, rue de Châteaudun – 75009 Paris
Tel : 01 49 70 81 33
Site : www.mcnex.com
Contactez Magali : drh@mcnex.com

Epitech : formation, open source, innovation et création d'entreprise



Après notre dossier du mois dernier sur l'emploi des développeurs open source, il nous est apparu opportun de revenir sur la formation de nos futurs ingénieurs avec le témoignage de Cyril Pierre de Geyer, Directeur des Etudes et de l'Innovation d'Epitech.

Un sondage réalisé pour Programmez dans les dernières offres d'emploi des entreprises qui parviennent à Epitech fait ressortir une forte attente sur les technologies open source et surtout le web. Plus d'une offre sur deux (55%) les mentionnent, et plus d'un tiers (35%) sont même full open source et exclusivement web (voir notre encadré).

« Aujourd'hui, quasiment tous les développements d'applications, en particulier la plupart des progiciels, se font en mode web », constate Cyril Pierre de Geyer, Directeur des Etudes et de l'Innovation d'Epitech, à qui nous avons posé quelques questions.

Programmez : Notre précédent dossier consacré au développeur open source fait apparaître une forte dissociation entre les langages et les CMS. Faites-vous le même constat ?

Cyril Pierre de Geyer : Effectivement, il y a quelques années les gens partaient directement sur des développements bas niveau, PHP ou ASP.NET, mais aujourd'hui nous avons tellement de technologies spécialisées qui sont arrivées au plus haut niveau, qui vont être à la fois CMS et framework, comme Drupal, Symfony ou Open Framework, que les entreprises font aujourd'hui leur demande directement sur ces technologies. Cela rend les choses également plus difficiles pour nous, car les écoles n'ont pas

pour vocation à former sur ces technologies et sur des produits aussi pointus. De plus, hier c'était Drupal 6 et Symfony 1, aujourd'hui c'est Drupal 7 et Symfony 2, à chaque fois la plateforme change ! Chez Epitech, nous apprenons à apprendre et à être très rapidement actif sur les technologies. Nous n'apprenons pas Symfony à nos étudiants, ils le font en stage, mais nous leur apprenons le C, le C++, tout ce qui leur permettra d'être rapidement productifs. Nous le constatons également au niveau technique. Certes il y a la bibliothèque, mais globalement soit vous développez en procédural, soit vous développez en objet. Nous donnons les bases. Après, ils devront apprendre le framework qu'ils vont utiliser eux-mêmes, généralement durant leur stage.

P ! : Il est apparu dans nos enquêtes que dans les entreprises le jeune recruté devra d'abord se former ?

C.P.G. : Les entreprises forment d'une part

MD considère qu'Epitech est le seul sourcing de compétences informatiques en Europe et représente l'excellence française en terme de formation, reconnue au niveau international.

“ 70% des étudiants arrivent en voulant faire du jeu vidéo. ”

à la technologie, d'autre part à leurs méthodes. D'une entreprise à l'autre les méthodes diffèrent. Les recrutés sont donc d'abord formés à ces méthodes, et également, s'ils ne connaissent pas le framework utilisé. L'évolution des technologies est tellement rapide qu'on ne peut former sur les dernières versions. Je me souviens, il y a quelques années, les annonces demandaient 10 ans d'expérience sur PHP alors que le langage venait juste d'apparaître... C'est le cas pour certains langages. Ils n'apparaissent pas dans notre sondage, mais si l'on retire le web qui représente 35% des demandes, mécaniquement tout le monde va remonter. On va alors retrouver du Java, du JEE, du Ruby on Rail, de l'infra, de l'application 3D, du développement iPhone et iOS, du technico-commercial, de l'ERP... Prenons l'exemple de Java, qui représente encore 10% des demandes. La demande de développeurs Java reste ponctuelle et liée au web. Ce langage a de l'existing, il est un peu complexe, et ce n'est pas la meilleure technologie pour faire du web.

J'aurais tendance à dire que la demande Java concerne plus de la maintenance de l'existant que du développement. De même pour les formations sur le mainframe et Cobol. Pour nous, la demande sur ces produits est négligeable. De la même façon, nous ne formons pas, la formation se fera directement dans l'entreprise.

P ! : Comment mettez-vous en adéquation la formation et la demande des entre-

Que recherchent les entreprises qui recrutent chez Epitech ?

Cyril Pierre de Geyer s'est livré pour nous à un rapide sondage sur les offres d'emploi qui parviennent à Epitech. L'école cumule 10 000 offres d'emploi et en reçoit quotidiennement 15 nouvelles. Les résultats du sondage indiquent une tendance.

- 55% des offres mentionnent les technologies open source, dont le web
- 35% des offres sont des demandes full open source exclusivement liées au web (PHP, Symfony, MySQL, etc.)
- 20% des offres mentionnent des technologies Microsoft
- 15% des offres sont liées au mobile
- 10% des offres concernent Java et JEE

prises, avec le décalage lié à la durée de la formation ?

C.P.G. : Nos formations sont extrêmement souples et fonctionnent à la demande. Nous proposons à nos étudiants un large catalogue de cours, dans lequel ils piochent. Nous avons aujourd'hui des promos de 1000 étudiants, et nous allons vers 5000 à terme. Au début, 70% d'entre eux arrivent en voulant faire du jeu vidéo. Une partie de notre travail est de changer leur orientation et de leur permettre de se créer un métier lié à l'informatique. Au final, il y en assez peu qui font du jeu vidéo. La première année est consacrée aux bases, à l'apprentissage du C et de Linux, par exemple. Nous leur apprenons à mettre les mains dans le cambouis et à rentrer dans le détail du code. Ensuite, ils vont faire du C++, puis ils vont se spécialiser en fonction de leurs aspirations et des stages qu'ils vont faire. Par ailleurs, quasiment 100% de nos étudiants ont une entreprise avant de terminer leur cursus. Quant à

“ Chef de projet, c'est le fantasme des étudiants. ”

ceux qui n'en ont pas, c'est qu'ils ont un projet. Quand une entreprise appelle mon service et demande à ce que nous lui conseillions quelqu'un, nous l'informons qu'en dernière année tous nos étudiants ont du boulot ! C'est trop tard, nos étudiants ont déjà du travail, ont déjà été mis en contact avec des sociétés qui recrutent, et travaillent avec elles. C'est pourquoi les entreprises qui recrutent à long terme se présentent aux étudiants dès la première année. Nos étudiants rencontrent ces entreprises, assistent à des conférences, vont y faire des stages. Sur le secteur informatique, les entreprises viennent courtiser nos étudiants de manière intensive. L'effet pervers sur ce point est que nos étudiants ne se rendent pas compte à quel point ils sont favorisés. Nous essayons de travailler sur ce plan afin d'éviter qu'ils soient hautains et les rendre plus modestes en entreprises.

P ! : A ce propos, il semble que la majorité des étudiants qui suivent un cursus de développeur, en réalité cherchent à accéder à des postes d'encadrement et de chef de projet...

C.P.G. : En effet, c'est le fantasme des étudiants qui affirment qu'ils veulent être chef

“ Quasiment 100% de nos étudiants ont une entreprise avant de terminer leur cursus... Ceux qui n'en ont pas, c'est qu'ils ont un projet. ”

de projet. Ils cèdent à une aura qui est liée à la fonction. Mais ils le disent sans avoir conscience de ce que c'est ! Ils doivent supposer que ce n'est pas eux qui font le travail, qu'ils vont au restaurant, qu'ils donnent les ordres et qu'ils se tournent les pouces ! En cinquième année, après de nombreux stages, et comme ils travaillent trois jours par semaine en entreprise, ils se rendent compte de ce qu'est le métier de chef de projet. La chance que nous avons à Epitech, c'est que nous mettons beaucoup de nos étudiants en entreprise. Lorsqu'ils voient ce qu'est un chef de projet - une personne qui prépare et organise - s'ils sont compétents, ils peuvent accéder à ce poste de par leur cursus Bac +5 et en faisant leur preuves. Mais beaucoup se rendent compte que ce qu'ils apprécient, c'est la technique. Alors ils sortent du fantasme...

P ! : Qu'en est-il des développeurs dans l'industrie et la mobilité ?

C.P.G. : Nous avons cette demande, autour en particulier de la mobilité et d'Android, mais elle est mise en retrait du fait de fortes demandes sur le web. Il est intéressant du point de vue de l'employabilité de constater qu'environ 2% des demandes sont orientées sur Google ! S'il y a beaucoup de demandes sur Android, sur Google, ses technos et ses API, en termes d'emploi, il n'y a finalement que peu de demandes.

P ! : Quel est le profil de vos étudiants et que leurs conseillez-vous ?

C.P.G. : Nos profils d'étudiants sont vraiment larges, hormis peut-être le geek, passionné de jeux vidéo et qui entre pour cela dans notre cursus. Le profil de l'étudiant à l'entrée, c'est le jeune qui aime le jeu vidéo et potentiellement les mangas, et qui aime bidouiller en informatique. En fait, 70% de nos étudiants viennent

par le jeu vidéo et le manga... Pour se préparer à entrer à Epitech, il faut être curieux, regarder ce qu'il se fait et manipuler son ordinateur. Peut-être pas pour avoir de la théorie mais plutôt de la pratique. C'est notre méthode à Epitech, permettre à nos

étudiants de toucher, et ensuite leur expliquer ce qu'ils ont touché. C'est différent de la plupart des cursus, qui font d'abord de la théorie et ensuite mettent en pratique. Comme nous avons induit nos étudiants vers ce vers quoi ils doivent aller, finalement quasiment tout le monde va dans cette direction. Mais là, nous nous sommes rendu compte qu'il y a une perte d'innovation. Or l'innovation est très importante pour nous. C'est pourquoi nous avons une technologie qui va avec et qui permet à nos étudiants d'y toucher.

P ! : Et qu'en est-il durant le cursus ?

Cyril Pierre de Geyer : Ce qui différencie nos étudiants qui entrent sur le marché du travail, c'est leur expérience professionnelle et associative. Lorsqu'au début on cherche un job, on n'a pas d'expérience professionnelle. L'étudiant peut en revanche jouer sur sa participation à des communautés de développeurs iOS, PHP, etc. C'est pourquoi nous les incitons à participer à la vie associative et à des laboratoires de recherche. En entrant dans ces laboratoires, l'étudiant participe à des projets communautaires et apprend. C'est là que l'open source prend toute sa valeur pour nous, l'étudiant apprend par ses pairs. Il aura accès à des gens pointus qui vont le conseiller. Mais ce n'est pas dans leur culture par défaut, et ils ont tendance à ne pas oser, même s'ils sont très bons.

P ! : Une autre particularité liée à Epitech, c'est la création d'entreprise...

C.P.G. : En effet, 16% des jeunes qui sortent du cursus créent une entreprise. C'est 7 fois plus que la moyenne des écoles d'ingénieurs. C'est notre côté innovation et entrepreneuriat qui ressort fortement. C'est lié à deux points : le fait que nous allons vers l'innovation, et le fait que nous faisons travailler nos étudiants au cours des

“ 16% des jeunes qui sortent de notre cursus créent une entreprise. ”

deux dernières années sur un projet, l'EIP, Epitech Innovation Project, qui se veut innovant. Ces projets s'exécutent par groupes de quatre étudiants et parfois aboutissent à la création d'une entreprise. Depuis cette année, nous travaillons également avec

0% de turn-over chez Zenika

Les recettes d'un leader technique qui mise avant tout sur les hommes

Crée il y a 6 ans par quatre consultants expérimentés et engagés, organisatrice en 2011 de la What's Next, première conférence internationale Java ayant rassemblé sur Paris plus de 700 personnes, cette société experte dans les technologies Java/JEE et les méthodes agiles poursuit une seule ambition : être leader sur son marché.

Un rêve ? Pas si sûr... L'objectif affiché de Zenika n'est pas le nombre de salariés, ni le volume de contrats ; beaucoup plus simplement, c'est d'offrir à ses clients une expertise technique sur mesure, grâce à l'engagement de consultants qui s'investissent à la fois humainement et professionnellement, et apportent leur savoir-faire, mais aussi leur personnalité, à chacun de leurs projets. C'est peut-être la raison pour laquelle Zenika, avec 75 salariés, affiche aujourd'hui 0% de turn-over, un taux record à faire pâlir d'envie les grandes structures aux effectifs plus nombreux, mais également plus impersonnelles. Le secret ? Quelques principes simples, des réalisations concrètes et une culture d'entreprise axée sur la valorisation individuelle et l'égalité.

Les principes

Un savant mélange entre Conseil, Formation et Réalisation. S'inspirant de leur propre expérience, les dirigeants et fondateurs ont imaginé ce que serait pour eux le métier de consultant idéal. Et ils l'ont fait, proposant à leurs salariés non pas de beaux discours, mais tout simplement ce qu'eux-mêmes auraient rêvé pour eux :

- un environnement dynamique et convivial, résolument tourné vers l'avenir ;
- l'accent mis sur le partage des connaissances et les échanges de savoir-faire et d'expérience ;
- une gestion de carrière personnalisée sur la durée à travers un suivi régulier et un dialogue permanent ;
- un modèle de rémunération motivant intégrant, pour tous, une part variable.

Les réalisations

Après un bilan de compétences personnalisé, les consultants inter-



L'équipe Zenika à la What's Next en mai 2011, première conférence internationale Java ayant rassemblé sur Paris plus de 700 personnes.

viennent dans des environnements variés et innovants ; ils peuvent également animer en parallèle des formations dans leur domaine d'expertise. Chaque consultant a ainsi la possibilité de moduler les 3 métiers selon ses propres affinités.

La culture d'entreprise

Le consultant Zenika se doit d'être autant un expert technique qu'un bon communicant. Loin de s'isoler dans son domaine d'expertise, il est en permanence invité à confronter ses idées et à améliorer ses connaissances. Jeune diplômé, consultant expérimenté, tous trouvent chez Zenika l'opportunité de partager leur passion. Comment ? En participant chaque mois à des réunions techniques, en accédant à l'intégralité du catalogue de formation et en contribuant activement à l'animation de la communauté (réécriture de blogs, participation à des conférences). Et pour être toujours au plus proche de ses salariés autant que de ses clients, Zenika s'exporte et accompagne ceux qui souhaitent s'installer en dehors de la capitale. Aujourd'hui, l'entreprise est présente à Paris, Rennes, Nantes, Lyon, Londres et Athènes ; et regarde déjà vers d'autres horizons.

Avec 400% de croissance en 5 ans et un classement dans le Deloitte Fast Technology 500 EMEA, Zenika s'impose résolument comme un leader technique au modèle et à la culture uniques sur son marché qui lui permettent aujourd'hui d'afficher, et c'est bien une réalité, 100% de consultants satisfaits.

Témoignage

Une relation directe et amicale

J'ai rejoint Zenika en septembre 2008, à la fin de mes études ; j'étais alors le 7e salarié de l'entreprise.

Dès le début, une relation directe et amicale avec les gérants s'est instaurée. Aujourd'hui avec 75 salariés, il n'y a toujours qu'un seul niveau de hiérarchie ; les dirigeants trouvent toujours une place dans leur agenda pour les consultants, ce qui est très appréciable.

Le tryptique « réalisation / conseil / formation » me permet d'animer régulièrement des formations ; cela fait partie du 'deal' Zenika et personnellement, j'adore ça ! La transmission de connaissance est essentielle selon moi. L'expérience 'terrain' fait que les formations sont pleines de sens et reflètent une véritable expertise mise à l'épreuve chez nos clients.

Avec la croissance de Zenika, il y a toujours de nouvelles personnes avec qui échanger, sur des sujets toujours plus percutants et intéressants. Il y a aussi des challenges d'organisation à affronter pour conserver cette unité qui, je le pense, fait la force de Zenika. »

Vincent - Consultant Zenika Paris



51, rue le Peletier - 75009 Paris

info@zenika.com

01 45 26 19 15

<http://www.zenika.com>



HEC, toujours dans cette idée de création d'entreprise. Une bonne entreprise qui lève du financement se base sur deux axes : il faut une idée, et il faut une équipe. Cette dernière doit être composée d'un bon technicien, sortant d'Epitech, un bon 'business guy', potentiellement sortant de HEC, et de cheveux gris qui ont de l'expérience. Nous créons ces équipes afin de permettre à ces 16% qui créent leur boîte d'aller plus loin, des PME aux groupes internationaux. C'est pourquoi nous avons des partenariats avec HEC et leur incubateur, qui ont également besoin de productifs. Lorsqu'un étudiant arrive en fin de cursus, il a déjà du travail, donc il est très difficile de monter des projets. En les faisant entrer en relation, nous créons des liens. Nous essayons de les aider à innover et de les placer dans un terreau pour aller de l'avant.

P ! : Peut-on encore créer en France des entreprises technologiques ?

C.P.G. : Le rêve américain est très présent chez les étudiants, mais je trouve qu'en France il est extrêmement facile de créer une entreprise. Un chômeur peut créer et continuer de toucher le chômage, faisant de l'Etat le premier Business Angel, même si derrière on se fait un peu assommer en termes de charges. Je pense qu'en France, dans la technologie, il y a de quoi faire. Et nous sommes très forts en informatique !

Portrait de Cyril Pierre de Geyer

Notre témoin est un fin connaisseur du monde des IT, du développement et de l'open source, ainsi que des start-ups françaises. Directeur des Etudes et de l'Innovation d'Epitech, il est également Directeur de l'innovation chez Comuto, éditeur du site covoiturage.fr, ainsi que CEO fondateur de Agoratic, qui crée des évènements IT. Précédemment, il a été le responsable du centre de formation IT d'Alter Way Group, le fondateur de la société de formation Anaska, et le co-fondateur de l'agence Kaptive. Il a également écrit plusieurs ouvrages, plus particulièrement autour du langage PHP.

Yves Grandmontagne

Tendance emploi

CES ENTREPRISES QUI RECRUTENT

Programmez a repéré pour ses lecteurs plus de **6 970 postes de Développeurs, Ingénieurs, Chefs de projets, Consultants, Support, etc., à pourvoir sur toute la France et en Europe, en 2012 et parfois jusqu'en 2015.**

● Développeurs et consultants

(30 postes)

SensioLabs, éditeur du CMS open source Symfony, pour accompagner sa croissance, notamment sur ses activités de services professionnels autour de Symfony, recherche plus de 30 profils, majoritairement des développeurs expérimentés et des consultants techniques. sensiolabs.com

● Ingénieurs (3 900 postes)

Alten, ingénierie et conseil en technologies, prévoit en 2012 de recruter 3900 ingénieurs, dont 2600 en France. Les postes proposés dans les métiers de l'ingénierie sont d'une rare diversité : électronique, informatique embarquée, AMOA et PMO, conception mécanique, méthode et process, qualité, logistique, systèmes d'information, développement SI, architecture SI, télécom, etc. www.alten.fr

● Ingénieurs développement et administrateurs systèmes

(500 postes)

L'hébergeur OVH recrute sur 3 ans 500 collaborateurs, principalement des ingénieurs développement et administrateurs systèmes, ainsi que des commerciaux et des techniciens itinérants, pour ses 'OVH city' de Strasbourg et Paris, ainsi que ses nouvelles implantations de Lyon, Brest, Sophia Antipolis et Gand (Belgique). www.ovh.jobs

Experts métiers (50 postes)

T-Systems, division grands comptes du groupe Deutsche Telekom, recrute pour Toulouse une cinquantaine de collaborateurs experts métiers de l'entreprise (PLM, SAP, ECM NTIC) afin de renforcer les effectifs de l'Aerospace Center toulousain et mener à bien une multitude de pro-

jets sur le plan international, en particulier dans le cadre d'un contrat d'infogérance et d'help desk avec Airbus. www.t-systems.fr

● Ingénieurs, informaticiens (1200 postes)

La SSII d'ingénierie et de conseil en technologies SII recrute 1200 informaticiens, dont 650 en France, jusqu'en mars 2013. Sont recherchés des consultants IT, des ingénieurs d'étude et développement, et des chefs de projets dans les technologies de l'information et de la communication, les logiciels et l'électronique embarqués, l'ingénierie systèmes et mécanique, l'informatique de gestion et l'infrastructure systèmes et réseaux. Les postes à pourvoir concernent à 80% de profils expérimentés, ainsi qu'une dizaine de postes destinés à des informaticiens en situation de handicap. www.groupe-sii.com/fr

● Ingénieurs, support, vente et marketing (750 postes)

Le géant du CRM dans le cloud Salesforce.com recrute en Europe et sur deux ans des collaborateurs qualifiés avec de l'expérience dans la vente, le marketing, l'IT et le support. Salesforce.com

● Ingénieurs (400 postes)

Le Groupe ESR, spécialiste de la gestion des infrastructures informatiques et télécoms, recrute en 2012 et pour la France des profils d'ingénieurs systèmes, sécurité, production, télécom etc., jeunes diplômés et experts. www.esrt.fr

● Ingénieurs et développeurs (140 postes)

Porté notamment par ses activités de services, le Groupe Hardis recrute en 2012 des profils technico-fonctionnels, assistance technique sur site client, développement au forfait, usine de développement à distance, développement agile en méthode Scrum, des business analysts avec une forte composante métier, développeurs Java, .Net, PHP, des architectes Java et .Net, des chefs de projet, des ingénieurs d'études, des Scrum masters, des coaches agiles, ainsi que pour son entité Hardis Conseil des consultants AMOA, qualification logicielle et système d'information. Les postes sont à pourvoir à Paris, Lyon, Grenoble et Nantes. www.hardis.fr

L'ALTERNANCE, UNE OPPORTUNITÉ POUR LES JEUNES FUTURS INFORMATICIENS

Selon le ministère du Travail, 627 000 contrats d'apprentissage et de professionnalisation ont été signés en France en 2011. L'alternance progresse, de 8,3% de 2010 à 2011, il n'est que temps ! En effet, l'alternance est une opportunité pour les jeunes (16-25 ans) et favorise l'insertion professionnelle. Le taux d'embauche en fin de contrat de professionnalisation ou d'apprentissage représente 70% en moyenne. Et même quasiment 100% dans les métiers de l'informatique. Opportunité pour l'entreprise également, pour former des jeunes à ses méthodes et à ses outils. L'apprentissage est enfin une des voies privilégiées pour faire face au renouvellement des sala-

ries avec les départs en retraite. Les offres en alternance vont du certificat d'aptitude professionnelle (CAP) au bac +5. Concrètement, un contrat d'apprentissage peut donc concerner jusqu'aux formations d'ingénieur. Les grandes entreprises commencent à le comprendre (jusqu'à présent l'alternance a surtout été pratiquée par les TPE et les PME). Elles sont même de plus en plus incitées à pratiquer l'alternance, le quota d'alternants dans les entreprises de plus de 250 salariés a ainsi été relevé de 4 à 5%, avec en objectif de porter en 2015 le nombre de contrats à 800 000. L'alternance est une opportunité pour les métiers de l'informatique, pensons-y !

Prenez une longueur d'avance : créez une application Windows 8 avec Visual Studio & TFS 2012

Partie 4 : La revue de code

Après avoir écrit nos premières lignes de code sous Visual Studio 2012, il est maintenant temps de s'intéresser un peu plus à un outil complémentaire qui s'avère quasi indispensable pour le design de nos applications Windows 8 : Blend for Visual Studio. Si le premier article de cette série, consacré au Storyboarding, traitait de l'interface de l'application dans son ensemble, l'outil Blend va, quant à lui, permettre de modifier et intégrer très facilement le design de l'application (en CSS dans le cadre de cet article).

Avant même de commencer à se replonger dans le code de l'application, il serait dommage de ne pas nous attarder sur une nouvelle fonctionnalité phare du duo Visual Studio 2012/Team Foundation Server 2012 : Le Code Review. L'intégration complète de ce scénario dans ces nouveaux outils est l'occasion idéale pour mettre en exergue ses avantages et se pencher sur la manière de le mettre en œuvre.

> Le Code Review

Le **scénario de revue de code** est quelque chose d'assez simple à implémenter, il est d'ailleurs parfois utilisé sans forcément être qualifié de scénario de « Revue de code » à proprement parler : « Demander à l'un de ses collaborateurs de relire ou vérifier un morceau de code écrit » est une forme simple de revue de code.

Il était déjà possible dans la version 2010 de Team Foundation Server d'implémenter ce type de scénario (pour cela les Shelvesets étaient utilisés). Avec l'arrivée de la nouvelle version de Team Foundation Server, ce scénario est dorénavant mieux outillé et dispose d'un *Workflow*, sous la forme d'un Work Item nommé **Code Review**, disponible nativement.

Utilité de la revue de code

La revue de code est une étape qui est très souvent sous-estimée, voire oubliée, mais qui apporte pourtant de nombreux avantages :

La première qualité qu'on lui reconnaît est sa capacité à détecter des bugs plus tôt dans le cycle de vie de l'application. Plus un bug est détecté tôt et plus il est considéré comme « facile » à corriger. D'ailleurs les méthodologies Agiles s'appuient beaucoup sur ce principe, et pour cela mettent en place un cycle de développement court pour pouvoir exécuter des tests et remonter des bugs très tôt et régulièrement.

Un autre aspect important est sa capacité à homogénéiser une équipe. En effet, dans le cas où des personnes n'ayant pas la connaissance du projet sont amenées à y participer, la revue de code peut alors aider à l'apprentissage de l'aspect aussi bien technique que fonctionnel du projet. Elle facilite la récupération de « conseils » ou de « validations » d'autres collaborateurs plus expérimentés sur le sujet. Dans le cas par exemple d'un stagiaire qui participe à un ajout de fonctionnalités sur un projet : Avant de faire un check in, son maître de stage peut faire une revue de son code et le conseiller sur de meilleures méthodes à utiliser. Autre exemple, si cette fois-ci le relecteur du code est moins expérimenté, il pourra profiter de chaque relecture pour apprendre de nouvelles techniques de programmation, voire éventuellement susciter chez lui l'envie de rechercher comment fonctionnent certaines parties de code qu'il ne comprend pas (**Note** : l'impact sur la montée en compétence est cependant moins important qu'il ne pourrait l'être avec la pratique du « pair programming »)

The screenshot shows the 'My Work' section of Team Explorer. It displays a list of work items under 'In Progress Work Items & Changes', showing 1 edit(s) and a link to 'View Changes'. Below this is a section for 'Suspended & Shelved Work', with a note that you can suspend changes. Under 'Available Work Items', it says 'No work items found.' At the bottom, there's a section for 'Code Reviews & Requests (1)' with an incoming request from 'Philippe' for a '7 - Ajout Division' item, which is marked as 'Finished (looks good)'.

The screenshot shows the 'Code Review' section of Team Explorer. It displays a message: 'You accepted this code review requested by vfabing.' Below this, there are buttons for 'Send Comments', 'Send & Finish', and 'View Changeset'. A dropdown menu is open, showing options: 'Looks Good', 'With Comments', and 'Needs Work'.

La revue de code permet également de tendre plus vers un « effort d'équipe » que vers un « effort individuel ». Cela passe dans un premier temps par le fait que les membres du projet d'équipe sont beaucoup plus au courant des modifications effectuées par les autres participants. Le simple fait de relire le code, même sans relever de modifications, permet aux relecteurs de prendre connaissance des nouvelles fonctionnalités ou des corrections apportées par ses collaborateurs. Il est également possible d'instaurer certaines règles au niveau de cette revue de code : par exemple pour qualifier une exigence comme terminée (c.f. : *article précédent*). On peut exiger un certain nombre de revues de code avant de considérer qu'un lot de modifications possède une qualité suffisante : relu par 4 membres de l'équipe minimum, toute l'équipe, etc.

À noter également que la revue de code se marie particulièrement bien avec les tests unitaires car la qualité d'un test unitaire garantit généralement aussi la qualité du code qu'il teste. Le contexte relativement indépendant du test unitaire permet de s'affranchir de la complexité que peut apporter la revue d'un morceau de code pris dans son contexte métier. De ce fait, on pourra faire des revues de code sur les tests unitaires plutôt que directement sur le code testé. (En vérifiant leur pertinence, etc.).

Présentation du Work Item de Code Review avec Team Foundation Server

Dans Team Foundation Server 2012, la revue de code est outillée avec deux types de Work Item appelés **Code Review Request** et **Code Review Response**. La *Code Review Request* contient le workflow global de la demande de revue de code et sera manipulée par le demandeur. À partir de cette demande vont être générés autant de Work Items de *Code Review Responses* qu'il y a de relecteurs. Ce sont ces derniers qui contrôlent le workflow du Work Item. La *Code Review Response* passe à l'état fermé dès que la revue de code est terminée. On distingue deux cas de figure pour les demandes de revue de code :

- partir des modifications en attente
- partir d'un changeset (lot de modification) déjà archivé

Le premier cas de figure convient bien mieux à une petite équipe et permet d'assurer une meilleure qualité du code archivé. La nouvelle demande de revue de code déclenche la création d'un Shelveset qui met le code en attente jusqu'à sa validation.

Le deuxième cas de figure, quant à lui, permet d'implémenter un scénario de revue de code plus flexible, sans contraindre les développeurs à attendre la relecture de leur code avant de l'archiver.

L'avantage de ces nouveaux Work Items est qu'ils sont parfaitement intégrés à Visual Studio 2012. En effet, l'une des nouveautés phare du nouveau Team Explorer 2012 est qu'il centralise désormais tout

le travail du développeur dans une section appelée **My Work** où se retrouvent également ces Work Items.

Les différentes étapes d'une Code Review

La première étape consiste à créer une nouvelle demande de revue de code sous forme d'un Work Item de *Code Request Review*. Cette opération se fait de manière totalement transparente à partir de la section **Modifications en attente** du Team Explorer, ou à partir de l'historique de l'explorateur de contrôle de code source [Fig.1].

Une fois les détails de sa demande de revue de code saisis, pour chaque relecteur précisé dans la demande sont créés des Work Item de type *Code Review Response* qui sont automatiquement associé au Work Item *Code Review Request*. Les relecteurs de leur côté n'ont plus qu'à visualiser le Work Item dans leur section *My Work*. Ils ont alors accès au code source qui a besoin d'être revu. Il est également possible de notifier l'initiateur de la demande en **Acceptant** ou en **Refusant** la demande de revue de code.

La revue de code avec Visual Studio 2012 est très simple, on dispose de la liste des fichiers à relire, et à côté de chaque fichier se trouve une case à cocher qui permet de confirmer la relecture comme effectuée.

La relecture des fichiers s'effectue au travers d'un système de dialogue basé sur des commentaires. Il est possible de créer des commentaires à plusieurs niveaux : Au niveau de la ou des ligne(s) de code, au niveau d'un fichier source, ou même au niveau de l'ensemble des modifications. Les commentaires étant asynchrones, il

The screenshot shows the 'Code Review' window in Team Explorer. The main title bar says 'Team Explorer - Code Review'. The window is titled 'Code Review | TEST'. It shows the following details:

- Send Comments** button
- Actions** dropdown with 'Accept' and 'Decline' options
- Reviewers** section: 1 item, 'Requested by vfabing'.
- Related Work Items** section: 1 item, '2 - Implémentation Division'.
- Comments** section: 3 items, including a new comment from 'vfabing' with the text 'Fait bien attention à la division par 0'.
- Files** section: A list of files with checkboxes:
 - \$/TEST/Main/Calculatrice (checked)
 - Form1.cs (checked)
 - Form1.Designer.cs (checked)
 - Form1.resx [add] (checked)

The screenshot shows the 'Code Review' window in Team Explorer. The main title bar says 'Team Explorer - Code Review'. The window is titled 'Code Review | TEST'. It shows the following details:

- Send Comments** and **View Changeset** buttons
- Actions** dropdown with 'Complete' and 'Abandon' options
- Reviewers** section: 1 item, 'Requested by vfabing'.
- Comments** section: 1 item, a reply from 'vfabing' with the text 'Fait bien attention à la division par 0'.



suffit alors de cliquer sur le bouton **Send Comments** pour immédiatement notifier son interlocuteur de ses nouveaux commentaires, et ainsi lui permettre de répondre.

Une fois la relecture terminée, le relecteur peut alors fermer la *Code Review Response* en spécifiant son appréciation globale : « Looks Good », « With Comments » ou « Not Good ».

Finalement, l'initiateur de la demande de revue de code peut consulter l'état de toutes les relectures en cours associées à sa *Code Review Request*, directement à partir de son espace *My Work*. Une fois tous les aller-retour avec les relecteurs terminés, il suffit ensuite de clôturer la *Code Review Request* en la qualifiant de « terminée » (Complete) ou « abandonnée » (Abandon).

Design de l'application

Le fait de décider de faire une application Windows 8 basée sur le duo HTML5 et Javascript implique que l'intégration du design de notre application passe par l'utilisation de styles CSS. S'il est bien sûr possible d'édition nos feuilles de style directement depuis Visual Studio 2012 en profitant de l'IntelliSense, il est possible dans cette nouvelle version d'utiliser également tous les outils proposés par Blend 5 pour l'intégration de notre design.

Blend 5 for Visual Studio

Blend est l'outil de création d'interfaces graphiques de la suite Expression de Microsoft. S'il était jusque-là exclusivement destiné à la création d'interface utilisant le langage XAML de Silverlight, l'arrivée de Windows 8 et le support des technologies HTML5 et CSS3 pour les applications natives ont changé la donne.

Ainsi à la création d'un nouveau projet, le choix est laissé à l'utilisateur entre un projet de type XAML ou HTML. Les Template d'application proposés pour faciliter le démarrage du projet sont les mêmes et les outils proposés par Blend offrent le même fonctionnement pour les 2 langages.

Blend est avant tout un outil destiné à la création et à l'intégration du design de vos applications et il est donc inutile d'essayer de l'utiliser pour développer du JavaScript car aucun des outils de Visual Studio n'est présent pour ce langage.

En revanche, l'interface est directement pensée pour les designers et leur offre notamment un ensemble d'outils permettant d'édition le code et les styles des pages de manière beaucoup plus graphique que dans Visual Studio. Par défaut, l'interface présente les outils répartis dans 5 zones : [Fig.2].

• **Partie 1** : La partie centrale, et la plus étendue également, est dédiée à l'affichage du rendu en direct des pages de l'application.

Cette vue permet en plus de voir les effets de nos modifications en direct, de sélectionner et modifier nos éléments du DOM. Elle offre la possibilité de basculer d'un mode d'édition à un mode interactif. Ce dernier permet de manipuler l'application et de la mettre en « pause » à tout moment afin de pouvoir, par exemple, styler facilement un élément dynamique tel qu'un message qui n'apparaîtrait qu'au clic sur un élément.

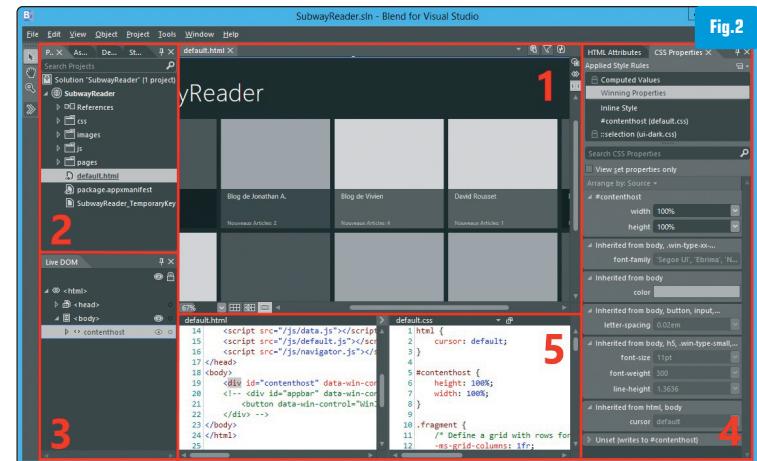
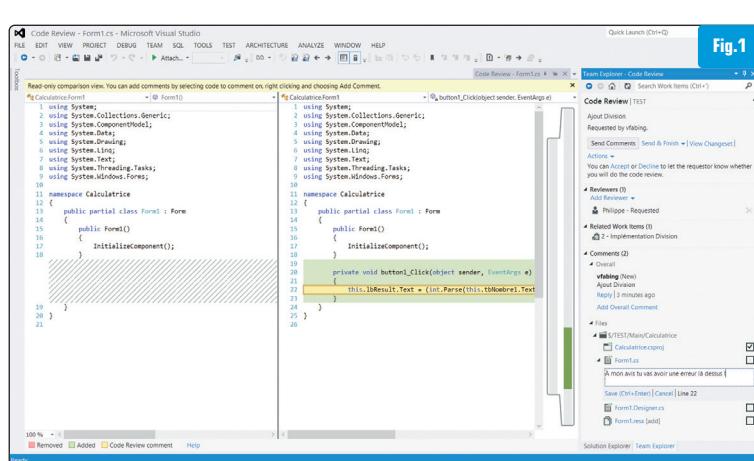
- **Partie 2** : Situé en haut à gauche de l'écran se trouvent les explorateurs permettant de parcourir les fichiers du projet, les éléments et Assets (composants) disponibles pour une utilisation dans un projet HTML5 METRO.
- **Partie 3** : L'explorateur du DOM permet de parcourir la structure de votre page de façon hiérarchique.
- **Partie 4** : Le volet de droite permet d'édition toutes les propriétés HTML et CSS de l'élément sélectionné et facilite grandement la saisie par des champs adaptés au type de la propriété (valeurs clés proposées dans une liste déroulante, sliders pour les valeurs numériques, etc.) Le debugging des styles CSS est également grandement facilité par la sélection des règles d'applications des styles classés par ordre d'application. Ainsi il est possible de savoir si un style est appliquée au niveau inline (le plus bas), calculé dynamiquement (par JavaScript) ou bien encore par quelle feuille de style, voire sélecteur CSS : type, classe ou identifiant.
- **Partie 5** : Le dernier volet affiche le code source de la page ainsi que la feuille de style associée.

Tous ces volets sont liés et le fait de sélectionner un élément du code dans un de ces volets va avoir une influence sur tous les autres. Par exemple le fait de sélectionner un élément DIV dans le Dom Explorer, place le curseur sur cet élément dans le volet de code, et affiche ses propriétés HTML ou CSS dans le volet de droite. La modification du code ou d'une propriété CSS, quant à elle, est instantanément répercutée sur le rendu affiché au centre et c'est bien là une des grandes forces de l'outil qui fait gagner un temps précieux dans la phase d'intégration.

Modification du style de notre application

L'application créée de base propose un Template de style nous permettant d'afficher les informations de manière assez agréable certes, mais cela manque de couleur et de design.

En se penchant sur les styles mis en place on remarque plusieurs types de sélecteurs CSS. Certains styles sont définis avec un sélecteur sur le type de la balise HTML :



```
html { cursor: default; }
```

On retrouve également les sélecteurs basés sur la propriété class des balises html et dont le nom de la classe est préfixé par un point :

```
.itemslist { height: 100%; : relative; width: 100%; z-index: 0; }
```

Il est également possible d'appliquer les styles pour un élément en particulier en ciblant ce dernier grâce à la propriété id de l'élément HTML, et ceci en préfixant l'id d'un caractère # dans la feuille de style :

```
#contenthost { height: 100%; width: 100%; }
```

A la fin de la feuille de style se retrouvent également les media-queries qui permettent de définir des propriétés conditionnelles qui ne s'appliquent que dans les cas qui valident la condition définie. Par exemple, pour définir le style d'une page lorsqu'elle est affichée en mode Snap (à savoir le mode de partage de l'espace de travail avec une autre application). C'est aussi grâce aux media-queries que l'on peut définir le style de l'application lorsque l'utilisateur passe en mode portrait. Les media-queries permettant de redéfinir les propriétés CSS de ces modes sont les suivantes :

```
@media screen and (-ms-view-state: snapped) {  
    /* définition de mes styles CSS pour la vue Snap */  
}  
  
@media screen and (-ms-view-state: fullscreen-portrait) {  
    /* définition de mes styles CSS pour la vue en mode portrait */  
}
```

Il suffit simplement d'ajouter quelques propriétés CSS modifiant la couleur de fond et l'espacement des éléments de notre ListView dans la feuille de style default.css, commune à toutes les pages de l'application, de cette façon :

```
body {  
    background-color: #288cff;  
}  
  
.win-listview > .win-horizontal .win-container {  
    margin-top: 30px;  
    margin-right: 30px;  
}
```

On obtient immédiatement un résultat plus agréable : [Fig.3 et 4].

Ajout des tuiles et du Splash Screen

Avec l'arrivée de Windows 8 et de l'interface Metro, les icônes actuelles prennent un petit coup de vieux et une application ne peut

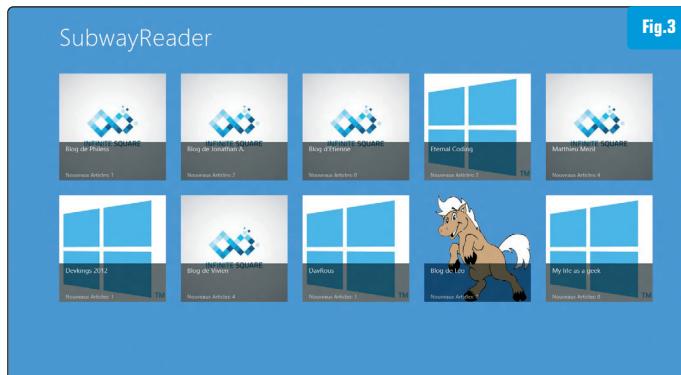


Fig.3

plus se passer de bonnes ressources graphiques. Cela ferait mauvais effet si un utilisateur qui décide d'installer une application se retrouve avec la tuile grise générique dans son menu démarrer et il en est de même pour le Splash Screen qui s'affiche au démarrage de l'application.

Ces ressources sont définies dans le manifeste d'application : package.appxmanifest et il en existe une pour chaque usage. Ainsi en éditant le manifeste d'application dans l'onglet dédié à l'interface graphique (Application UI) il est possible de définir :

- **Le Logo** : d'une taille de 150x150 pixels. Il s'agit de la tuile affichée dans le menu démarrer pour lancer l'application
- **Le Wide Logo** : d'une taille de 310x150 pixels, soit exactement la largeur de 2 tuiles standard séparées d'un espace de 10 pixels. Il s'agit de la tuile affichée en mode large dans le menu démarrer. Si cette tuile large n'est pas obligatoire dans le projet, c'est celle-ci qui est utilisée par défaut par l'application dans le menu démarrer si elle est présente.
- **Le Small Logo** : d'une taille de 30x30 pixels, c'est cette icône qui est affichée lorsque le zoom sémantique est activé sur le menu démarrer.
- **Le Badge Logo** : d'une taille de 24x24 pixels, c'est cette image qui sert d'icône d'identification de l'application et apparaît sur l'écran de verrouillage de la session si votre application supporte les notifications.
- **Le Splash Screen** : d'une taille de 620x300 pixels, il s'agit de l'image affichée au lancement de l'application.
- **Le Store Logo** : Celui-ci se trouve dans l'onglet packaging du manifeste et définit l'image affichée sur le store pour votre application. Il est d'une taille de 50x50 pixels.

Dans la plupart des cas, un minimum composé du Logo, du Small Logo, du Splash Screen, et bien sûr du Store Logo pour une mise à disposition sur le store, suffit pour un rendu satisfaisant.

> Conclusion

Cet article marque l'achèvement de la première version alpha de l'application Subway Reader. Celle-ci dispose désormais d'un design travaillé et la qualité des fichiers de code source qui la composent a pu être vérifiée grâce à des sessions de *Code Review*.

Le prochain article se focalisera sur la mise en place d'un Process de Build automatisé permettant d'automatiser la génération de notre application, couplé à l'exécution de tests unitaires.

Philippe Didiergeorges et Vivien Fabing *Consultants équipe ALM*

<http://blogs.developpeur.org/Philess>

<http://blogs.developpeur.org/vivien>

Infinite Square : <http://www.infinitesquare.com>



Fig.4

IntelliJ IDEA 11 CE

Développer pour Android

Pour le développement d'applications Android, l'IDE (Integrated Development Environment = Environnement de Développement Intégré) mis en avant par la communauté et par Google est Eclipse, avec son plugin ADP.

Cependant Eclipse n'est pas exempt de défauts, et lui trouver une alternative n'est pas chose aisée. Parmi les candidats en lice, IntelliJ IDEA Community Edition – version gratuite et open source, mais limitée, de l'IDE Java IntelliJ IDEA – se démarque de ses concurrents en reprenant la même interface et un grand nombre des fonctionnalités ayant fait le succès d'IntelliJ IDEA. Les différences par rapport à la version commerciale de l'éditeur se situent principalement au niveau du support des différentes technologies utilisées en entreprise (Spring, Struts, Hibernate...). Le support du SDK Android est par contre complètement opérationnel dans la version CE, pour notre plus grand plaisir. Nous proposons ici un aperçu de cet atelier de développement, disponible pour Linux, OSX et Windows, de ses fonctionnalités liées au langage Java en général, avec un focus pour le développement d'applications Android en particulier.

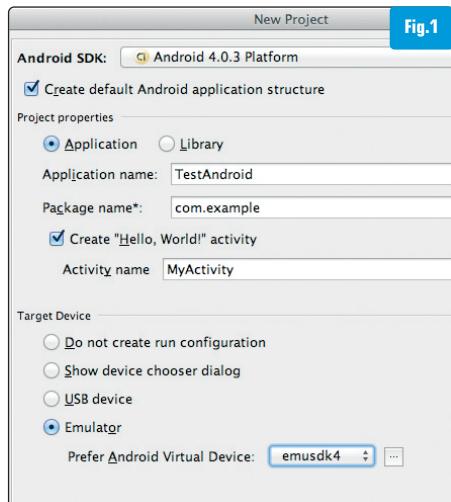
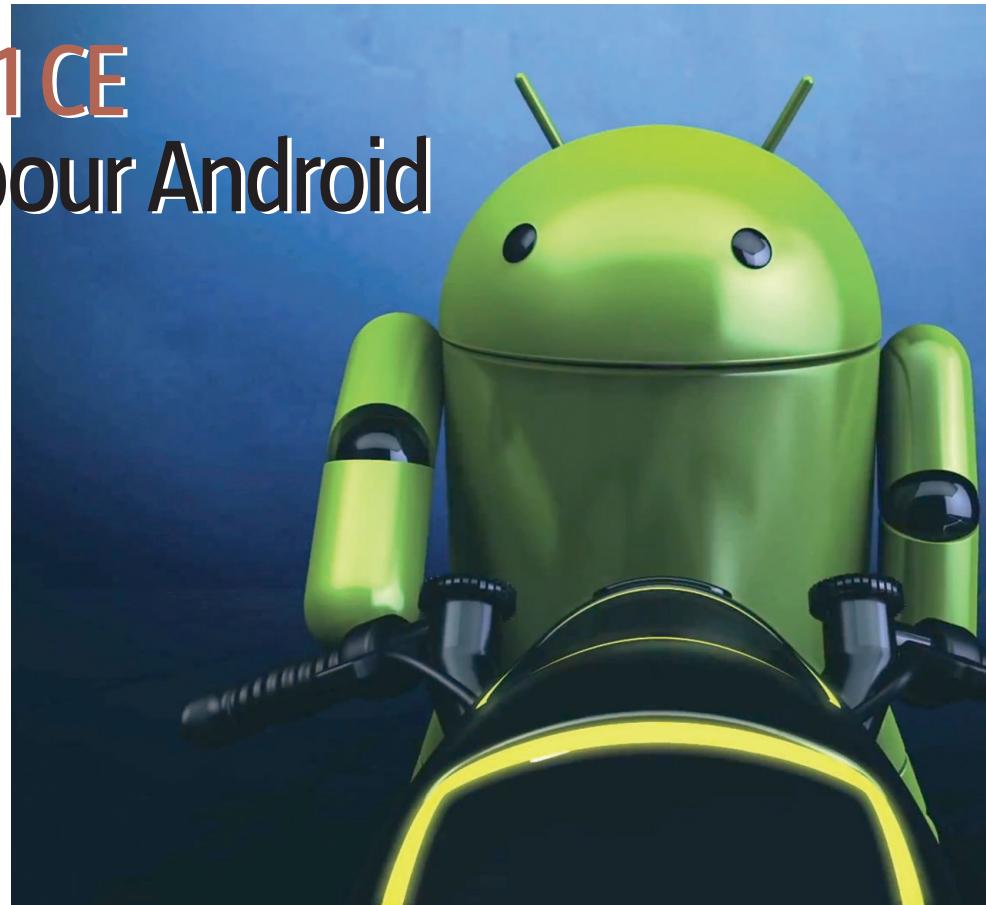


Fig.1



Installation d'IntelliJ et du SDK d'Android

Avant de pouvoir commencer le développement d'un projet Android avec IntelliJ IDEA, il faut installer le JDK (Java) ainsi que le SDK Android, et configurer les versions souhaitées de la plateforme Android. La dernière branche en date est la 4.0.x, mais pour une compatibilité plus large avec les terminaux mobiles en circulation, il est possible d'utiliser la branche 2.3.x encore très répandue. L'installation d'IntelliJ en elle-même ne pose pas de problème particulier. Le programme se récupère sur le site de l'éditeur pour la plateforme souhaitée. IntelliJ est écrit en Java, une machine virtuelle est donc requise pour son exécution. Aucune autre installation n'étant imposée, IntelliJ est prêt par défaut au développement Android, sans ajout de plugin.

Création et configuration du projet

La création d'un projet Android peut se faire de différentes façons. Il peut être créé à partir de sources existantes, localement, sur un serveur distant ou sur un gestionnaire de versions (SVN, Git, Mercurial, ...), ou bien évidemment à partir de rien. Pour ce dernier

cas, il faut choisir une version du SDK Android. Par ailleurs, une activité par défaut peut être créée avec un écran de type « Hello World » [Fig.1].

À la création du projet, le choix du moyen de débogage du projet est également donné, à savoir à l'aide de l'émulateur fourni avec le SDK ou à l'aide d'un téléphone en mode « debug ».

Les fonctionnalités générales

Ci-après sont présentées quelques unes des fonctionnalités d'IntelliJ IDEA CE liées au développement Java en général. Le but n'est pas ici d'être exhaustif, mais d'apporter une vision d'ensemble et de pointer quelques originalités [Fig.2].

Au niveau interface, IntelliJ est intuitif et propose un accès au maximum de fonctionnalités sous forme d'onglets de chaque côté de la fenêtre. Une fois sélectionné, un onglet s'ouvre sous la forme d'un volet comme les volets « projets » et « debug » de la Figure 3. Le reste est accessible par les menus de l'application, par les menus contextuels ainsi que par les raccourcis clavier.

Les fonctions principales d'un IDE concernent probablement l'aide à l'écriture de code

source. On retrouve ici les aides « visuelles » classiques, notamment les plus communes comme la coloration syntaxique, l'affichage de la documentation, la mise en exergue des erreurs ou messages d'alerte, mais aussi la mise en évidence des incohérences, comme les morceaux de code inutiles, ou encore de petits pictogrammes pour signaler des méthodes particulières (surchargées d'une

autre classe, implémentant une interface...). Conjointement, on retrouve des outils d'aide à l'écriture de code avec bien évidemment l'auto-complétion qui est accessible depuis de nombreux types de fichiers, mais aussi l'implémentation automatique de portions de code à l'aide des « live templates » qui permettent de générer des structures de code, comme celui d'une boucle `foreach` par

exemple. Certains sont accessibles par un menu contextuel comme la génération des déclarations des méthodes d'une interface, la surcharge des méthodes d'une classe parente, la génération des `getter` / `setter` d'une propriété privée ou protégée, ... comme présenté en [Fig.3].

Lors du « refactoring », le développeur peut s'aider des possibilités regroupées au sein du menu consacré. Par ailleurs, les « intents » mettent en exergue certaines portions de code qui peuvent être simplifiées ou complétées et proposent de le faire d'un simple clic. C'est ce que montre la Figure 6, avec la proposition de la simplification d'un test par la suppression de « `== true` » inutile [Fig.4].

La navigation dans le code ou la recherche de portions de code sont épaulées par des fonctions de recherche ou de navigation efficaces. La recherche se retrouve par ailleurs à de nombreux endroits dans l'interface. On peut ainsi filtrer dans la liste des méthodes et propriétés de la classe courante [Fig.8], ou encore dans la fenêtre de préférences de l'interface et du projet, pour retrouver une section bien cachée... La Figure 10 illustre la recherche parmi les options accessibles depuis la barre de menus de l'application, pratique pour retrouver rapidement cette fonctionnalité dont on a oublié le raccourci. Parallèlement à cela, il est aisément de naviguer de l'utilisation d'une méthode à son implémentation, ou de retrouver les utilisations

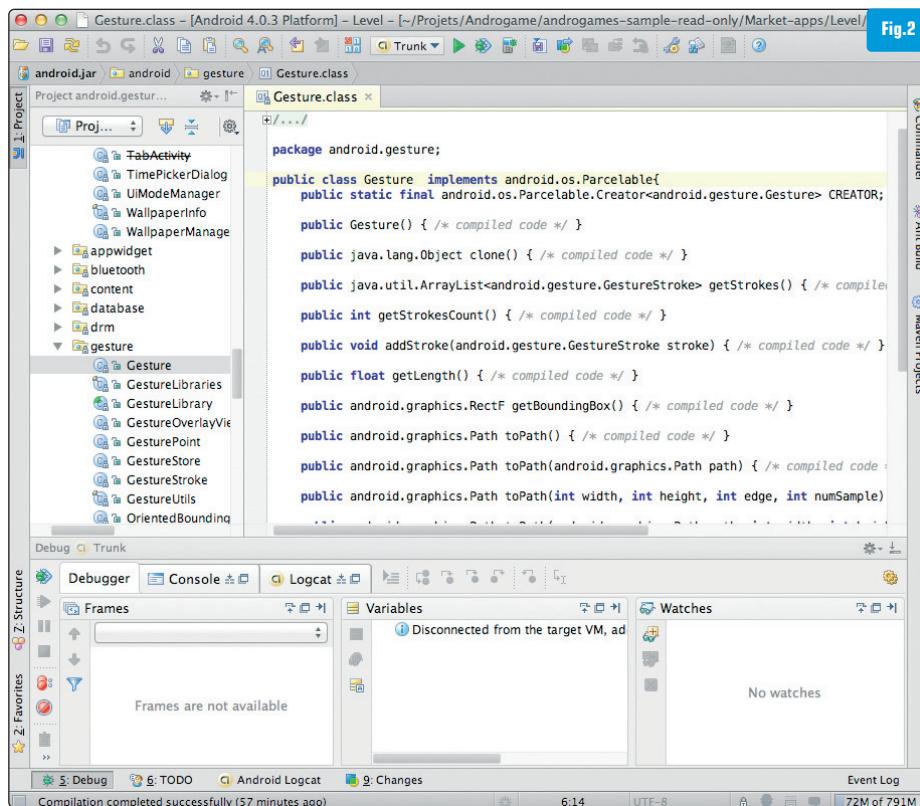


Fig.2

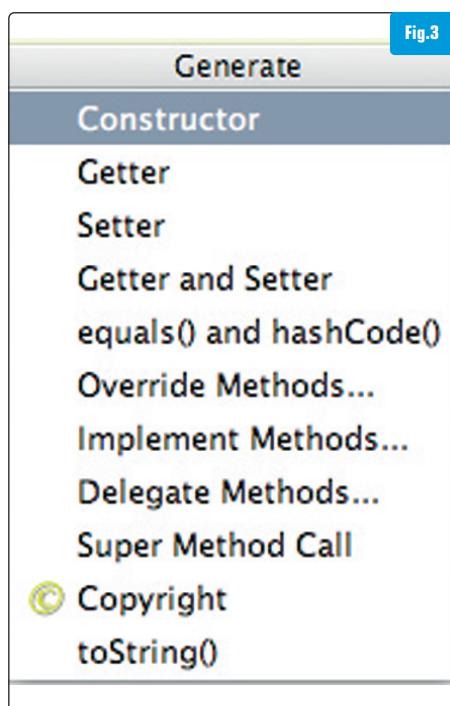


Fig.3

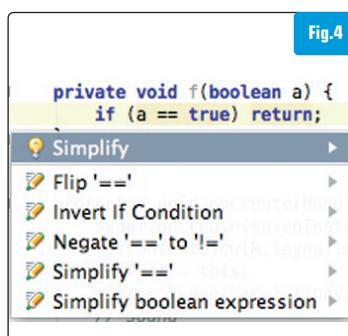


Fig.4

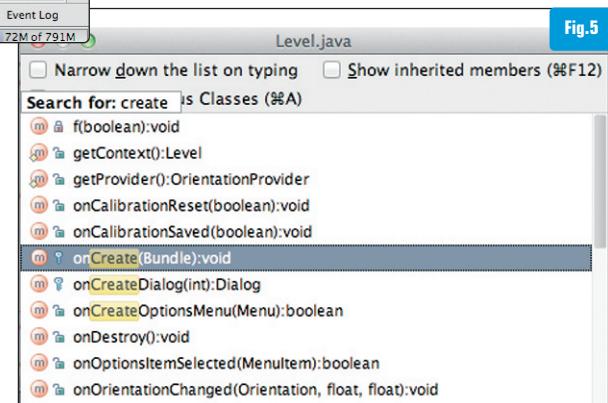


Fig.5

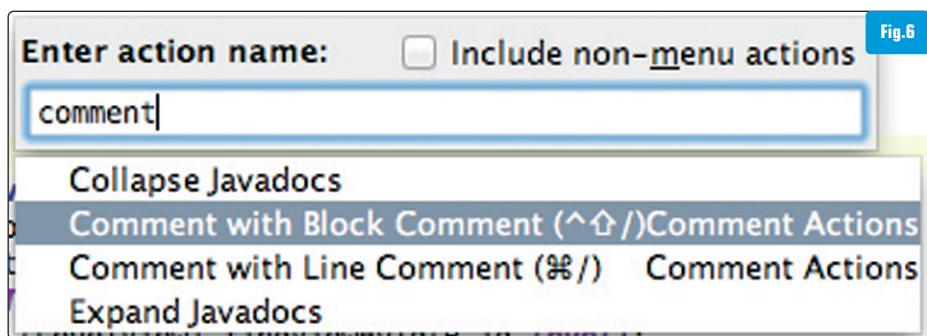


Fig.6

d'une méthode [Fig.5 et 6]. Le principe des IDE étant de rassembler tous les outils utiles aux développeurs au sein d'une même interface, au delà de la simple écriture de code elle-même, IntelliJ intègre ou s'interface avec de nombreux outils. C'est le cas de l'interconnexion avec un gestionnaire de versions, avec une base de données, avec un outil de gestion de tâches ou encore l'intégration d'outils de déploiements.

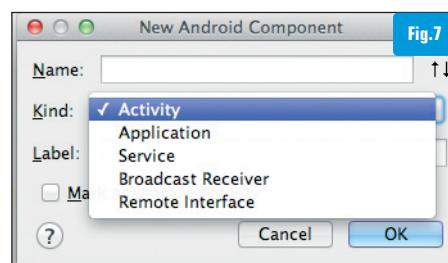
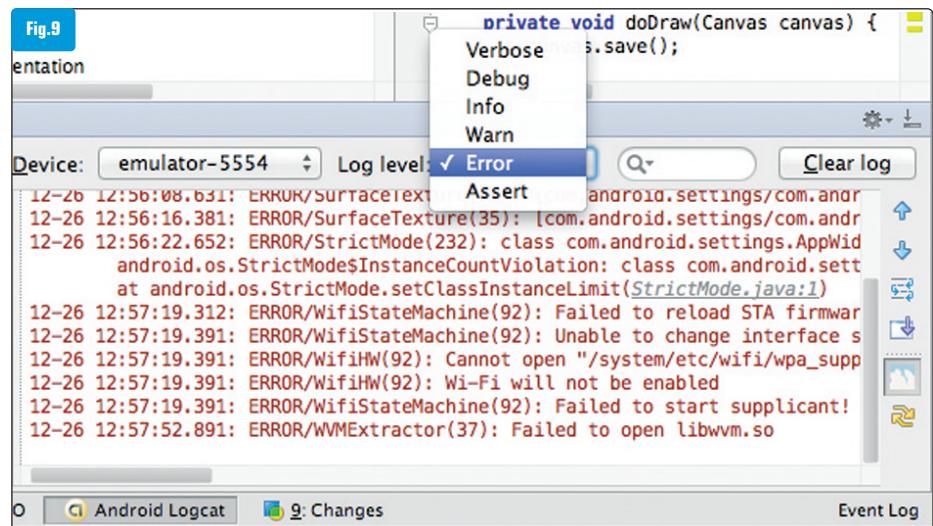
L'utilisation d'IntelliJ au quotidien offre son lot de petites fonctionnalités bien intégrées et disponibles directement. Citons le cas de l'historique local activé par défaut et indépendant de tout gestionnaire de versions, ou l'outil de comparaison et de résolution de conflits utilisable à des fins diverses comme la comparaison d'une portion de code avec le contenu du presse-papier.

Par rapport à la version complète (commerciale) d'IntelliJ IDEA, il manque quelques fonctionnalités, dont certaines auraient pu se révéler intéressantes, comme les diagrammes de classes ou la recherche structurelle (structural search) qui a pour but de rechercher un pattern personnalisé pour en proposer dynamiquement, par exemple, la suppression ou le remplacement.

N'oublions pas, pour terminer, qu'à l'instar d'Eclipse, les fonctionnalités d'IntelliJ sont extensibles au moyen de plugins.

L'intégration avec le SDK Android

En plus des fonctionnalités utiles pour tout développement Java en général, IntelliJ IDEA CE propose une intégration du SDK Android permettant d'accélérer les développements d'applications mobiles. Lors de la création d'un projet en partant de zéro, il est proposé de créer une activité par défaut avec une page de type « Hello World ». L'interface met



à disposition des assistants (Figure 12), tels que la création d'une nouvelle application ou activité, disponibles depuis l'interface, et ayant pour but de générer des squelettes de code automatiquement [Fig.7]. Le menu de création de fichiers propose de créer des fichiers adaptés au répertoire parent, comme la génération de fichiers de ressources lorsque l'on se place sur le répertoire « res ». L'accès à la documentation est rapide et présente des liens hypertextuels, menant directement vers l'aide en ligne officielle [Fig.8]. L'auto-complétion, en plus des fichiers de code source Java, est opérationnelle depuis les fichiers de ressources ainsi que dans les fichiers « manifest ».

L'aspect « debug » de l'application n'est pas en reste. Comme présenté précédemment, le mode « debug » peut utiliser l'émulateur ou un appareil physique. Le débogage est effectué depuis l'IDE à l'aide des traditionnels points d'arrêt, variables surveillées, évaluation d'instructions « à la volée » etc. La présentation des messages de log propose d'affiner l'affichage par la sélection du niveau de log souhaité et met à disposition une fonction de recherche [Fig.9].

Un point négatif est à noter

cependant : l'absence d'éditeur graphique pour la création de l'interface utilisateur. IntelliJ présente tout de même un aperçu des différents « layouts », ce qui permet d'éviter le recours à l'émulateur lorsqu'il s'agit de vérifier rapidement l'alignement d'un libellé. Enfin, IntelliJ intègre la gestion des tests unitaires créés avec le framework de tests d'Android (« *Android Testing Framework* »). On peut ainsi écrire des tests unitaires et les lancer depuis l'interface.

Conclusion

Bien que ne révolutionnant pas le développement sous Android, IntelliJ IDEA CE propose une interface complète avec toutes les fonctionnalités qu'on attend d'un bon IDE. De plus, leur intégration réussie peut séduire les développeurs les plus exigeants, d'autant plus que cette version est open source et totalement gratuite.

On peut lui reprocher cependant de ne pas intégrer d'éditeur graphique pour la réalisation des interfaces, de ne pas reprendre certaines des fonctionnalités intéressantes de la version commerciale ou encore de ne pas exister en version française. Sans pour autant dépasser Eclipse au niveau des fonctionnalités, IntelliJ met à disposition une interface efficace.

Les fonctionnalités sont là où on les attend et lorsqu'on le maîtrise bien, le gain en temps de développement est indéniable.

De plus, IntelliJ se veut simple d'emploi, et ce dès l'installation. Si vous étiez à la recherche d'une alternative à Eclipse, vous l'avez trouvée !

Gilles Bougenière,
consultant Osaxis (<http://www.osaxis.fr>)



Base de données locale WP7

Aujourd’hui, les applications mobiles sont de plus en plus sophistiquées et couvrent tous les domaines d’application. Par nature, elles sont connectées dans la plupart des cas et peuvent ainsi accéder à des données localisées dans le cloud.

Néanmoins, différentes raisons rendent l’utilisation d’un espace de stockage situé sur le téléphone lui-même indispensable, par exemple :

- Un cache local permet à la fois d’accélérer le démarrage puis l’exécution de l’application (les nouvelles données peuvent être téléchargées en arrière-plan), de réduire la bande passante utilisée (les données non modifiées n’ont pas à être téléchargées à chaque exécution de l’application)
- Les données créées par l’utilisateur lui-même n’ont pas nécessairement vocation à être transférées ou partagées (même si en pratique, les utilisateurs sont de plus en plus demandeurs de pouvoir accéder à leurs données depuis différentes machines : PC fixe, téléphone, tablette).
- Enfin, dans le cadre d’applications métier, la synchronisation vers le backoffice peut intervenir à des instants précis du processus métier (par exemple, en début et fin de journée).

Pour couvrir ces différents scénarios, Windows Phone 7 offre à chaque application un espace privé, appelé « Isolated Storage » dans lequel il est possible de créer des répertoires et des fichiers. Avec la première version de Windows Phone, les développeurs stockaient ainsi typiquement des fichiers XML ou des objets .NET serialisés lus et écrits en entier à des instants précis du cycle de vie des applications. Windows Phone Mango a apporté l’été dernier un mécanisme bien plus riche, hérité de son prédecesseur Windows Mobile, à savoir une vraie base de données relationnelle. Celle-ci n’est pas explicitement nommée dans la documentation officielle, on parle de « Local Database » mais il suffit de regarder d’un peu plus près cette documentation pour découvrir qu’il s’agit en fait d’une version de SQL Server Compact.

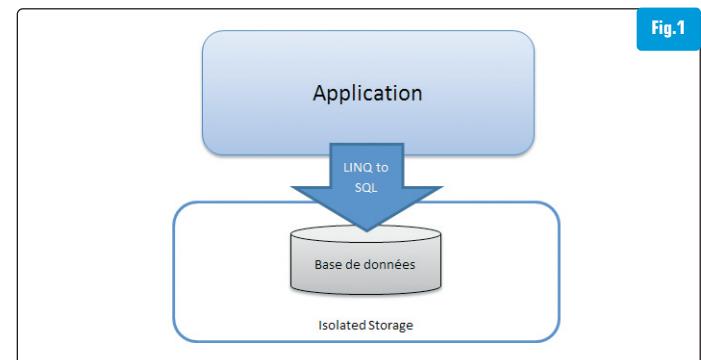
La base de données relationnelle de Windows Phone se différencie des bases de données SQL Server habituelles par différentes caractéristiques, en particulier :

- Il n’y a pas de processus SQL Server séparé : le code de la base est localisé dans des DLL qui seront simplement chargées dans le processus de l’application.
- Il n’y a pas de procédure stockée (on verra plus loin dans cet article que le développeur n’a même pas accès au langage SQL).
- Comme les fichiers constituant les bases de données sont localisés dans l’Isolated Storage, il n’y a pas de partage de données possible entre différentes applications (tout au moins sans passer par une synchronisation avec un serveur web).

Contrairement aux deux points précédents qui étaient déjà une caractéristique des autres versions de SQL Compact, le dernier aspect est une nouveauté qui est liée au modèle très sécurisé de Windows Phone qui empêche les applications de perturber leurs semblables.

La base de données et son schéma

Comme on vient de le voir, le développeur n’utilise pas le langage SQL pour accéder à la base de données : il passe obligatoirement



par une couche LINQ to SQL utilisée en mode « Code First », y compris pour la création de la base de données elle-même : on ne crée pas celle-ci à l’aide de l’Enterprise Manager mais en enrichissant des classes C# à l’aide d’attributs précisant les noms d’objets SQL, leurs types, les index à créer, etc [Fig.1]. Une table assez basique contenant un identifiant auto incrémenté de type entier ainsi qu’un libellé de type chaîne de caractères pourra être définie ainsi :

```
[Table]
public class Couleur : INotifyPropertyChanged
{
    private int _id;
    [Column(IsPrimaryKey=true, IsDbGenerated=true, DbType="INT NOT NULL Identity",
        CanBeNull=false, AutoSync=AutoSync.OnInsert)]
    public int Id { ... }

    private string _nom;
    [Column(IsPrimaryKey=false, DbType="NVARCHAR(20) NOT NULL")]
    public string Nom { ... }
}
```

On voit dans cet exemple les attributs permettant de définir la correspondance entre une classe et une table ainsi qu’entre une propriété et une colonne. TableAttribute indique qu’une table doit être créée pour assurer la persistance de la classe ; cet attribut n’a comme propriété supplémentaire que « Name » qui permet de préciser un nom de table différent de celui de la classe. En revanche ColumnAttribute est plus riche et accepte plusieurs propriétés dont les principales sont décrites dans le tableau ci-dessous :

Propriété	Type	Rôle
Autosync	Type énuméré	Indique quand récupérer la valeur automatiquement affectée par la base de données. Utilisé typiquement pour les colonnes de type identité
DbType	Chaîne	Indique le type de donnée de la colonne associé à la propriété
IsDbGenerated	Booléen	Indique si la valeur de la propriété est générée par la base de données
IsPrimaryKey	Booléen	Indique si la colonne participe à la clé primaire



Pour créer un index sur une table, il faut ajouter un attribut « Index » après l'attribut « Table » de la classe considérée, par exemple :

```
[Table]
[Index(Columns=>Nom, IsUnique=true, Name=>IX_Appellation_Nom)]
public class Appellation : BindableObject
{
```

Une relation entre tables est un peu plus compliquée à modéliser qu'une simple colonne : l'idéal est d'avoir des propriétés « orientées objet » au niveau des classes C# mais les données sauvegardées en base sont de simples identifiants. Des classes génériques comme EntityRef et EntitySet permettent cette modélisation... en les accompagnant d'un peu de code. Ainsi, une relation 1-N sera typiquement implémentée de la manière suivante :

- dans la classe fille, l'entité parente sera définie en C# à l'aide d'un attribut simple (contenant la clé étrangère) et d'une « EntityRef » décorés d'attributs permettant de décrire plus précisément le mapping en base de données
- dans la classe mère, l'association sera stockée dans un « EntitySet » dont on s'abonne aux événements Attach et Detach si on veut automatiquement mettre à jour une entité fille lorsqu'elle est ajoutée à cette collection.

```
[Table]
public class Appellation : INotifyPropertyChanged
{
...
[Column(IsPrimaryKey = false, DbType = « INT NOT NULL », CanBeNull = false)]
internal int IdRegion { get; set; }

private EntityRef<Region> _region;
[Association(Storage = « _region », ThisKey = « IdRegion », OtherKey = « Id »,
    IsForeignKey = true)]
public Region Region { ...}

}

[Table]
public class Region : INotifyPropertyChanged
{
private EntitySet<Appellation> _appellations;
[Association(Storage = « _appellations », OtherKey = « IdRegion »,
ThisKey = « Id »)]
public EntitySet<Appellation> Appellations
{
    get { return this._appellations; }
    set {
        this._appellations.Assign(value);
        OnPropertyChanged(« Appellations »);
    }
}
public Region()
{
...
_appellations = new EntitySet<Appellation>(
    new Action<Appellation>(this.attach_Appellation),
    new Action<Appellation>(this.detach_Appellation)
```

```
    );
}
private void attach_Appellation(Appellation appellation)
{
    appellation.Region = this;
}
private void detach_Appellation(Appellation appellation)
{
    appellation.Region = null;
}
```

Une fois toutes les classes définies et correctement décorées d'attributs, on peut les regrouper dans un DataContext qui va être le point de passage obligé pour tous les échanges avec la base, c'est-à-dire à la fois la création de celle-ci, les requêtes de sélection et les requêtes de modifications.

```
public class WineDataContext : DataContext
{
    public WineDataContext(string connectionString = null)
        : base(connectionString ?? DefaultConnectionString) { }

    public Table<Pays> Pays;
    public Table<Couleur> Couleurs;
    public Table<TypeAppellation> TypesAppellation;
    public Table<Region> Regions;
    public Table<SousRegion> SousRegions;
    public Table<TypeGarde> TypesGardes;
    public Table<Appellation> Appellations;
    public Table<Vin> Vins;
    ...
}
```

Création de la base de données

Une fois défini l'ensemble des classes modélisant le schéma de la base de données, l'étape suivante est la création de la base elle-même : cette création doit avoir lieu sur un téléphone ou un émulateur. Toute autre méthode de constitution du .SDF n'est pas supportée même si elle semble fonctionner au moins dans les cas simples : « *Microsoft SQL Server Compact (SQL CE) databases that have been created on the desktop may work with a Windows Phone application, but are not officially supported*[1] ». C'est donc en appellant la méthode CreateDatabase du DataContex que le fichier SDF initial est créé :

```
// Re-création forcée de la base de données
using (WineDataContext db = new WineDataContext(« Data Source =isostore:/BaseVins.sdf »))
{
    if (db.DatabaseExists())
    {
        if (MessageBox.Show(« La base existe déjà, voulez-vous vraiment la supprimer ? », « Confirmation », MessageBoxButton.OKCancel) == MessageBoxResult.Cancel)
        {
            return;
        }
    }
}
```

[1] [http://msdn.microsoft.com/en-us/library/hh286411\(v=VS.92\).aspx](http://msdn.microsoft.com/en-us/library/hh286411(v=VS.92).aspx)

```

    }
    db.DeleteDatabase();
}
db.CreateDatabase();
}

```

Lorsque ce fichier est créé, l'ensemble des tables appartenant au `DataContext` utilisé le sont aussi, avec les index mais aussi les relations entre tables. Si créer le fichier SDF depuis l'Enterprise Manager puis le déplacer sur le téléphone n'est pas supporté, il est en revanche possible de récupérer ce fichier une fois créé par `CreateDatabase` et de l'ouvrir avec l'Enterprise Manager.

Pour effectuer cette opération, on peut utiliser l'« Isolated Storage Explorer Tool » installé avec le SDK Windows Phone et situé habituellement dans le répertoire « Program Files (x86)\Microsoft SDKs\Windows Phone\v7.1\Tools\IsolatedStorageExplorerTool ». La commande de transfert des fichiers de l'émulateur vers le PC est la suivante :

```
ISETTool.exe ts xd guid_de_l_application_lu_dans_WAppManifest.xml IsolatedStorage
```

Une fois le fichier transféré, on l'ouvre en sélectionnant « SQL Server Compact » dans les options de connexion et en choisissant le .SDF ainsi récupéré [Fig.2].

On peut ainsi vérifier à la fois la structure des tables et des index et les données insérées en exécutant quelques requêtes SQL [Fig.3].

Fig.3

```

select Region.Nom, Appellation.Nom
from Region inner join Appellation on Region.Id=Appellation.IdRegion
order by Region.Nom, Appellation.Nom

```

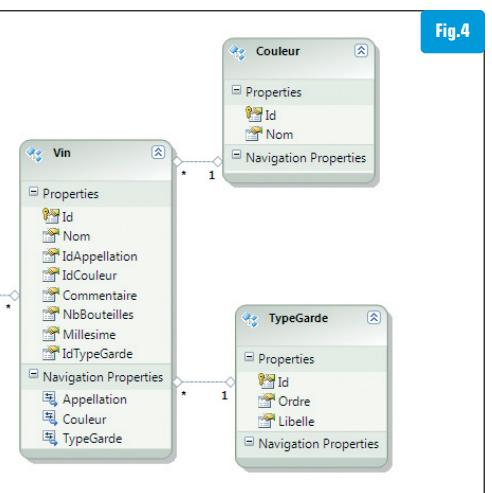
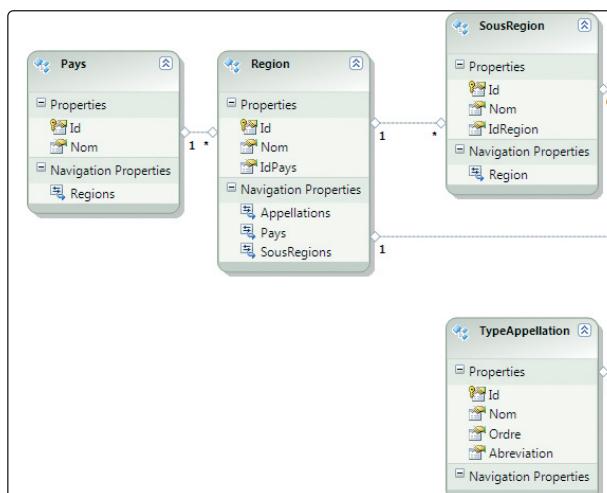
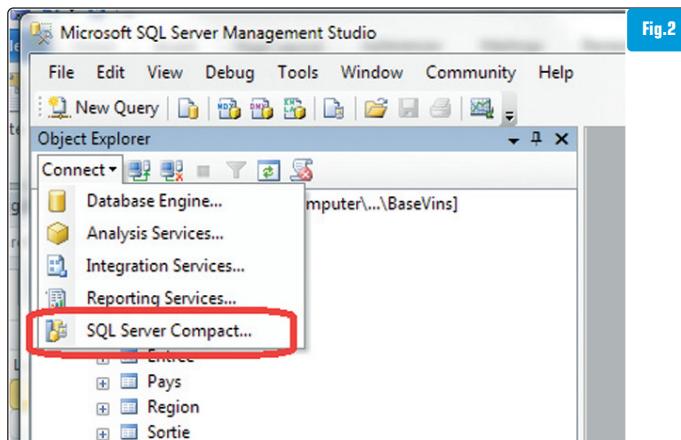
Region	Appellation
1 Abruzzes	Montepulciano D'Abruzzo Colline Teramane
2 Alsace	Alsace
3 Alsace	Alsace Grand Cru
4 Alsace	Alsace Cotes d'Alsace
5 Alsace	Vin D'Alsace
6 Beaujolais	Beaujolais
7 Beaujolais	Beaujolais-Villages
8 Beaujolais	Brouilly
9 Beaujolais	Chénas
10 Beaujolais	Chiroubles
11 Beaujolais	Côte De-Brouilly
12 Beaujolais	Fleurie
13 Beaujolais	Juillenais
14 Beaujolais	Morgon
15 Beaujolais	St. Jean

Remarque : il n'est pas possible de générer un diagramme de cette base avec SQL Enterprise Manager... Une solution consiste à créer un projet d'application Windows dans Visual Studio puis d'ajouter un ADO.NET Entity Data Model dans lequel on ajoute ensuite l'ensemble des tables de la base SQL Compact. On peut ainsi visualiser graphiquement le schéma généré à l'aide des annotations [Fig.4].

Si la génération du fichier SDF sur un PC n'est pas supportée, il est en revanche possible de copier et utiliser sur n'importe quel téléphone un fichier ainsi généré et récupéré. Il faut néanmoins noter que le fichier SDF devra être copié dans l'Isolated Storage avant d'être ouvert par l'application Windows Phone, cela donne donc différentes possibilités de déploiement, chacune nécessitant une opération au premier lancement de l'application :

- créer la base avec le code montré précédemment et la remplir (par exemple à partir d'un fichier XML contenant les données initiales) au premier lancement de l'application,
- déployer le SDF avec l'application, le recopier depuis le fichier XAP vers l'Isolated Storage au premier lancement de l'application,
- télécharger le fichier SDF depuis un serveur Web et le stocker dans l'Isolated Storage.

En termes de performances, la meilleure solution consiste à déployer le fichier avec l'application. On peut néanmoins combiner cela avec une interrogation d'un serveur Web, ce qui permet de mettre à jour la base initiale sans avoir à soumettre une nouvelle fois l'application au niveau du marketplace.





Mise à jour du schéma

Lorsque le schéma de la base de données évolue, il est possible dans un certain nombre de cas de mettre une base de données existante à jour. Bien évidemment, si les changements sont trop importants, il va falloir créer une nouvelle base et transférer par du code ad hoc les données préalablement créées dans l'ancienne base. Dans les cas « favorables », on peut s'appuyer sur la classe `DatabaseSchemaUpdater` pour effectuer la mise à jour du schéma. Le principe est le suivant :

- à la création de la base, on enregistre le numéro de version du schéma dans la base elle-même : ceci se fait à l'aide de la propriété `DatabaseSchemaVersion` du `DatabaseSchemaUpdater`.
- lorsque le schéma évolue, on s'appuie sur les méthodes génériques `AddTable<>`, `AddColumn<>`, `AddIndex<>` et `AddAssociation<>` du `x` pour mettre à jour le schéma : il suffit de préciser le nom de l'objet à ajouter car sa description est lue ensuite dans les attributs de la classe C#. Par exemple :

```
dbUpdater.AddColumn<Sortie>(`Commentaire`);  
dbUpdater.DatabaseSchemaVersion = WineDataContext.SchemaVersion;  
dbUpdater.Execute();
```

Options de connexion à la base de données

Ce paragraphe sur la mise en place de la base de données ne serait pas complet si on ne décrivait pas quelques options intéressantes qu'on peut positionner dans la chaîne de connexion. Il est par exemple possible de crypter la base de données en fournissant un mot de passe, ce qui assure une certaine sécurité des données si un téléphone venait à être égaré ou volé. Voici les principaux paramètres disponibles :

Paramètre	Description
Data Source	Chemin du fichier .SDF : Isostore:xxx ou appData:xxx
Password / pwd	Mot de passe : la base est alors encryptée à l'aide de AES-128 et le mot de passe hashé à l'aide de SHA-256.
Max Buffer Size	Taille en Ko du tampon que la base peut créer en mémoire vive avant d'aller écrire sur le disque (i.e. la mémoire flash)
Max database Size	Taille maximale de la base de données en Mo, par défaut = 32 Mo
Culture Identifier	Culture utilisée pour les comparaisons, par défaut en-US. Doit être précisée lors de la création de la base car est ignorée ensuite.
Case Sensitive	Sensibilité à la casse. Doit être précisée lors de la création de la base car est ignorée ensuite.

Exécution de requêtes

Comme il a déjà été dit, le programmeur n'a pas accès au langage SQL pour manipuler la base de données : il faut obligatoirement passer par les classes du datacontext qui reflètent les tables de la base et on n'écrit finalement des requêtes qu'en lecture.

Requêtes de sélection

On récupère les entités à l'aide d'une requête Linq sur le datacontext, par exemple :

```
var query = from Region r in _db.Regions where r.Pays.Nom == `France`  
          orderby r.Nom select r;  
RegionsSimples = new ObservableCollection<Region>(query);
```

Dans le cas où le nombre de lignes retourné est trop important, on peut faire de la pagination à l'aide de `Skip<>` et `Take`, par exemple :

```
var query = from Region r in _db.Regions where r.Pays.Nom == `France`  
          orderby r.Nom select r;  
RegionsSimples = new ObservableCollection<Region>(query.Skip(10).Take(5));
```

Enfin, dans le cas où on sait à l'avance qu'on va chercher les entités filles pour tous les éléments affichés, on peut les récupérer en même temps que les entités mères en précisant des options d'exécution de la requête. Dans l'exemple ci-dessous, on charge les appellations de chaque région dès le départ :

```
DataLoadOptions options = new DataLoadOptions();  
options.LoadWith<Region>(r => r.Appellations);  
var query = from Region r in _db.Regions where r.Pays.Nom == `France`  
          orderby r.Nom select r;  
Regions = new ObservableCollection<Region>(query);
```

Opérations de modification

Pour insérer des entités en base de données, il faut créer l'entité à ajouter et appeler `InsertOnSubmit` sur la table à laquelle elle se rattaché : l'enregistrement sera effectif quand les modifications du datacontext seront enregistrées à l'aide de `SubmitChanges` :

```
Vin nouveau = new Vin();  
nouveau.Appellation = Appellations.FirstOrDefault();  
nouveau.Couleur = Couleurs.FirstOrDefault();  
nouveau.TypeGarde = TypesGarde.FirstOrDefault();  
_db.Vins.InsertOnSubmit(nouveau);  
Vins.Add(nouveau);
```

Puis :

```
_db.SubmitChanges();
```

Pour supprimer une entité, il faut appeler `DeleteOnSubmit` sur la table d'où provient l'entité datacontext :

```
_db.Vins.DeleteOnSubmit(vin);
```

Au niveau des modifications, il n'y a rien de particulier à faire car les modifications sur une entité chargée depuis un `DataContext` seront enregistrées lors de l'appel à `SubmitChanges` sur celui-ci. Par ailleurs, il faut noter qu'en implémentant l'interface `INotifyPropertyChanged`, ces entités peuvent être directement « bindées » au niveau du XAML.

Conclusion et références

Même si elle est moins mise en avant que ses versions précédentes pour Windows Mobile, la base de données relationnelle de Windows Phone 7 capitalise sur ce passé et constitue un excellent moyen de manipuler des données structurées sur le téléphone, que ce soit pour des applications d'entreprise ou des applications grand public ; la compatibilité des entités manipulées avec le modèle de développement MVVM en réduit par ailleurs le coût de mise en œuvre en simplifiant l'utilisation.

Pour aller plus loin, consulter la documentation MSDN : [http://msdn.microsoft.com/en-us/library/hh202865\(v=VS.92\).aspx](http://msdn.microsoft.com/en-us/library/hh202865(v=VS.92).aspx) et en particulier la page sur les Best Practices : [http://msdn.microsoft.com/en-us/library/hh286406\(v=vs.92\).aspx](http://msdn.microsoft.com/en-us/library/hh286406(v=vs.92).aspx)

Alain Zanchetta - Architecte - Microsoft Services

Donnez une nouvelle dimension à vos applications Android grâce aux Widgets

La bataille du mobile continue de faire rage avec une domination désormais sans partage des plateformes Android et iOS. Si chaque camp ne cesse de mettre en avant la prédominance de sa plateforme, il existe pourtant une fonctionnalité exclusive à Android qui fait pâlir de jalouse les iPhone addicts de la première heure. Cette fonctionnalité, c'est les Widgets dont nous allons voir la mise en œuvre afin d'étoffer vos applications Android.

A force de travail et à coup de mises à jour au rythme toujours plus effréné, la plateforme du géant de Mountain View a comblé son retard sur iOS pour s'inscrire en référence sur plusieurs points parmi lesquels se trouvent les Widgets. Concrètement, un Widget est un élément d'une application qui peut être embarqué au sein d'autres applications mais qui se destine avant tout à être affiché sur l'écran d'accueil.

Recevant des notifications de mise à jour périodiques, les Widgets permettent d'afficher une vue des fonctionnalités d'une application directement sur l'écran d'accueil d'un appareil Android. Apportant une plus-value non négligeable à une application, ils sont de fait très appréciés des utilisateurs finaux.

Globalement, ces Widgets peuvent être classés en 2 catégories :

- Les Widgets natifs, installés nativement sur le périphérique avec comme exemple célèbre la barre de recherche Google.
- Les Widgets non natifs, ajoutés au périphérique à l'installation de nouvelles applications.

Ici, nous nous intéresserons à ce second type de Widgets qui va nous permettre de proposer aux utilisateurs une personnalisation de leur bureau Android via certaines fonctionnalités d'une application.

STRUCTURE D'UN WIDGET

Arrivés avec la version 1.5 du SDK Android, les Widgets bénéficient d'un framework dédié.

L'AppWidget Framework permet ainsi la gestion à la fois des App Widget Host, qui sont des composants applicatifs destinés à recevoir des Widgets, et également des App Widget Provider qui permettent la publication de vues en tant que Widgets et sont donc le sujet central de cet article.

Techniquement parlant, un Widget est simplement un BroadcastReceiver Android classique couplé à des données de configuration XML décrivant ses caractéristiques. De fait, la création d'un Widget ne nécessite que les 3 éléments suivants :

- Un objet de type AppWidgetProviderInfo déclaré au sein d'un fichier XML dans le dossier res/xml et qui permet notamment de définir les dimensions du Widget, son layout initial ainsi que sa fréquence de rafraîchissement.
- Une implémentation de la classe AppWidgetProvider permettant d'interagir avec les évènements que le Widget va recevoir du fait de son statut de BroadcastReceiver.
- La déclaration du Widget en tant que BroadcastReceiver directement au sein de l'AndroidManifest de l'application.

La définition des dimensions du Widget se fait en tenant compte des spécificités de l'écran d'accueil des périphériques Android. Ce dernier utilisant un layout basé sur des cellules, il est nécessaire pour définir les dimensions d'un Widget de raisonner en termes de cellules et donc de déterminer le nombre de cellules d'écran que le Widget pourra occuper aussi bien en largeur qu'en hauteur. Pour ce faire, il est nécessaire d'appliquer l'équation suivante :

Taille minimum en dip = (Nombre de cellules * 74dip) - 2dip

Ainsi, un Widget de type 4x2 soit 4 cellules en largeur et 2 en hauteur aura une dimension minimum de 294dip en largeur et de 146dip en hauteur.

Du fait de leur nature de BroadcastReceiver, les Widgets sont exposés aux blocages de type ANR (Application Not Responding). Pour éviter ce phénomène et proposer aux utilisateurs des Widgets réactifs, la vue d'un Widget est construite et envoyée sous la forme d'un objet RemoteViews depuis un service tournant en arrière-plan.

Concernant la partie présentation, le layout initial d'un Widget se construit de manière classique à cela près que ce dernier est basé sur un objet RemoteViews lequel ne supporte pas tous les types de layouts et de vues. Cela limite ainsi les layouts au FrameLayout, au LinearLayout et au RelativeLayout.

Au niveau des vues, une douzaine sont supportées parmi lesquelles on citera les Button, TextView, ImageView et autres ImageButton. Bref, les limitations inhérentes aux RemoteViews ne sont pas une barrière pour la réalisation de Widgets au visuel convaincant.

Enfin, un mot sur la fréquence de rafraîchissement des Widgets. Les Widgets étant conçus pour proposer du contenu à durée de vie plutôt longue, il est évident que la fréquence définie pourra influer sur la consommation batterie du périphérique. Il conviendra donc de la définir avec le plus grand soin.

UN PREMIER WIDGET

Les bases théoriques posées, il est temps de passer à la réalisation d'un premier Widget dont la fonctionnalité sera d'afficher le classique message « Hello World » sur l'écran d'accueil d'un périphérique Android suivi de sa date de dernière mise à jour. Nous créons un projet FirstWidget sans activité.

Le projet cible Android 4.0 tout en autorisant une version minimum du SDK en 1.5 correspondant à la première mise à disposition de l'AppWidget Framework.

La première étape consiste à créer le layout du Widget. Trivial, ce



layout se contente d'afficher le message « Hello World ! » suivi de la date de dernière mise à jour. Son contenu est défini dans le fichier `widget_layout.xml` :

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello"
        android:layout_centerHorizontal="true"
        style="@style/widgetFont" />
    <TextView
        android:id="@+id/lastUpdate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dip"
        android:layout_centerHorizontal="true"
        style="@style/widgetFont" />
</RelativeLayout>
```

Cette première étape franchie, nous passons à la création du Widget à proprement parler, via la définition de son `AppWidgetProviderInfo` au sein d'un fichier XML nommé `widget_hello` et placé dans le dossier `res/xml` :

```
<appwidget-provider
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:minWidth="294dip"
    android:minHeight="72dip"
    android:updatePeriodMillis="3600000"
    android:initialLayout="@layout/widget_layout"
    >
</appwidget-provider>
```

Notre Widget aura une dimension de type `4x1`, c'est-à-dire qu'il occupera 4 cellules d'écran en largeur et 1 en hauteur.

Globalement, il est bon de garder à l'esprit que pour garantir une portabilité optimale entre les différents types de périphériques Android, il ne faut jamais dépasser des dimensions de `4x4` pour un Widget.

En appliquant l'équation définie plus haut, on obtient ainsi une largeur minimale de `294dip` ainsi qu'une hauteur de `72dip`.

La fréquence de rafraîchissement, définie en millisecondes, est d'une heure. Enfin, le layout `widget_layout` défini précédemment est appliqué comme layout initial du Widget.

La configuration XML du Widget bien avancée, il est temps de passer à la définition d'une implémentation de la classe `AppWidgetProvider` qui va permettre au Widget d'interagir avec les événements qu'il va recevoir. Dans notre cas, seule la méthode `onUpdate` est à surcharger et c'est au sein de celle-ci que nous lançons le service chargé de sa mise à jour.

La création de ce service s'exécutant en arrière-plan vise à prévenir d'éventuels problèmes d'ANR. Le code du Widget ainsi que celui du service somme toute assez simple est le suivant :

```
public class HelloWorldWidget extends AppWidgetProvider {
    private static SimpleDateFormat DATE_FORMATTER = new SimpleDateFormat("HH:mm:ss dd/MM/yyyy");

    @Override
    public void onUpdate(Context context, AppWidgetManager appWidgetManager, int[] appWidgetIds) {
        context.startService(new Intent(context, UpdateService.class));
    }

    // Service de mise à jour du Widget
    public static class UpdateService extends Service {
        @Override
        public IBinder onBind(Intent intent) {
            return null;
        }

        @Override
        public void onStart(Intent intent, int startId) {
            RemoteViews updateViews = buildUpdate(this);
            ComponentName thisWidget = new ComponentName(this, HelloWorldWidget.class);
            AppWidgetManager manager = AppWidgetManager.getInstance(this);
            manager.updateAppWidget(thisWidget, updateViews);
        }

        public RemoteViews buildUpdate(Context context) {
            RemoteViews updateViews = new RemoteViews(context.getPackageName(), R.layout.widget_layout);
            updateViews.setTextViewText(R.id.lastUpdate, DATE_FORMATTER.format(new Date()));
            return updateViews;
        }
    }
}
```

On notera bien l'utilisation d'un objet `RemoteViews` qui est ensuite transmis au Widget et qui encapsule le contenu visuel de celui-ci. Il est intéressant également de souligner que l'objet `RemoteViews` ne permet pas d'accéder directement aux instances des widgets visuels d'un layout mais qu'il est nécessaire d'utiliser des méthodes dédiées. Ainsi, la modification de la date de mise à jour sur le champ texte du `widget_layout` se fait via l'appel à la méthode `setTextViewText` avec en entrée l'id du champ et la valeur que l'on souhaite lui affecter.

La dernière étape consiste à câbler le tout dans l'`AndroidManifest` de l'application `FirstWidget` de la sorte :

```
<receiver
    android:label="@string/widget_name"
    android:name=".HelloWorldWidget" >
    <intent-filter >
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
    </intent-filter>

    <meta-data
        android:name="android.appwidget.provider"
        android:resource="@xml/widget_hello" />

```

```
</receiver>

<service android:name=".HelloWorldWidget$UpdateService" />
```

Le Widget HelloWorldWidget est déclaré en tant que BroadcastReceiver répondant à l'Intent APPWIDGET_UPDATE alors que la balise meta-data permet de définir l'objet AppWidgetProviderInfo utilisé. Enfin, le service de mise à jour du Widget est également déclaré. Notre application installée, la sélection du Widget se fait via le menu idoine où il est affiché avec son nom « Hello World ! Widget » [Fig.1]. Son affichage sur un bureau Android montre qu'il remplit bien sa fonction en affichant le célèbre message « Hello World » suivi de la date de mise à jour [Fig.2].

UN WIDGET HORLOGE

Aussi sympathique qu'il soit, ce Widget reste limité et il est intéressant de se confronter à un cas plus concret pour mettre en exergue certaines problématiques inhérentes aux Widgets.

Dans cette optique, nous allons mettre en place un Widget de type Horloge digitale ayant pour mission d'afficher l'heure courante. La démarche reste la même que celle suivie précédemment avec la déclaration d'un Widget de type 4x2 pour lequel la fréquence de rafraîchissement sera minimale afin d'afficher correctement les heures et minutes :

```
<appwidget-provider
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:minWidth="294dp"
    android:minHeight="146dp"
    android:updatePeriodMillis="5000"
    android:initialLayout="@layout/clock_widget"
    >
</appwidget-provider>
```

Le layout clock_widget est assez simple, se contentant d'afficher les heures et minutes au sein de 2 rectangles de couleur bleu clair afin de donner un peu de style au Widget :

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/
    res/android"
```

```
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dip" >

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="5dip" >

        <RelativeLayout
            android:layout_width="90dip"
            android:layout_height="wrap_content"
            android:layout_marginRight="10dip"
            android:background="@drawable/hour_border" >

            <TextView
                android:id="@+id/hours"
                style="@style/hourWidgetFont"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerInParent="true" />

            <RelativeLayout
                android:layout_width="90dip"
                android:layout_height="wrap_content"
                android:background="@drawable/hour_border" >

                <TextView
                    android:id="@+id/min"
                    style="@style/hourWidgetFont"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_centerInParent="true" />

            </RelativeLayout>
        </RelativeLayout>
    </LinearLayout>
```



Fig.1 Sélection du Widget

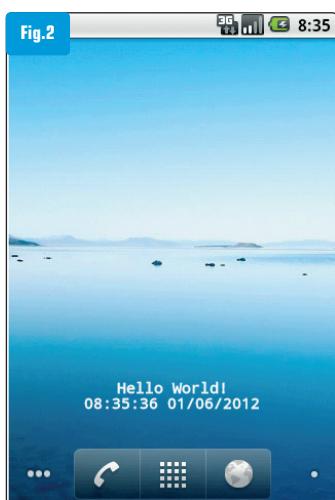


Fig.2 Affichage du Widget

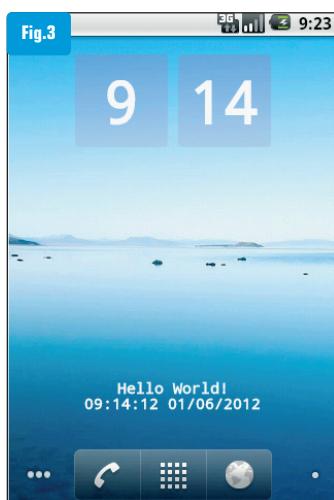


Fig.3 Widget Horloge

La classe implémentant la logique de traitement du Widget se base sur le même principe que ce qui a été vu précédemment avec un service UpdateService lancé lors de l'appel à sa méthode onUpdate. La construction de la RemoteViews associée au Widget se différenciant seulement par le fait qu'il est nécessaire d'afficher l'heure courante ainsi que les minutes courantes aux widgets visuels du layout clock_widget. Au niveau de l'AndroidManifest de notre application, le câblage est également du même acabit. Le lancement de



notre application nous permet de vérifier que notre Widget affiche correctement une horloge digitale [Fig.3].

GESTION DU RAFRAÎCHISSEMENT

Le Widget affiche bien une horloge digitale mais on peut se rendre compte sur la copie d'écran de la figure 3 que l'heure n'est pas correctement mise à jour en revanche. La fréquence de rafraîchissement du Widget étant de 5 secondes, il est étonnant de constater que cela n'a pas d'effet sur notre horloge.

En s'intéressant de plus près à la documentation de l'AppWidget Framework, on y décèle rapidement la cause : il est mentionné que la propriété `updatePeriodMillis` permet bien de définir la fréquence de rafraîchissement du Widget mais il est surtout indiqué que la valeur définie n'est en aucun cas garantie par l'OS ...

La solution à cette problématique réside dans l'utilisation du service système `AlarmManager` qui permet à travers les mécanismes d'`Intent` et de `BroadcastReceiver` de positionner des exécutions programmées d'une application dans le futur.

Dans notre cas, nous utilisons un `Intent` spécifique `CLOCK_WIDGET_UPDATE` pour lequel le Widget sera à l'écoute. Cela se traduit par l'ajout de l'action suivante au sein de la zone `intent-filter` du `ClockWidget` :

```
<action android:name="fr.app.firstwidget.ClockWidget.CLOCK_WIDGET_UPDATE" />
```

Au niveau de la classe `ClockWidget`, nous allons programmer à son activation l'envoi d'un `Intent` `CLOCK_WIDGET_UPDATE` via l'`AlarmManager`. Dans sa méthode `onReceive`, nous lançons le service de mise à jour de l'horloge digitale. Enfin, il ne faut pas oublier d'annuler l'envoi de l'`Intent` par l'`AlarmManager` lors de la désactivation du Widget. Le code rajouté permettant ce fonctionnement est le suivant :

```
public class ClockWidget extends AppWidgetProvider {  
    public static final String CLOCK_WIDGET_UPDATE = "fr.app.firstwidget.ClockWidget.CLOCK_WIDGET_UPDATE";  
  
    private PendingIntent createUpdateIntent(Context context) {  
        Intent intent = new Intent(CLOCK_WIDGET_UPDATE);  
        PendingIntent pendingIntent = PendingIntent.getBroadcast(context, 0, intent, PendingIntent.FLAG_UPDATE_CURRENT);  
        return pendingIntent;  
    }  
  
    @Override  
    public void onDisabled(Context context) {  
        super.onDisabled(context);  
        AlarmManager alarmManager = (AlarmManager) context.getSystemService(Context.ALARM_SERVICE);  
        alarmManager.cancel(createUpdateIntent(context));  
    }  
  
    @Override  
    public void onEnabled(Context context) {  
        super.onEnabled(context);  
        AlarmManager alarmManager = (AlarmManager) context.getSystemService(Context.ALARM_SERVICE);  
        Calendar calendar = Calendar.getInstance();
```

```
        calendar.setTimeInMillis(System.currentTimeMillis());  
        calendar.add(Calendar.SECOND, 1);  
        alarmManager.setRepeating(AlarmManager.RTC, calendar.getTimeInMillis(), 1000, createUpdateIntent(context));  
    }  
  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        super.onReceive(context, intent);  
        if (CLOCK_WIDGET_UPDATE.equals(intent.getAction())) {  
            context.startService(new Intent(context, UpdateService.class));  
        }  
    }  
    // ...  
}
```

ECRAN DE CONFIGURATION

Comme expliqué précédemment, placer une alarme qui se répète toutes les secondes reste quelque chose de très consommateur pour la batterie du périphérique. Dans le cas d'une horloge, cela s'impose mais dans la plupart des cas cela n'a pas vraiment de raison d'être. Prenons l'exemple d'une application météo qui affiche, via un Widget, une vision actualisée de la météo courante de la ville où se trouve l'utilisateur. Dans ce cas d'utilisation, il va être préférable de laisser l'utilisateur choisir la fréquence de rafraîchissement qu'il souhaite pour les données météo, ce qui lui donne la possibilité de gérer la consommation de sa batterie.

Le choix de cette fréquence va prendre place au sein de l'écran de configuration du Widget. Pour ce faire, l'objet `AppWidgetProviderInfo` propose une propriété `configure` dans laquelle il faut préciser l'activité utilisée pour sa configuration.

Au sein de cette activité, nous pouvons proposer un spinner permettant à l'utilisateur de sélectionner la fréquence de son choix. Avant de terminer l'activité avec la méthode `finish` il faudra prendre soin d'enregistrer son choix au sein des préférences de l'application par exemple. C'est par ce biais qu'il faudra ensuite récupérer la fréquence retenue pour l'appliquer à l'`AlarmManager` lors de la mise à jour du `ClockWidget`.

CONCLUSION

Fonctionnalité exclusive du monde Android, les Widgets connaissent un grand succès. Dès lors, il devient primordial pour une application de proposer son ou ses Widgets spécifiques exhibant une fonctionnalité particulière ou permettant de contrôler l'exécution de l'application. Facile d'accès, l'AppWidget Framework se révèle très puissant donnant aux développeurs la possibilité d'imaginer toutes sortes de Widgets. Il convient néanmoins de prendre soin d'exécuter les traitements de ces derniers en arrière-plan afin d'éviter les fameux ANR qui nuisent grandement à l'expérience utilisateur mais également, et cela doit être un souci constant chez un développeur Android, de prendre grand soin de l'utilisation de la batterie en gérant au mieux la fréquence de rafraîchissement du Widget.

Sylvain Saurel – Ingénieur d'Etudes Java / JEE
sylvain.saurel@gmail.com

Créez votre propre navigateur avec Webkit et Qt

Webkit est un moteur de rendu HTML sur lequel se basent de nombreux navigateurs parmi lesquels Google Chrome et Safari. Intimement lié à la librairie Qt, Webkit forme avec elle un framework avec lequel vous pouvez donner libre cours à votre imagination.

WebKit est un moteur de rendu HTML qui dérive de KHTML, le moteur de rendu pour KDE et Konqueror. C'est un moteur de très grande qualité qui obtient un score de 100% au test ACID 3 depuis 2008, avec de grandes performances à l'exécution et une faible empreinte mémoire. Il est le moteur de rendu des navigateurs Safari et Google Chrome parmi les plus connus, mais aussi de nombreux autres. Il s'agit d'un projet open source, mais il n'est pas utilisable aisément en cet état. On utilisera ses intégrations dans des bibliothèques. Le choix est grand: Cocoa, wxWidgets, GTK+... C'est avec l'intégration de Webkit dans la librairie C++ Qt, appelée QtWebKit, que nous allons travailler aujourd'hui. QtWebKit est intéressant à plus d'un titre.

Il s'agit d'une intégration étroitement liée à WebKit car ce sont les développeurs de WebKit eux-mêmes qui l'ont réalisée. Ceci n'a d'ailleurs rien de surprenant car KDE est lui-même fondé sur Qt. Qt, et donc QtWebKit, est une librairie remarquablement portable : Windows, Mac, Linux et UNIX et certains mobiles. Qt est un environnement complet qui va très au-delà de la simple librairie graphique ou GUI. Qt offre en outre un moteur Javascript. Ainsi avec QtWebKit, il très simple de construire un navigateur complet, et cet article a pour but de donner au lecteur les éléments pour le faire. Il serait dommage de s'arrêter là cependant, car avec de l'imagination, il est possible de trouver des quantités d'applications à QtWebKit. On peut par exemple penser à s'orienter vers un navigateur expérimental doté d'outils personnalisés. On peut encore imaginer des applications de bureau portables et extensibles par du Javascript. Les possibilités n'ont en fait de limite que l'imagination. Avant de rentrer dans le vif du sujet, mentionnons encore une qualité de Qt qui est plutôt rare dans le monde de l'informatique : elle est agréable à programmer :-)

1 QT ET LE MÉCANISME SIGNAL/SLOT

Cet article part du principe que le lecteur dispose d'un SDK Qt, version 4.8 ou supérieure, installé sur sa machine. Ce SDK peut-être téléchargé depuis <http://qt.nokia.com/downloads/>. QtWebKit fait partie du SDK. Une des originalités de Qt est son mécanisme signal/slot. On rencontrera celui-ci non seulement dans l'interface graphique autour de notre navigateur, mais aussi dans la communication avec le code Javascript des pages Web. C'est pourquoi nous allons commencer par nous intéresser brièvement à lui. Ce mécanisme permet la communication entre les objets. Un objet émetteur émet un signal (sans rapport avec les signaux UNIX). Le ou les objets récepteurs dont les slots sont connectés au signal voient les méthodes des slots invoquées. Voici pour se fixer les idées, un exemple très basique, BasicSignalSlot, que vous trouverez, ainsi que tous les autres exemples de cet article, sur notre site.

```
// fichier émetteur.h
#include <QObject>
#include <string>

class Emetteur : public QObject
{
    Q_OBJECT
public:
    Emetteur(QObject *parent=0) : QObject(parent)
    {
    }

    void doEmettre(std::string s)
    {
        emit(emettre(s));
    }

signals:
    void emettre(std::string);
};

// fichier recepteur.h
#include <QObject>
#include <iostream>
#include <string>

class Recepteur : public QObject
{
    Q_OBJECT
public:
    Recepteur(QObject *parent=0) : QObject(parent)
    {
    }

public slots:
    void recevoir(std::string s)
    {
        std::cout << s << std::endl;
    }
};

// fichier main.cpp
#include «emetteur.h»
#include «recepteur.h»
#include <iostream>
#include <string>
```

```

int main(int argc, char **argv)
{
    Emetteur emetteur;
    Recepteur recepteur;

    QObject::connect(&emetteur, SIGNAL(emettre(std::string)),
                     &recepteur, SLOT(recevoir(std::string)) );

    emetteur.doEmettre("Programmez! - Abonnez-vous :-)");

    return 0;
}

```

La classe Emetteur déclare un signal et émet ce signal lorsque sa méthode doEmettre est invoquée. La classe Recepteur déclare un slot. On remarquera la concordance de types entre le signal et le slot. Le mécanisme signal/slot de Qt est type safe. On remarque que signaux et slots sont déclarés au moyen de mots clés qui sont des extensions à C++. Un fait qui est parfois reproché à QT, mais qui, en contrepartie, permet une programmation vraiment aisée. La fonction *main* de notre exemple, montre comment connecter un signal avec un slot. Il est possible de connecter plusieurs slots à un signal, mais il faut savoir que l'ordre d'invocation des slots n'est aucunement garanti. Enfin on remarque la macro `Q_OBJECT` au début de chaque classe. Cette macro est vue par l'outil qmake qui se chargera d'invoquer un Meta Object Compiler, ou MOC, lors de la construction du projet. C'est le MOC qui se charge de générer le code qui assure le fonctionnement du mécanisme signal/slot. Le programmeur n'a jamais besoin de voir ce qui se passe dans ce code ainsi généré, mais les curieux d'esprit trouveront intéressant d'y jeter un coup d'œil. Construire l'application à partir de nos trois fichiers `emetteur.h`, `recepteur.h` et `main.cpp` est tout ce qu'il y a de simple. D'abord on donne :

```

qmake -project
afin qu'un fichier de projet.pro soit généré, puis :
qmake
qui génère des fichiers moc_xxxx.cpp et un makefile. Enfin :
make

```

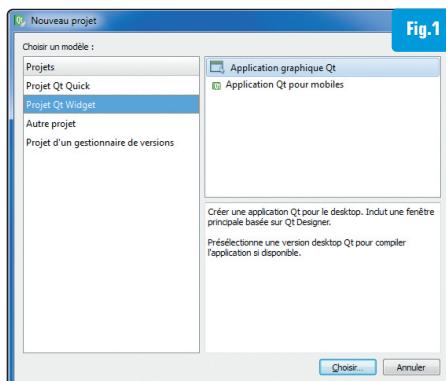
va construire le projet sous Linux ou sous Windows/Cygwin. Si on ne dispose pas de Cygwin, il est bon de savoir qu'il existe un plugin Qt

pour Visual Studio (des versions professionnelles). Sinon, l'outil QtCreator, dont nous parlerons plus loin, saura détecter et utiliser le compilateur d'un Visual Studio même en version Express.

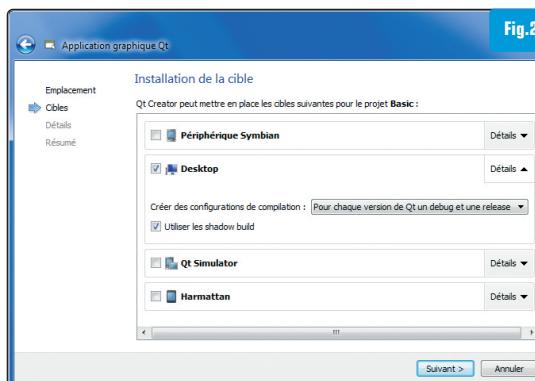
2 CONSTRUIRE UN NAVIGATEUR EN 2 MINUTES AVEC QTCREATOR

Le QtSDK comporte un environnement de développement intégré : QtCreator. Celui-ci, écrit en C++ et avec Qt, est disponible sur toutes les plateformes et nous allons maintenant l'utiliser. Sa prise en main est très aisée. Créons un premier projet. Pour cela, cliquez sur le bouton 'Créer un projet' que vous trouverez dans la page correspondant à l'onglet 'Accueil'. Créez un projet pour une application graphique Qt, comme illustré [Fig.1]. A la boîte de dialogue suivante, vous donnez le nom de votre projet. Nous avons appelé cet exemple 'Basic'. Dans la boîte de dialogue suivante, sélectionnez une application de type bureau ('Desktop') [Fig.2]. Si vous cochez la case 'Utiliser les shadow build', les fichiers de construction du projet (.obj, etc.) seront générés dans un répertoire distinct de celui de votre projet. Vient ensuite une boîte de dialogue dans laquelle vous devez préciser de quel widget QT doit dériver le widget principal de votre application. [Fig.3] Choisissez tout simplement un QWidget qui conviendra à un exemple rudimentaire. Le widget proposé par défaut, QMainWindow, sera celui qui conviendrait à l'interface d'un navigateur bien léché, avec barre de menu et barre d'outils par exemple. Nous ne verrons pas comment mettre en place de tels éléments avec Qt, afin de nous concentrer le plus rapidement possible sur Webkit. Le lecteur trouvera les renseignements nécessaires dans la documentation de QtCreator et de Qt.

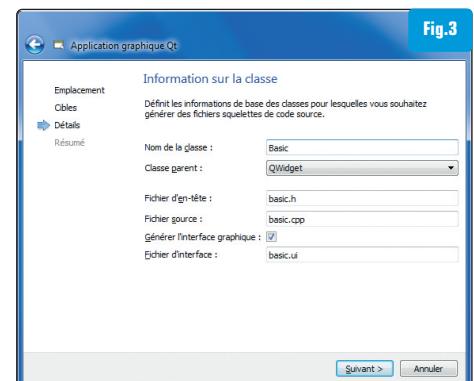
Continuez de traverser les boîtes de dialogues jusqu'à cliquer sur un bouton 'Terminer'. Vous arrivez alors dans l'environnement de QtCreator [Fig.4]. Là, dans l'arborescence du projet, cliquez sur le fichier de formulaire `basic.ui`. Ce fichier est un descriptif au format XML du widget principal de l'application. Le runtime de QT saura instancier le widget lors de l'exécution à partir de ce simple fichier. Cliquer sur ce fichier vous amène dans le concepteur visuel [Fig.5]. Une fois dans ce concepteur visuel, placez sur la fiche du widget et au moyen de glisser/déposer divers composants. Placez au moins un composant `QWebView`, qui, l'on s'en doute, permettra de visualiser les pages Web, et un composant `QLineEdit` qui sera la barre d'URL du navigateur. Dans notre exemple, nous avons baptisé ce composant `urlBar`. Les boutons '<' et '>' ne sont pour le moment là que pour la figuration :-) Nous voulons qu'une page Web s'affiche dans notre beau navigateur lorsqu'une URL est saisie dans la barre, c'est-à-dire lorsque l'utilisateur y presse la touche Return. Il s'agit donc de connecter le signal `returnPressed` du `QLineEdit`, mais à quel slot ?



Création de notre premier projet QtCreator



Choisissez une application de bureau comme cible de votre projet.



N'omettez pas de sélectionner la classe de Widget qui convient à votre application.

Ce slot aura pour nom 'naviguer'. Nous devons commencer par informer le concepteur visuel de son existence. Pour cela, faites un clic droit sur le widget principal, puis choisissez 'Modifier signaux/slots...' dans le menu qui est apparu. Dans le dialogue qui va suivre [Fig.6] et après un clic sur le bouton + vert, donnez naviguer(). Les parenthèses sont importantes. Puis validez. Vous êtes revenu dans le concepteur visuel, il ne vous reste plus qu'à ajouter une connexion entre le signal returnPressed du composant et notre slot dans l'éditeur de connexions [Fig.7]. Là encore, cette connexion, qui est décrite dans le fichier XML de l'interface, sera effectivement réalisée par le runtime au démarrage de l'application. C'est lui qui invoquera pour vous la méthode QObject::connect. Nous devons maintenant définir notre slot dans le code. Rien de plus simple. D'abord le fichier en-tête basic.h

```
#ifndef BASIC_H
#define BASIC_H

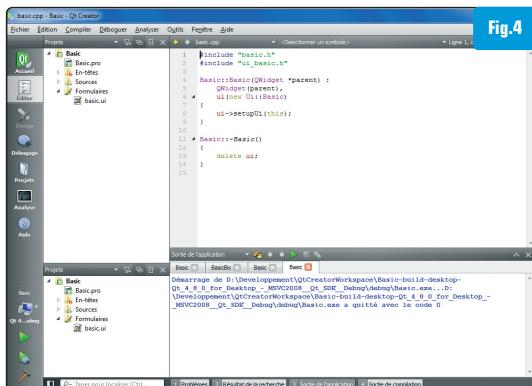
#include <QWidget>
namespace Ui {
class Basic;
}
class Basic : public QWidget
{
    Q_OBJECT
public:
    explicit Basic(QWidget *parent = 0);
    ~Basic();

private:
    Ui::Basic *ui;
protected slots:
    void naviguer();
};

#endif // BASIC_H
```

Puis le fichier basic.cpp

```
#include «basic.h»
#include «ui_basic.h»
Basic::Basic(QWidget *parent) :
    QWidget(parent),
    ui(new Ui::Basic)
{
    ui->setupUi(this);
}
```



Vue d'ensemble de QtCreator

```

}
Basic::~Basic()
{
    delete ui;
}
void Basic::naviguer()
{
    QUrl url(ui->urlBar->text());
    ui->webView->load(url);
}
```

Le constructeur, généré par QtCreator, montre que l'interface graphique de l'application sera construite automatiquement. Un pointeur (ui) sur le widget parent de cette interface est alors fourni. Notre méthode/slot naviguer récupère l'URL devant être visitée dans le QLineEdit et demande au composant QWebView, c'est-à-dire à WebKit de charger la page. C'est aussi simple que cela. Pour construire le projet, vous devez modifier, depuis l'éditeur de QtCreator, une ligne dans le fichier de projet .pro, afin que l'édition de liens se fasse correctement avec le module Webkit de Qt :

QT += core gui webkit

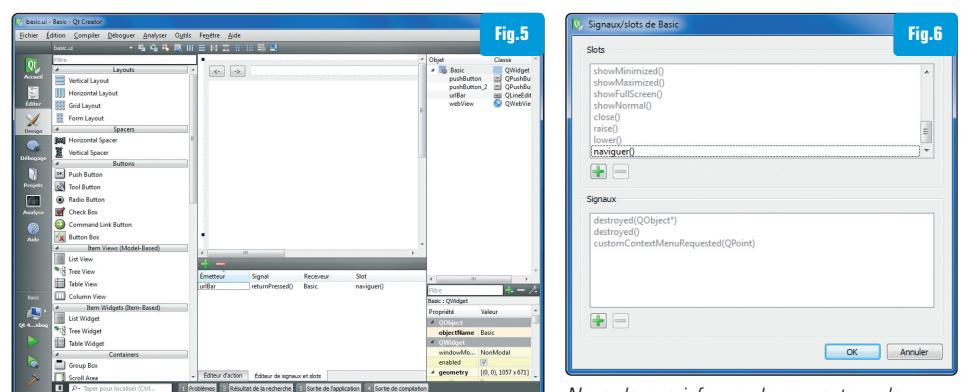
Lancez l'application, vous avez construit un navigateur :-) [Fig.8].

Le lecteur trouvera sur notre site un projet BasicBis qui utilise davantage le mécanisme signal/slot pour présenter un navigateur un peu plus élaboré. Une connexion au signal loadFinished émis par le composant QWebView permet de mettre à jour la barre d'URL. Les boutons fléchés permettent d'avancer ou reculer dans l'historique de navigation. Cette implémentation est naïve et n'a que pour but de suggérer toutes les possibilités offertes par Qt et WebKit. Ainsi, le lecteur qui souhaitera approfondir trouvera plus pertinent de tirer profit de la classe QWebHistory, par exemple.

3 QTWEBKIT ET LE DOM DES PAGES WEB

QtWebKit, l'intégration de Webkit dans Qt propose un jeu de classes très complet, bien entendu, documenté. Voici les classes les plus importantes :

Classe	Description
QWebView	Est le widget de rendu des pages
QWebPage	Encapsule une page Web
QWebFrame	Encapsule les frames dans une page Web. Un objet QWebFrame contient donc un ou plusieurs objets QWebFrame
QWebElement	Encapsule un élément du DOM de la page
QWebElement	Est une collection de QWebElement



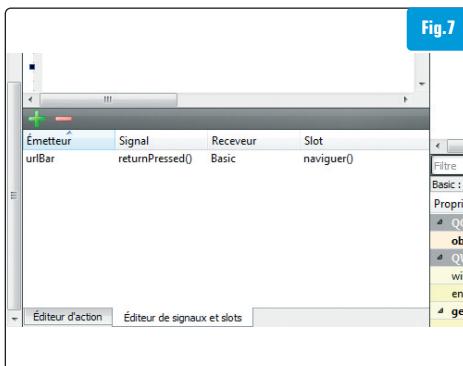
Nous devons informer le concepteur de l'existence de nos slots.

Faire un navigateur pour faire un navigateur n'est pas très intéressant en soi. Les navigateurs existants le font déjà très bien. Mais ce qui est plus intéressant, est de bricoler un outil qui pourrait satisfaire des besoins lors du développement des sites, par exemple pour la mise au point de la structure des pages. Supposons, par exemple, que nous soyons perdus dans la structure d'une page Web. Il est simple de décrire cette structure au moyen de QtWebKit. Ainsi notre programme d'exemple suivant BrowseDom (Code disponible sur notre site), écrit, dans un fichier texte, la structure du DOM d'une page dès que celle-ci est chargée [Fig.9]. Pour ce qui est de l'interface, notre application est copiée de l'exemple BasicBis. A la fin du code du slot updateUrlBar, nous récupérons le pointeur sur un objet QWebFrame qui encapsule le frame de la page. Ensuite la méthode documentElement de QWebFrame retourne le premier élément du DOM. C'est a priori, la balise HTML, à la racine du document. Ensuite nous appelons la méthode browseElements, qui parcourt tous les éléments du DOM récursivement. Le principe est simple. On commence par récupérer l'élément enfant de l'élément reçu en argument par la méthode. Si cet élément existe, on l'écrit dans le fichier, puis on invoque de nouveau la méthode pour descendre d'un niveau dans l'arborescence. Au retour de la méthode, donc au retour au niveau de départ, on traite en revanche le cas d'un éventuel élément parallèle, ou frère (Sibling). Et ainsi de suite, tout le DOM est parcouru.

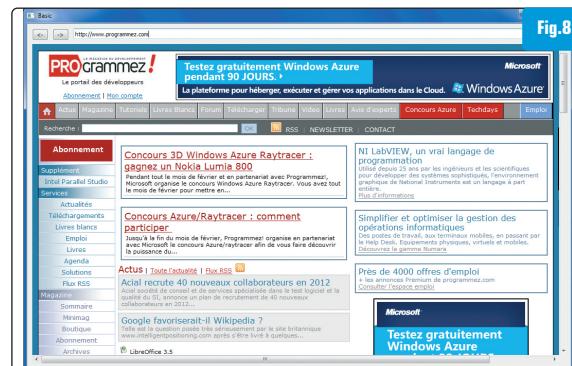
4 MODIFIER UN DOM

Notre exemple suivant, BidouilleDOM, montre avec quelle facilité il est possible de modifier un DOM. L'interface de notre exemple est construite sur la base de celles des exemples précédents. Nous avons ajouté un QLineEdit destiné à recevoir les éléments du DOM à modifier. Ceux-ci sont alors passés en couleur verte. Les éléments sont donnés comme on le ferait pour des sélecteurs (de classes, descendants, etc.) dans une feuille de style. Essayez de donner simplement DIV dans le QLineEdit et vous obtenez des contenus d'encadrés colorés en vert sur la page d'accueil de programmez.com. Par contre si vous donnez DIV A, ce sont tous les liens qui sont colorés en vert. Vous pouvez encore essayer un sélecteur de classe, comme ceci : DIV.box_bleu_autopromo. Le code pour obtenir ces résultats est très simple :

```
void BidouilleDom::bidouiller()
{
    QWebFrame *frame = ui->webView->page()->mainFrame();
    QWebElement document = frame->documentElement();
    QWebElementCollection elements =
        document.findAll(ui->lineEdit->text());
```



Connectez le signal returnPressed du QLineEdit à notre nouveau slot.



Un navigateur rudimentaire, mais qui fonctionne, réalisé avec quelques lignes de code seulement.

```
foreach (QWebElement element, elements)
    element.setAttribute("style", "color: #00ff00");
}
```

Comme dans l'exemple précédent, nous commençons par récupérer l'élément racine du DOM, encapsulé dans un objet QWebElement, dont une méthode, findAll, retourne une collection de tous les éléments enfant de l'élément pour lequel elle est invoquée et correspondant au sélecteur pour laquelle elle est invoquée. Il suffit ensuite de modifier le style de chacun des éléments de la collection. foreach est ici une macro de Qt.

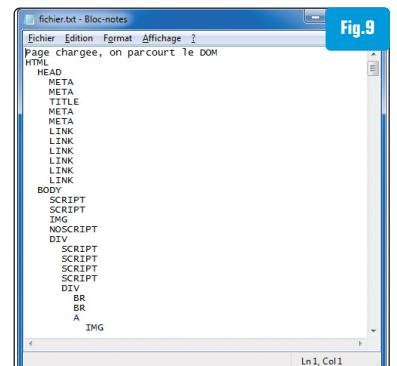
5 TRAVAILLER AVEC JAVASCRIPT

Il est possible d'exécuter le code Javascript d'une page du côté de Qt. Voici les parties importantes de notre exemple ExecuteJavascript qui fait cela : [Fig.10].

```
#include <QVariant>
#include <QtWebKit>
#include <QWebPage>
#include <QWebFrame>

const QString page_html =
<<html><
<<head><
<<meta http-equiv=\Content-Type\ content=\text/html; charset=ISO-8859-1\>><
<<title>Programmez!</title><
<<script><
    function mafonction(texte)<
    {<
        alert(texte);<
    }<
    function moncalcul()<
    {<
        return 1+2;}<
    }<
</script><
</head><
<<body><
    <h1><strong>PROGRAMMEZ !</strong></h1><
</body><
</html><
```

```
void ExecuteJavascript::chargerPage()
```



La structure de la page d'accueil de programmez.com

```

{
    ui->webView->setHtml(page_html);
}

void ExecuteJavascript::hello()
{
    QWebView* webView = ui->webView;
    QWebPage* page = webView->page();
    QWebFrame* frame = page->mainFrame();
    frame->evaluateJavaScript(
        «mafonction('Abonnez-vous ! :->'); null»);
}

void ExecuteJavascript::calculer()
{
    QWebView* webView = ui->webView;
    QWebPage* page = webView->page();
    QWebFrame* frame = page->mainFrame();
    QVariant resultat = frame->evaluateJavaScript(«moncalcul();»);
    ui->lineEdit->setText(resultat.toString());
}

void ExecuteJavascript::calculerPourRien()
{
    QWebView* webView = ui->webView;
    QWebPage* page = webView->page();
    QWebFrame* frame = page->mainFrame();
    QVariant resultat = frame->evaluateJavaScript(«moncalcul(); null»);
    ui->lineEdit->setText(resultat.toString());
}

```

Cet exemple montre aussi, qu'il est possible de constituer une page HTML de toutes pièces à partir d'une chaîne. Notre méthode hello appelle la fonction Javascript mafonction de la page via la méthode evaluateJavascript d'un objet QWebFrame. Ce qui est intéressant dans cet appel est l'ajout de null à la chaîne transmise en argument, alors que :

```
frame->evaluateJavaScript(«mafonction('Abonnez-vous ! :->');»);

```

suffit pour que cela fonctionne. Ajouter null améliore les performances. Curieusement, ce point n'est pas mentionné dans la documentation. Toute invocation d'une fonction Javascript voit son résultat retourné du côté de Qt. Si on ajoute null, il n'y a plus de valeur de retour, comme on le constate en exécutant les méthodes calculer et calculerPourRien.



Exécution de code Javascript avec le moteur de Qt.

6 INJECTER DU CODE JAVASCRIPT

Exécuter le code Javascript présent dans une page est bien, mais il est beaucoup plus intéressant d'y ajouter du code afin de pouvoir agir sur le document. A ce titre, injecter le code de JQuery, par exemple, serait une application toute trouvée. Notre exemple InjecterJavascript (Code disponible sur notre site) présente une page dans laquelle, au départ, un clic sur le mot Programmez! est sans effet. Après injection de code, une alerte s'ouvre. [Fig.11].

L'opération est en fait toute simple. Il suffit de placer le code Javascript dans une chaîne. Ce code peut bien sûr être lu depuis un fichier ou une ressource, pour constituer la chaîne. Il ne reste alors plus qu'à évaluer cette chaîne de code pour que celui-ci se retrouve visible du côté de la page Web.

7 LE PONT QT / JAVASCRIPT

Si nous venons de voir qu'il est finalement très facile de se construire un navigateur à partir des exemples des paragraphes précédents, nous allons maintenant aborder une technologie remarquablement puissante que QtWebKit met à notre disposition. Et c'est cette technologie qui vous permettra de donner libre cours à votre imagination pour créer des applications originales. Cette technologie s'appelle le QtWebKit Bridge. Comme le nom le suggère fortement, elle réalise un pont entre le code C++ du côté QT et le code Javascript des pages Web. Le QtWebKit Bridge tire son origine dans la technologie QtScript qui rend les applications Qt «scriptables». Une grande part des fonctionnalités de QtScript se retrouve donc dans le QtWebKit-Bridge, mais pas toutes. Le QtWebKit Bridge étend l'environnement Javascript pour lui permettre d'accéder à des instances d'objets natifs, pourvu que ceux-ci soient des QObjects. Le mécanisme s'appuie sur les possibilités d'introspection du modèle objet de Qt. En particulier, ce mécanisme assure un mapping entre les propriétés des QObject et les propriétés Javascript. En outre, nous retrouvons le mécanisme signal/slot dans cet environnement, ce qui ouvre des possibilités très intéressantes. Ainsi du code C++ peut connecter un signal C++ à une fonction Javascript qui se comporte donc comme un slot, mais sans que l'objet natif soit exposé côté script. Réciproquement, un script peut établir des connexions signal/slot entre des objets natifs prédéfinis, pourvu que ceux-ci soient exposés uniquement côté script par l'application. Dans ce cas, les slots sont écrits en C++. Mais les connexions sont établies dynamiquement du côté script. Autre possibilité, un script peut définir des handlers, autrement dit des slots écrits en Javascript et établir la connexion entre



Injection de code Javascript dans une page Web.

des signaux émis par des objets natifs, et les slots en Javascript. Là encore les objets natifs doivent être exposés côté script par l'application. Notre prochain exemple utilisera les deux dernières possibilités.

8 DES APPLICATIONS ORIGINALES

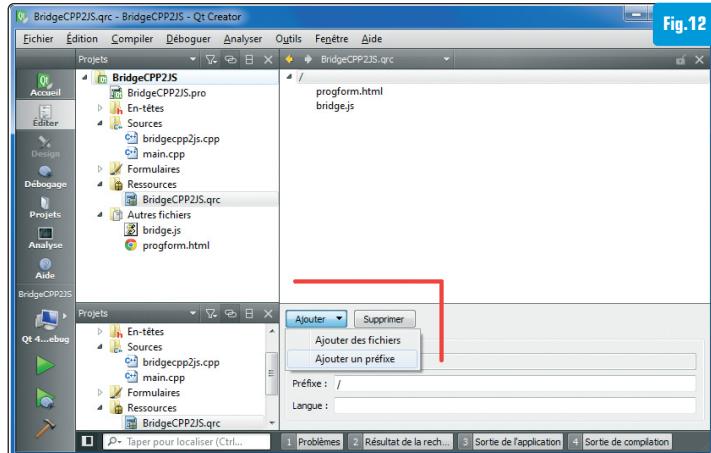
Le QtWebKit Bridge peut trouver son emploi dans des applications natives de bureau qui tirent profit de divers outils Web tout prêts, players, plugins, etc. Grâce au QtWebKit Bridge, une telle application peut garder son corps principal en C++. Ce qui signifie performance pour les fonctionnalités un peu gourmandes en ressources, et ce qui signifie aussi possibilité d'accéder aux fonctionnalités du système d'exploitation sous-jacent. Un autre domaine d'emploi est le client léger qui expose à l'utilisateur une interface totalement Web, composée de HTML, Javascript et CSS. Ce design est très à la mode en ce moment. Là encore un backend de l'application écrit en C++ assurera les performances.

9 LES RESSOURCES QT

Nous allons maintenant passer à notre exemple BridgeCPP2JS. Cet exemple charge une page HTML dans un composant QWebView, puis charge du code Javascript. Il n'est pas viable, pour une application réelle, de placer de tels éléments en dur dans le code, dans des chaînes QString comme nous l'avons fait dans les exemples précédents. Ces éléments doivent être placés dans une ressource. Pour obtenir cela, faites un clic droit sur le nom de votre projet dans QtCreator, puis sélectionnez 'Ajouter nouveau...'. Choisissez alors 'Fichiers de ressource Qt', puis suivez les boîtes de dialogues. Une fois la ressource créée, la première chose à faire est d'y ajouter un premier préfix. Pour cela cliquez sur le nom de votre fichier de ressource dans l'arborescence de l'éditeur, pour faire apparaître l'éditeur de ressources, puis dans cet éditeur, cliquez sur le bouton 'Ajouter'. Choisissez un préfix minimaliste, par exemple '/'. Une fois le préfix ajouté, il vous sera possible d'incorporer des fichiers à votre ressource, comme illustré [Fig.12].

10 MISE EN ŒUVRE DE QT WEBKIT

Notre application d'exemple charge donc les contenus HTML et Javascript depuis une ressource. L'utilisateur peut alors cliquer sur le bouton Go du formulaire rendu dans le composant QWebView. Le code de soumission de ce formulaire n'est pas écrit en Javascript,



Mise en place d'une ressource Qt avec QtCreator.

mais en C++ et il est un slot C++ qui sera invoqué grâce au pont. Ce code côté C++ lit les valeurs des champs de formulaires, qui sont donc côté pages Web, puis place la concaténation des deux valeurs lues dans un composant QLineEdit qui est donc côté C++. A ce moment vous verrez le contenu d'un second composant QLineEdit être modifié [Fig.13]. Cette modification n'est pas réalisée côté C++, mais côté Javascript, ce côté recevant un signal d'un composant natif C++ et y réagissant. Examinons maintenant le code de cette application. Voici d'abord le code de la page HTML :

```
<html>
<body>
<h1>Programmez ! Le magazine du développement </h1>
<form onsubmit=>BridgeCPP2JS.submit()>
Conseil:
<input type=>text name=>prog
id=>programmez value=>Programmez!<br />
Meilleur conseil:
<input type=>text name=>abo
id=>abonnez value=>Abonnez-vous ! :-)<br />
<input type=>submit value=>Go !>
</form>
</body>
</html>
```

Un formulaire très classique donc, avec des valeurs de champs pré-remplis. Voici maintenant le code Javascript :

```
// fichier bridge.js
function lireLineEdit()
{
    texte = lineEdit.text;
    alert(texte);
    lineEdit2.setText(`sans tarder`);
}
function connecter()
{
    lineEdit.textChanged.connect(lireLineEdit);
}
connecter();
```



Communication bi-directionnelle C++/Javascript grâce à QtWebKit Bridge.

Il s'agit d'un code Javascript injecté à la page HTML, comme nous l'avons montré à notre exemple précédent. La fonction 'connecter' est appelée lors de l'injection, et elle met en place une connexion signal/slot. Le signal est textChanged, du composant de nom lineEdit. Ce composant est accessible et son nom est connu car à ce stade, il est déjà exposé. Nous verrons plus loin comment cela est réalisé. Cette approche pour se connecter fonctionne avec tout type de composant. On peut écrire une sorte de pseudo code générique pour fixer les idées :

```
nomComposant.quelquechoseChanged.connect(unefonctionjavascript);
```

Ici la fonction Javascript lireLineEdit devient de facto un slot. Dans le corps de cette fonction, nous lisons la valeur du texte dans le QLineEdit, dont toutes les fonctionnalités sont, comme on le voit ici, pleinement accessibles. Nous affichons cette valeur dans une alerte, puis nous modifions la valeur du second QLineEdit, également exposé, donc également totalement accessible. Regardons maintenant le côté C++ de l'application. Tout d'abord, l'en-tête du widget principal de l'application :

```
// fichier bridgecpp2js.h
#ifndef BRIGDECPP2JS_H
#define BRIGDECPP2JS_H

#include <QWidget>
namespace Ui {
class BridgeCPP2JS;
}

class BridgeCPP2JS : public QWidget
{
    Q_OBJECT

public:
    explicit BridgeCPP2JS(QWidget *parent = 0);
    ~BridgeCPP2JS();
public slots:
    void submit();
    void initialiserBridge();
protected slots:
    void go();

private:
    Ui::BridgeCPP2JS *ui;
};

#endif // BRIGDECPP2JS_H
```

Nous remarquons dans cet en-tête la déclaration du slot submit, qui ne diffère en rien d'un slot classique, bien qu'il soit destiné à être invoqué depuis le code Javascript. Voici maintenant le reste du code de l'application :

```
// fichier bridgecpp2js.h
#include «bridgecpp2js.h»
#include «ui_bridgecpp2js.h»
#include <QtWebKit>
#include <QWebPage>
#include <QWebFrame>
#include <QFile>
```

```
BridgeCPP2JS::BridgeCPP2JS(QWidget *parent) :
    QWidget(parent),
    ui(new Ui::BridgeCPP2JS)
{
    ui->setupUi(this);
    connect(ui->webView->page()->mainFrame(),
            SIGNAL(javaScriptWindowObjectCleared()),
            this, SLOT(initialiserBridge()));
}
BridgeCPP2JS::~BridgeCPP2JS()
{
    delete ui;
}

void BridgeCPP2JS::go()
{
    ui->webView->setUrl(QUrl("qrc:/progform.html"));
}
void BridgeCPP2JS::submit()
{
    QWebFrame* frame = ui->webView->page()->mainFrame();
    QWebElement prog = frame->findFirstElement("#programmez");
    QWebElement abo = frame->findFirstElement("#abonnez");
    QString s1 = prog.evaluateJavaScript("this.value").toString();
    QString s2 = abo.evaluateJavaScript("this.value").toString();
    ui->lineEdit->setText(s1 + " " + s2);
}

void BridgeCPP2JS::initialiserBridge()
{
    QWebFrame* frame = ui->webView->page()->mainFrame();
    frame->addToJavaScriptWindowObject("BridgeCPP2JS", this);
    frame->addToJavaScriptWindowObject("lineEdit", ui->lineEdit);
    frame->addToJavaScriptWindowObject("lineEdit2", ui->lineEdit2);
    QFile file;
    file.setFileName(":/bridge.js");
    file.open(QIODevice::ReadOnly);
    QString codeJavascript = file.readAll();
    file.close();
    frame->evaluateJavaScript(codeJavascript);
}
```

Le code du constructeur du widget met en place un mécanisme qui assure que les Widgets C++ continueront d'être exposés si QWebView voit sa page rechargée. Ce code, toujours le même, est requis dans ce genre d'applications. La méthode go charge la page HTML depuis la ressource. Bien remarquer que nous retrouvons notre préfix / dans le nom de la ressource à charger. Vient ensuite la méthode submit. Nous y retrouvons des savoir-faire vus dans les exemples précédents, c'est-à-dire trouver des éléments du DOM bien précis, et évaluer du code Javascript. Vient enfin la méthode initialiserBridge. Cette méthode expose, via la méthode addToJavaScriptWindowObject de QWebFrame, trois objets C++, le widget principal de l'application, et les deux composants QLineEdit. Le reste du code lit le code Javascript dans le fichier de ressource (bien remarquer le préfix / à nouveau) et l'injecte. Aucune difficulté à ce niveau. Et finalement, avec peu d'un code très clair, nous obtenons un bien beau résultat.

11 EMBARQUER UN WIDGET QT DANS UNE PAGE WEB

L'idée est trop tentante et surtout tout à fait réalisable. En effet, le moteur de rendu QtWebKit utilise lui-même les widgets Qt pour afficher les pages. Et comme les Widgets Qt peuvent être des conteneurs (des parents) d'autres widgets, une page Web peut devenir le parent d'un autre Widget. La documentation de QtWebKit propose une méthode pour ce faire qui est assez lourde, qui demande à exposer les widgets à embarquer, et à utiliser de surcroît une classe fabrique. Je vous propose ici une autre approche plus légère et basée sur l'héritage, pour embarquer un composant QCalendar dans une page. Pour parvenir à ce résultat nous commençons par créer une page Web qui contient une balise object, comme ceci :

```
<html>
  <head>
    <title>Demo de widget Qt embarqué</title>
  </head>
  <body>
    <h1>Programmez! chaque jour :-)</h1>
    <object type="application/x-qt-plugin"
      classid="EmbeddedCalendar" name="calendar">
      height=300 width=300</object>
    <script>
      calendar.setGridVisible(true);
      calendar.setCurrentPage(2012, 7);
    </script>
  </body>
</html>
```

Le point très important est le type Mime de la balise object : application/x-qt-plugin. Notre approche ne fonctionne qu'avec ce type Mime. Tandis que l'approche de la documentation que nous avons évoquée fonctionne avec des types Mime divers et variés. Dans le code ci-dessous on voit (balise script) que le widget sera exposé avec ses fonctionnalités accessibles, tout comme dans l'exemple précédent. Nous ne pouvons pas embarquer QCalendar directement, mais seulement un objet qui en dérive, comme ceci :

```
// fichier embeddedcalendar.h

#ifndef EMBEDDEDCALENDAR_H
#define EMBEDDEDCALENDAR_H
#include <QMetaType>
#include <QCalendarWidget>
class EmbeddedCalendar : public QCalendarWidget
{
  Q_OBJECT
public:
  EmbeddedCalendar(QWidget* parent = 0);
  // Q_DECLARE_METATYPE demande
  // un constructeur de copie. Même si ce
  // constructeur de copie ne fait rien :-)
  EmbeddedCalendar(const EmbeddedCalendar& copy);
};

Q_DECLARE_METATYPE(EmbeddedCalendar)
#endif // EMBEDDEDCALENDAR_H
```

Deux points sont à souligner. D'abord notre widget embarqué doit avoir un constructeur de copie, même si celui-ci ne fait rien. Le constructeur

normal ne fait rien non plus d'ailleurs :-) Enfin, cette dérivation est nécessaire, car elle nous permet de placer la macro Q_DECLARE_METATYPE. Sans la déclaration d'un métatype ainsi faite, le runtime ne saura pas exposer notre widget correctement au côté de Javascript. Nous devons ensuite dériver QWebPage comme ceci :

```
// fichier embeddingwebpage.h

#ifndef EMBEDDINGWEBPAGE_H
#define EMBEDDINGWEBPAGE_H
#include <QWebPage>

class EmbeddingWebPage: public QWebPage
{
  Q_OBJECT
protected:
  QObject* createPlugin(
    const QString& classid,
    const QUrl& url,
    const QStringList& paramNames,
    const QStringList& paramValues);
public:
  EmbeddingWebPage(QObject *parent = 0);
};

#endif // EMBEDDINGWEBPAGE_H
```

Le but étant de pouvoir redéfinir la méthode virtuelle createPlugin. Cette méthode est automatiquement invoquée par le runtime lorsque le moteur de rendu WebKit rencontre une balise object dont le type Mime est application/x-qt-plugin. Voici le corps de cette méthode :

```
// fichier embeddingwebpage.cpp
QObject *EmbeddingWebPage::createPlugin(
  const QString &classid,
  const QUrl&,
  const QStringList&,
  const QStringList&)
{
  QObject *result = 0;
  result = reinterpret_cast<QObject*>(
    QMetaType::construct(
      QMetaType::type(
        classid.toLatin1())));
  return result;
}
```

Il ne reste plus qu'à utiliser notre composant dérivé de QWebPage au démarrage, ce qui est tout simple :

```
// fichier embedding.cpp
void Embedding::go()
{
  EmbeddingWebPage* ewp = new EmbeddingWebPage(ui->webView);
  ui->webView->setPage(ewp);
  QFile page(<:/page.html>);
  page.open(QIODevice::ReadOnly);
  QString html = page.readAll();
  ui->webView->setHtml(html);
}
```

Frédéric Mazué - fmazue@programmez.com

JQueryMobile, quelques lignes pour une interface mobile

De nos jours, les mobiles communicants sont partout et on fait tout avec : naviguer sur Internet, consulter l'e-mail, poster sur Facebook, etc. Avoir un site Web utilisable sur son mobile n'est plus un luxe, spécialement si on peut l'adapter sans trop d'efforts.

Dans cet article nous allons construire un site Web optimisé pour les mobiles, composé de deux pages : une page d'accueil et une page de contact sous forme d'accordéon.

Si vous voulez éviter de partir de zéro pour développer une interface mobile pour le web, le meilleur moyen est d'utiliser un framework, composé normalement d'un mélange de Javascript, CSS, HTML, et des images. JQueryMobile se distingue du lot des frameworks Web-mobile, car il possède plusieurs atouts :

- Une utilisation simple.
- Une belle, interactive, et native like interface, qui s'adapte automatiquement selon le navigateur (ordinateur, mobile, tablette)
- Compatible avec la plupart des smartphones et optimisé pour écrans tactiles.
- Basé sur HTML5, jQuery et CSS
- Open Source

> Sites

<http://jqmymobile.com> et <http://jquery.com>

> Versions utilisées

JQueryMobile 1.0.1 et jQuery 1.7.1

> Affichage

Un émulateur Android 2.2 sous Eclipse 3.7 (Indigo SR2).

> Hello World

Pour commencer le plus simplement possible, nous allons afficher Hello World dans le mobile (émulateur), pour constater la simplicité de développement avec jQueryMobile. Nous avons d'abord une page HTML, la plus simple (HTML5), comme suit :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8"/>

    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Hello World jQueryMobile</title>
  </head>

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

Nous allons ajouter des instructions pour inclure jQueryMobile.

Il y a deux moyens d'inclure jQueryMobile :

1 Télécharger les paquetages sur le site de jQueryMobile, et jQuery. Décompresser, mettre le dossier jQueryMobile résultant dans la racine de votre projet Web. Mettre enfin des instructions pour référencer la bibliothèque dans la page Web. On ajoute aussi une référence vers le fichier .js de jQuery.

Note :

Quelques lignes sont coupées par le formatage du texte. Les instructions sont, bien entendu, sur la même ligne, par exemple, le contenu de l'attribut « href » de « link », et « src » de « script » sont chacun sur une seule ligne.

viewport dans la balise meta est l'endroit où les pages sont affichées. Cette instruction est nécessaire, spécialement pour iOS :

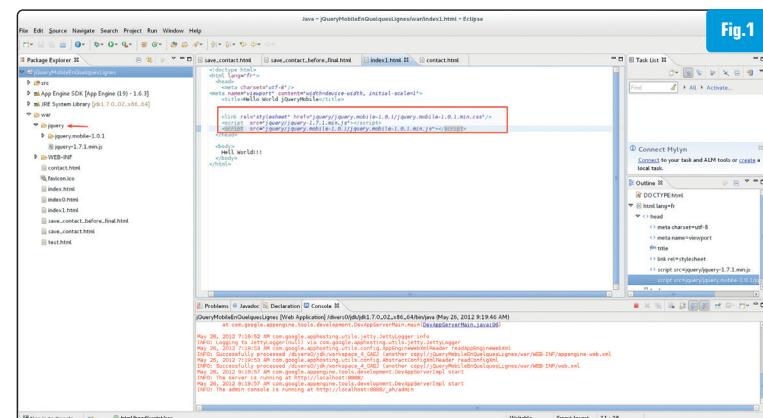
```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8"/>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Hello World jQueryMobile</title>

    <link rel="stylesheet" href="jquery/jquery.mobile-1.0.1/jquery.mobile-1.0.1.min.css"/>
    <script src="jquery/jquery-1.7.1.min.js"></script>
    <script src="jquery/jquery.mobile-1.0.1/jquery.mobile-1.0.1.min.js"></script>
  </head>

  <body>
    Hello World!!!
  </body>
</html>
```

[Fig.1]

2 Référencer directement les fichiers désirés sur le site Web de jQuery ou CDN-hosted (content delivery network) :



```

<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8"/>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Hello World jQueryMobile</title>

    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.css" />
    <script src="http://code.jquery.com/jquery-1.6.4.min.js"></script>
    <script src="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.js"></script>

  </head>

  <body>

  </body>
</html>

```

> Ajoutez des déclarations pour styler la page

jQueryMobile utilise l'attribut «**data-***» dans des balises HTML5 (div ou autres) pour styler des pages Web. L'attribut **data-role** notamment permet de définir le rôle de chaque élément d'une page Web et donner un look précis à cet élément. **data-role=>page**» est utilisé pour désigner une page, **data-role=>header**», pour l'en-tête, **data-role=>footer**» pour le pied d'une page, et **data-role=>content**» pour le contenu. L'ajout de ces attributs permet à jQueryMobile de styler correctement et automatiquement des pages Web.

Note :

jQueryMobile utilise des sections **<section>** pour assembler des pages Web dans une seule page Web. Quand vous utilisez des pages séparées pour vos pages web, jQueryMobile se charge du reste.

index.html

Notez que la page a été stylée automatiquement, selon les attributs data, avec header, footer et contenu. J'ai juste ajouté un petit texte pour la démonstration.

```

<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8"/>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Hello World jQueryMobile</title>

    <link rel="stylesheet" href="jquery/jquery.mobile-1.0.1/jquery.mobile-1.0.1.min.css" />
    <script src="jquery/jquery-1.7.1.min.js"></script>
    <script src="jquery/jquery.mobile-1.0.1/jquery.mobile-1.0.1.min.js"></script>
  </head>

  <body>
    <section id="home" data-role="page">

      <header data-role="header">
        <h1>Article jQM</h1>
      </header>

      <div data-role="content"><h1>Hell World!!!</h1>
        De nos jours les mobiles communicants sont partout et on fait tout avec;
        <br>naviguer sur Internet, consulter l'e-mail, poster sur Facebook, etc.<br>
        Avoir un site Web utilisable aussi sur le mobile n'est plus un luxe, <br>
        spécialement si on peut adapter le site sans beaucoup d'efforts.
      </div>

      <footer data-role="footer">
        <h4>Pour Programmez magazine</h4>
      </footer>

    </section>
  </body>
</html>

```

```

</header>

<div data-role="content"><h1>Hell World!!!</h1>
  De nos jours les mobiles communicants sont partout et on fait tout avec;
  <br>naviguer sur Internet, consulter l'e-mail, poster sur Facebook, etc.<br>
  Avoir un site Web utilisable aussi sur le mobile n'est plus un luxe, <br>
  spécialement si on peut adapter le site sans beaucoup d'efforts.
</div>

<footer data-role="footer">
  <h4>Pour Programmez magazine</h4>
</footer>

</section>
</body>
</html>

```

[Fig.2]

> Avoir l'en-tête et le pied de page à une position fixe

Pour avoir le pied de page et/ou le header, toujours dans la même position, on utilise l'attribut **data-position=>fixed**, tout simplement :

```

<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8"/>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Hello World jQueryMobile</title>

    <link rel="stylesheet" href="jquery/jquery.mobile-1.0.1/jquery.mobile-1.0.1.min.css" />
    <script src="jquery/jquery-1.7.1.min.js"></script>
    <script src="jquery/jquery.mobile-1.0.1/jquery.mobile-1.0.1.min.js"></script>
  </head>

  <body>
    <section id="home" data-role="page">

      <header data-role="header" data-position="fixed">
        <h1>Article jQM</h1>
      </header>

      <div data-role="content"><h1>Hell World!!!</h1>
        De nos jours les mobiles communicants sont partout et on fait tout avec;
        <br>naviguer sur Internet, consulter l'e-mail, poster sur Facebook, etc.<br>
        Avoir un site Web utilisable aussi sur le mobile n'est plus un luxe, <br>
        spécialement si on peut adapter le site sans beaucoup d'efforts.
      </div>

      <footer data-role="footer" data-position="fixed">
        <h4>Pour Programmez magazine</h4>
      </footer>

    </section>
  </body>
</html>

```

```

<h4>Pour Programmez magazine</h4>
</footer>

</section>
</body>
</html>

```

Note :

On applique **data-position=>fixed** sur l'en-tête pour l'empêcher de bouger quand on fait défiler la page.

> Ajoutez des boutons pour naviguer

Il est tout à fait possible d'ajouter des liens dans une balise `<div>` avec le rôle de bar de navigation **data-role=>navbar** et laisser jQueryMobile formater les liens sous forme de boutons, mais nous voulons faire des choses plus complexes en ajoutant des boutons qui ressemblent aux boutons d'un vrai mobile.

Commençons d'abord par le plus simple :

```

<!doctype html>
<html lang=>fr>
<head>
  <meta charset=>utf-8/>
<meta name=>viewport content=>width=device-width, initial-scale=1>
  <title>Hello World jQueryMobile</title>

  <link rel=>stylesheet href=>jquery/jquery.mobile-1.0.1/
jquery.mobile-1.0.1.min.css/>
  <script src=>jquery/jquery-1.7.1.min.js></script>
  <script src=>jquery/jquery.mobile-1.0.1/jquery.mobile-1.
0.1.min.js></script>
</head>

<body>
<section id=>home data-role=>page>

<header data-role=>header>
  <h1>Article jQM</h1>
</header>

```

<div data-role=>content><h1>Hell World!!!</h1>
De nos jours les mobiles communicants sont partout et on fait tout avec ; naviguer sur Internet, consulter l'e-mail, poster sur Facebook, etc. Avoir un site Web utilisable aussi sur le mobile n'est plus un luxe, spécialement si on peut adapter le site sans beaucoup d'efforts.

```

<footer data-role=>footer data-position=>fixed>
<div data-role=>navbar data-id=>nav>
  <a href=>index.html>Home</a>
  <a href=>contact.html>Contact</a>
</div>
<h4>Pour Programmez magazine</h4>
</footer>

</section>
</body>
</html>

```

[Fig.3]

> Des boutons, des vrais

Pour ajouter des boutons ressemblant aux boutons d'une application native mobile, on commence par ajouter l'attribut suivant au pied de la page `<footer>`, **class=>ui-bar**.

Remplacez **data-role=>navbar** ajouté avant, par **data-role=>controlgroup** et ajoutez dans la même balise `<div>` un attribut **data-type=>horizontal**. Ajoutez ensuite un attribut **data-icon** pour ajouter une icône parmi celles qui viennent avec jQueryMobile. Pour indiquer la page courante on ajoute au lien correspondant **class=>ui-btn-active ui-state-persist**.

Ajoutez enfin **data-role=>button** dans chaque lien :

```

<!doctype html>
<html lang=>fr>
<head>
  <meta charset=>utf-8/>
<meta name=>viewport content=>width=device-width, initial-scale=1>
  <title>Hello World jQueryMobile</title>

```



```

<link rel="stylesheet" href="jquery/jquery.mobile-1.0.1/
jquery.mobile-1.0.1.min.css"/>
<script src="jquery/jquery-1.7.1.min.js"></script>
<script src="jquery/jquery.mobile-1.0.1/jquery.mobile-1.
0.1.min.js"></script>
</head>

<body>
<section id="home" data-role="page">

<header data-role="header">
<h1>Article jQM</h1>
</header>

<div data-role="content"><h1>Hell World!!!</h1>
De nos jours les mobiles communicants sont partout et on
fait tout avec;
<br>naviguer sur Internet, consulter l'e-mail, poster sur
Facebook, etc.<br>
Avoir un site Web utilisable aussi sur le mobile n'est
plus un luxe, <br>
spécialement si on peut adapter le site sans beaucoup d'efforts.
</div>

<footer data-role="footer" data-position="fixed" data-id="nav"
class="ui-bar">
<div data-role="controlgroup" data-type="horizontal">
<a href="index.html" data-icon="home" data-role="button"
class="ui-btn-active ui-state-persist">Home</a>
<a href="contact.html" data-icon="info" data-role="button">Contact</a>
</div>
<h4>Pour Programmez magazine</h4>
</footer>
</section>
</body>
</html>

```

[fig.4]

Ces couleurs sont sombres. J'utilise des thèmes

jQueryMobile a cinq groupes de couleurs prédéfinies, appelés aussi swatches. Si les couleurs par défaut ne vous conviennent pas, il suffit d'insérer l'attribut **data-theme** dans une balise en lui donnant une valeur de «a» à «e». Il est même possible de composer vous-même votre propre style sur le site de jQueryMobile.

```

<!doctype html>
<html lang="fr">
<head>
<meta charset="utf-8"/>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Hello World jQueryMobile</title>

```

```

<link rel="stylesheet" href="jquery/jquery.mobile-1.0.1/
jquery.mobile-1.0.1.min.css"/>
<script src="jquery/jquery-1.7.1.min.js"></script>
<script src="jquery/jquery.mobile-1.0.1/jquery.mobile-1.
0.1.min.js"></script>
</head>

<body>
<section id="home" data-role="page">

<header data-role="header">
<h1>Article jQM</h1>
</header>

<div data-role="content"><h1>Hell World!!!</h1>
De nos jours les mobiles communicants sont partout et on
fait tout avec;
<br>naviguer sur Internet, consulter l'e-mail, poster sur
Facebook, etc.<br>
Avoir un site Web utilisable aussi sur le mobile n'est
plus un luxe, <br>
spécialement si on peut adapter le site sans beaucoup d'efforts.
</div>

<footer data-role="footer" data-position="fixed" data-id="nav"
class="ui-bar">
<div data-role="controlgroup" data-type="horizontal">
<a href="index.html" data-icon="home" data-role="button"
class="ui-btn-active ui-state-persist">Home</a>
<a href="contact.html" data-icon="info" data-role="button">Contact</a>
</div>
<h4>Pour Programmez magazine</h4>
</footer>
</section>
</body>
</html>

```

> Ajoutez une nouvelle page pour le contenu

Enregistrez sous la page index.html, nommez le contact.html. Supprimez la partie entre `<div data-role="content">` et `</div>`. Nous voulons garder l'en-tête et le pied de la page.

> Une balise `<div> collapsible`

Une balise dont le contenu peut être caché. On peut l'ouvrir pour voir les articles cachés sous le titre ou la fermer pour cacher tout (sauf son titre) :

```

<!doctype html>
<html lang="fr">
<head>
<meta charset="utf-8"/>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Hello World jQueryMobile</title>

```

```

<link rel="stylesheet" href="jquery/jquery.mobile-1.0.1/
jquery.mobile-1.0.1.min.css"/>
<script src="jquery/jquery-1.7.1.min.js"></script>
<script src="jquery/jquery.mobile-1.0.1/jquery.mobile-1.
0.1.min.js"></script>
</head>

<body>
<section id="contact" data-role="page">

<header data-role="header">
<h1>Article jQM</h1>
</header>

<div data-role="content">

<div class="collapsible" data-role="collapsible" data-content
-theme="a" data-theme="b">
<h3>Contact</h3>
<nav data-role="navbar">
<a href="">E-mail</a>
<a href="">Tél</a>
</nav>
</div>

</div>

<footer data-role="footer" data-position="fixed" data-id
="nav" class="ui-bar" data-theme="e">
<div data-role="controlgroup" data-type="horizontal">
<a href="index.html" data-icon="home" data-role="button" class
="ui-btn-active ui-state-persist">Home</a>
<a href="contact.html" data-icon="info" data-role="button">
Contact</a>
</div>
<h4>Pour Programmez magazine</h4>
</footer>

</section>
</body>
</html>

```

[Fig.5]

> Construire une page en forme d'accordéon

La forme d'accordéon est composée de plusieurs balises <div> ouvrables (collapsible), permettant d'avoir plusieurs informations cachées sous chacun des titres de la page. L'ensemble des balises <div> collapsible est entouré d'une balise, <div>. L'attribut data-role de cette balise est **data-role="collapsible-set"**

```

<!doctype html>
<html lang="fr">
<head>
<meta charset="utf-8"/>
<meta name="viewport" content="width=device-width, initial
-scale=1">

```

```

<title>contact page jQueryMobile</title>

<link rel="stylesheet" href="jquery/jquery.mobile-1.0.1/
jquery.mobile-1.0.1.min.css"/>
<script src="jquery/jquery-1.7.1.min.js"></script>
<script src="jquery/jquery.mobile-1.0.1/jquery.mobile-1.
0.1.min.js"></script>
</head>

<body>
<section id="contact" data-role="page">

<header data-role="header" data-theme="b">
<h1>Article jQM</h1>
</header>

<div data-role="content">

<!--Composition de collapsibles -->
<div data-role="collapsible-set">
<h1>Contactez-nous</h1>

<div data-role="collapsible" data-theme="e">
<h3>Par Email</h3>
<h4>Vous pouvez nous envoyer un message</h4>

<form>
<div data-role="fieldcontain">
<label for="prenom">Prénom :</label>
<input name="prenom" id="prenom">

<label for="nom">Nom :</label>
<input name="nom" id="nom">

<label for="email">Email :</label>
<input type="email" name="email" id="email">

<label for="message">Texte de votre message</label>
<textarea name="message" id="message"></textarea>

<input type="submit" data-role="none" value="envoyez">
</div>

</form>
</div>

<div data-role="collapsible" data-theme="e">
<h3>Par Téléphone</h3>
<a href="tel://00 00 00 00 00">Touchez pour nous appeler </a>

```

```

</div>
</div>

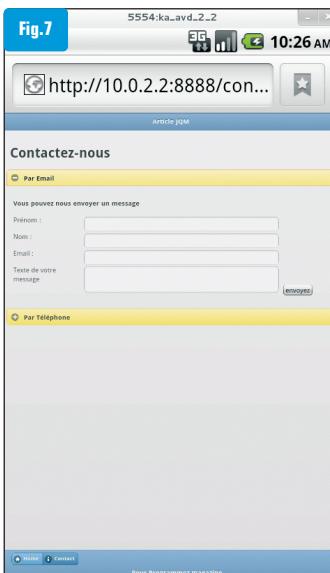
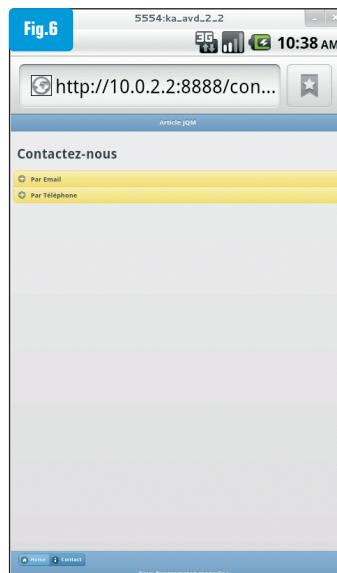
</div>

<footer data-role="footer" data-position="fixed" data-id="nav" class="ui-bar" data-theme="b">
<div data-role="controlgroup" data-type="horizontal">

<a href="index.html" data-icon="home" data-role="button" class="ui-btn-active ui-state-persist">Home</a>
<a href="contact.html" data-icon="info" data-role="button">Contact</a>

</div>

```



```

<h4>Pour Programmez magazine</h4>
</footer>

</section>
</body>
</html>

```

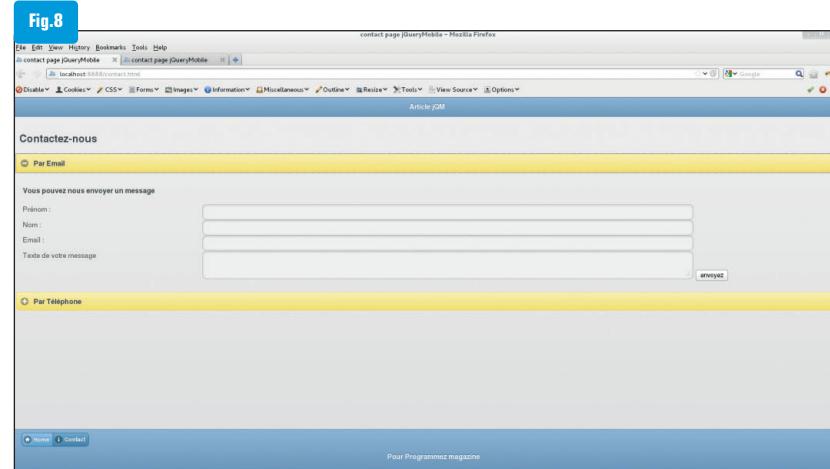
[Fig.6, 7 et 8]

CONCLUSION

jQueryMobile est un outil performant, permettant avec un minimum de code d'adapter un site Web, pour le rendre utilisable et ergonomique sur de petits écrans, comme des mobiles, et des tablettes. Sa puissance ne s'arrête pas là, ce framework, style aussi les sites Web consultables en utilisant des navigateurs standard sur les ordinateurs. Il sait adapter l'affichage selon les résolutions utilisées et la plateforme.

Kaesar Alnijres

Développeur Java, Leader jug-cergy (Java User Group – Cergy)
<http://www.java-javafx.com>



L'INFO permanente

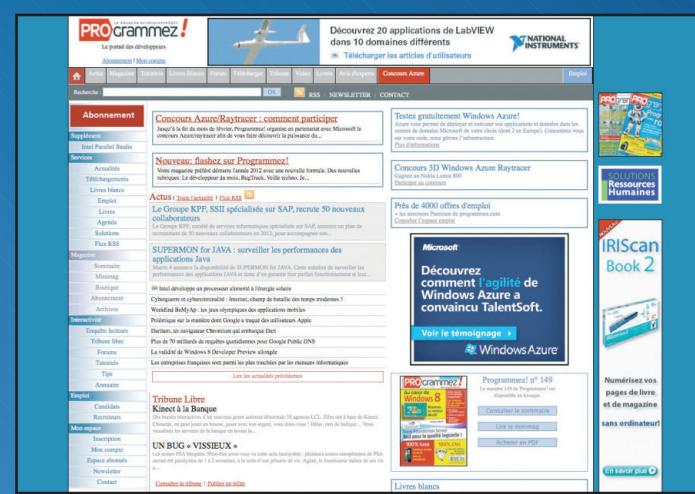
- **L'actu** : le fil d'info quotidien de la rédaction
- **La newsletter hebdo** : abonnez-vous, comme 38 000 professionnels. **C'est gratuit !**

C'est PRATIQUE !

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

www.programmez.com

PROgrammez! Le magazine du développement www.programmez.com



MySQL : identifier les tables saturées

Lors de la réalisation d'un projet web, de nombreuses solutions existent pour mémoriser des informations. Ces informations ont toutes de l'importance au niveau de l'historisation et de l'archivage. Or, vous pouvez toujours être à la merci d'une saturation de la base car vous ne l'aviez pas prévu (eh oui, on ne pense pas à tout !).

PRÉSENTATION

De nombreux projets internet sont amenés à utiliser une base de données pour une application spécifique, du e-commerce ou du stockage d'informations diverses pour afficher des statistiques par exemple. L'ensemble de ces informations peuvent être enregistrées de différentes manières :

- Une sauvegarde sur le disque en fichier log
- Dans la base de données
- En mémoire
- Ne rien faire

Si vous choisissez de placer toutes ces données dans une base de données, vous pouvez voir un jour le message suivant :

```
Exceed the file size limit for the operating system...
```

Que faire ?

COMMENT LE COMPRENDRE

Si votre choix de stocker des données s'est porté sur un archivage dans une base MySQL, c'est dans le but de réaliser des pages web pour ressortir facilement des résultats statistiques sur une période définie, pouvant être représentée sous la forme d'un tableau ou d'un graphique (histogrammes, courbes, etc.). Ces informations sont très importantes pour effectuer des analyses précises sur de longues périodes. Cependant, lors de l'installation de votre environnement AMP (Apache MySQL PHP), cette préoccupation de volumes de données et d'espace nécessaire n'est pas toujours prise en compte dans la configuration de base de votre projet.

DÉTECTOR LE PROBLÈME

Pour repérer ce phénomène de saturation et identifier une table concernée, celui-ci peut s'effectuer de différentes manières. Tout d'abord, vous pouvez vous rendre compte que les données ne s'enregistrent plus dans la base, et par conséquent les résultats de synthèse sont faux ou absents. Mais vous ne maîtrisez pas à quel moment une ou plusieurs tables vont être saturées. Ensuite, vous devez connaître la version de votre système d'exploitation car il a un rôle important envers votre base de données. Selon la configuration par défaut, en 32 ou 64 bits et de votre OS, vous n'êtes pas obligé de choisir une configuration par défaut de la base de données. Mais vous pouvez effectuer les modifications de taille maximum à tout moment. Dans la majorité des cas, pour chaque table la taille d'une base de données est de 2 Go, et le standard devient de plus en plus à 4 Go, voire plus. La dernière possibilité de savoir si une table risque d'être saturée, consiste à interroger directement votre base de données et l'opération peut s'effectuer de 2 manières :

- si vous pouvez accéder à la base de données en ligne de commandes, vous pouvez effectuer l'opération suivante :

```
> SHOW TABLE STATUS ;
```

Le résultat qui sera affiché, propose le volume de chaque table occupé et la taille maximum définie par défaut.

- Si ce n'est pas le cas, vous pouvez effectuer l'opération par programmation :

```
<?php
$cnx = mysql_pconnect($serveur, $user, $password);
mysql_select_db($bdd,$cnx);
$rows = mysql_query("SHOW TABLE STATUS");

while ($row = mysql_fetch_array($rows)) {
echo "Nom de la table : ".$row['Name']."<br>";
echo "Nombre de lignes : ".$row['Rows']."<br>";
echo "Espace occupé de la table (base de données + index) : ".$row
['Data_length'] + $row['Index_length'])."<br>";
echo "Maximum prévu de la table : ".$row['Max_data_length'].
"<br>";
echo "<hr>";
}
?>
```

SOLUTION

Quelle que soit la méthode de calculs que vous utilisez pour repérer la table qui va saturer en premier, la solution nécessite une intervention dans la base de données, et plus précisément les tables qui stockent les informations.

2 solutions importantes doivent être mises en place :

- La première consiste à sauvegarder juste les ID des autres tables. C'est-à-dire qu'il ne faut pas enregistrer les libellés, ou les noms. Par conséquent, vous devez reconstruire une table dite « propre » et correctement configurée. Pour obtenir un historique, l'utilisation des jointures devient nécessaire pour afficher les différentes vues souhaitées. Mais ce type d'intervention est provisoire car vous risquez de retrouver le même problème d'ici quelque temps.
- La deuxième solution consiste à augmenter la taille des tables définie par défaut, mais attention car il faut que votre système d'exploitation puisse supporter ce volume de données plus important. La fonction que vous devez utiliser pour effectuer l'opération, est :

```
> ALTER TABLE nom_de_table MAX_ROWS=1000000000 AVG_ROW_LENGTH=nnn;
```

Ces 2 critères supplémentaires à la configuration de la table correspondent :

- MAX_ROWS : permet de définir le nombre de lignes
- AVG_ROW_LENGTH : Ce critère est indispensable pour les tables ayant des champs BLOB/TEXT

Enfin, pour certains sites qui sollicitent beaucoup de stockage, il faut régulièrement surveiller ces tables et surtout ne pas garder obligatoirement plus de données qu'il n'en faut.

Christophe Villeneuve

Consultant pour Alter Way solutions, auteur du livre « PHP & MySQL-MySQLi-PDO, Construisez votre application », aux Éditions ENI. Rédacteur pour nexen.net, membre des Teams DrupalFR, AFUP, LeMug.fr, PHPTV.

Formez et fidélisez vos équipes techniques

en leur offrant un **Abonnement à Programmez !**

Une formation et une veille technologique, à prix réduit



• Vos développeurs vous le demandent

Les développeurs attendent des formations (30%), avant des augmentations de salaires (28%)

• Ils se forment d'abord par eux-même, et par la lecture

68% des développeurs déclarent se former par eux-même, 65% au moyen de livres et manuels : la lecture est privilégiée.

Source : Linux Fondation

→ 1 an d'abonnement = 39 € seulement par abonné (à partir de 10 abonnés)

**Pour 3,25 € par mois par développeur,
apportez-leur : information, productivité, satisfaction.**

+Offre de lancement : 1 page de Publicité offerte pour votre entreprise !
À partir de 100 abonnés, offre de lancement

Adressez un mail à diff@programmez.com, en indiquant le nombre d'abonnés souhaité, pour recevoir le devis.

www.programmez.com

2 1
C'EST MIEUX QU'
JUSQU'AU
20 JUILLET

OPÉRATION
**2 POUR
1 EURO
DE PLUS**

exemple:

Samsung
GALAXY S III

Samsung
GALAXY Tab 2
10.1



Aucun abonnement
à souscrire pour
bénéficier de cette offre.

**Choisissez
2 matériels au choix !**



Nouveau smartphone
ultra performant
Samsung Galaxy S3
Ecran 4.8 pouces 720 x 1080
Android 4.0
APN 8 Mégas
Tous les détails techniques sur
www.pcsoft.fr



Nouvelle tablette
ultra performante
Samsung Galaxy Tab 2
Ecran 10.1 pouces
Android 4.0 - 16 Go
Wifi ou Wifi/3G
Tous les détails techniques sur www.pcsoft.fr



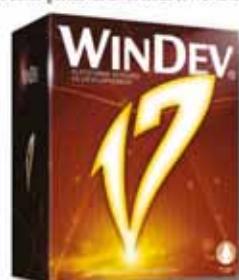
Mi-tablette
Mi-Smartphone
Samsung Galaxy Note
Ecran 5.3 pouces 1280 x 800
Android
Détails techniques sur
www.pcsoft.fr



Smartphone étanche
Samsung X Cover
Ecran 3,65 pouces
Android 2.3
Tous les détails techniques
sur www.pcsoft.fr



PC Portable
Samsung 300E5A
Ecran 15,6 pouces LED
2,3 kg
Processeur Intel
Disque dur 500 Go
Windows 7 familial



VOUS AUSSI,
DÉVELOPPEZ
10 FOIS PLUS
VITE

Atelier de Génie Logiciel Pro-
fessionnel : Windows, Linux,
.Net, Java, PHP, J2EE,
XML, Internet, Cloud, Windows
CE & Phone, Android, iPhone...



Fournisseur Officiel de la
Préparation Olympique



www.pcsoft.fr

info@pcsoft.fr
Tél: 04.67.032.032