

PROGRAMMEZ !

www.programmez.com

Mensuel n°155 - Septembre 2012

Choisir son **language** de programmation !



© istockphoto C.Stepan Kapl

© fotografiedk fotolia.com

HTML 5

CSS 3 : libérez vos interfaces web

**Créer son clone
Dropbox**

Les **nouveautés** d'Eclipse 4.2

Ubercart : votre boutique
e-Commerce sur Drupal

● Webmaster

Administrer le raid sur
un serveur Linux

● Android

Hautes performances
graphiques avec **Renderscript**

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

M 04319 - 155 - F. 5,95 €



**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.**

Nouveau !

- 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**.

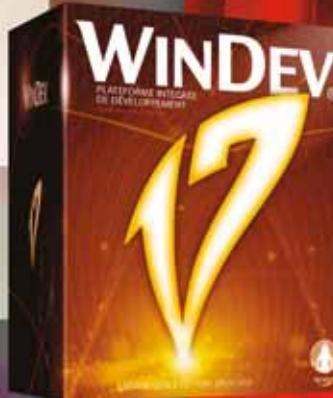
Intégralement en français.
Support Technique inclus.
Ouvert à tous les standards,
à toutes les bases de données.

Elu «Langage le plus productif du marché»

WINDEV, WEBDEV et WINDEV Mobile sont compatibles

**VERSION
EXPRESS
GRATUITE**

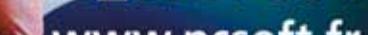
Téléchargez-la !



Scannez ce code pour recevoir le dossier

► **DEMANDEZ VOTRE DOSSIER GRATUIT**

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



www.pcsoft.fr

EXIGEZ WINDEV 17

**POUR LE DÉVELOPPEMENT
DE VOS APPLICATIONS**

Fournisseur Officiel de la
Préparation Olympique

Google I/O 2012 : Android, tablette Google, et quelques surprises

La 5^e édition de Google I/O, la conférence technologique et développeur de Google, s'est déroulée du 27 au 29 juin dernier à San Francisco. Beaucoup de rumeurs sur une tablette Google, des lunettes Google, le nouvel Android... Cette édition a remis à plat une partie de la stratégie. Revenons sur les principales annonces.

Android a été la principale annonce de la journée avec Android 4.1 alias Jelly Bean. Le marché est capital pour l'éditeur et les partenaires : 400 millions de terminaux fonctionnent avec ce système. Et la croissance est phénoménale : +200 % en France, 500 % au Brésil.

ANDROID 4.1, GRAPHIQUEMENT DOPÉ !



Android 4.1 subit de nombreuses évolutions. La grande nouveauté est l'arrivée au cœur du système du projet Butter, dont l'objectif est d'améliorer le rendu graphique, sa qualité, ses performances, sa réactivité. Le moteur graphique est souvent le point faible d'Android. La partie animation sera à la hauteur des derniers terminaux ainsi que la réactivité, notamment quand on touche l'écran. Cela devenait indispensable pour exploiter au mieux les nouveaux processeurs mobiles et la partie GPU. Google utilise systrace pour récupérer les données du noyau. Un SDK sera disponible pour les éditeurs et constructeurs. Par rapport à la v4, la 4.1 est effectivement bien plus réactive à en croire les démos. De nombreuses applications seront adaptées à cette nouvelle technique. On peut même redimensionner des fenêtres, à voir, au quotidien, si cela est réellement pratique.

Le clavier virtuel a subi un important lifting. Pour les connaisseurs du clavier du futur Blackberry 10, on retrouve des similitudes. On bénéfice de la saisie vocale en mode

déconnecté. La v4.1 améliore de nombreuses fonctions : accessibilité, support de langues, la caméra, etc. Le NFC étend ses fonctionnalités comme l'envoi de photos et de vidéos. La partie recherche a été largement remaniée sur l'interface. On dispose maintenant d'un graphe de connaissances qui s'enrichit selon vos recherches, les résultats, pour aller plus vite. La recherche vocale a été améliorée. Android 4.1 est disponible depuis la mi-juillet. Les mises à jour constructeurs sont en cours.

LE FAUX DÉPART DU NEXUS Q

Oui, Google a bel et bien sa propre tablette tactile : la Nexus 7, conçue et construite par Asus. Elle inclut Tegra 3, un écran HD (mais pas Retina comme iPad), deux caméras, toutes les technologies réseaux et pèse 340 grammes. Il s'agit d'une tablette 7". Google mise sur la vidéo, la musique, le jeu. Ce positionnement vise plutôt le Kindle d'Amazon et pas forcément Samsung ou Apple.

Le côté création n'a été que très peu mis en avant. Le stockage est limité à 16 Go, ce qui est peu pour la vidéo. La tablette est disponible en France depuis quelques semaines. Le tarif conseillé est de 249 € pour la version 16 Go.

Google a profité de la conférence I/O pour dévoiler une étrange boule noire : la Nexus Q. Elle peut interagir avec des terminaux et être commandée par ces mêmes terminaux. Une sorte de mini PC sous Android. Initialement son principal intérêt était le streaming de vidéo depuis un stockage en ligne. Mais son prix (299 \$), le peu de fonctionnalités, l'absence d'une fonction comparable à Air



Play d'Apple ont été fortement critiqués. Google a décidé de retarder le lancement de sa boule, le temps de la remplir de quelques fonctions supplémentaires et peut être de faire un sérieux effort sur le prix. En comparaison, une Apple TV est vendue 109 €.

CHROME ET CLOUD COMPUTING

Google Chrome dépasse 310 millions d'utilisateurs, soit le double par rapport à 2011. Surtout, Chrome est disponible sur desktop, smartphone et tablette.

Au passage, notons aussi que Google aime bien Apple car de nombreux portables Apple étaient utilisés durant les démos ! Chrome arrive aussi sur iOS.

Sur la partie cloud, l'annonce la plus importante est celle d'un service infrastructure IaaS : Google Compute Engine, quelques jours après l'annonce de Microsoft autour de Windows Azure. Il s'agit d'un service d'infrastructure de type IaaS (cloud) utilisant des instances et des ressources. L'instance contient n ressources (stockages, CPU, mémoire vive, etc.) et différents types d'instances sont disponibles.

Compute Engine repose sur l'infrastructure Google qui dépasse les 600 000 cœurs processeurs ! Google mise sur la haute disponibilité, la montée en charge, les performances, la sécurité. Il est possible de gérer les applications et environnements déployés sur Compute Engine via des outils tiers comme OpsCode, RightScale.

Tous les détails sur :
<https://developers.google.com/compute/>

**BUZZWORD**

Google I/O 2012 :
Android, tablette Google et quelques surprises3

ACTUS

En bref6
Agenda8
PHP Day et Masterclass de Rasmus10

RESSOURCES

Livres et événements du mois14

OUTILS

Websico : un CMS simple comme une souris15

DÉVELOPPEUR DU MOIS

Jonathan Roquelaure :
« c'est dans l'informatique que j'ai finalement
trouvé le bonheur ».....16

**MATÉRIEL**

Linutop 4.0, un couteau suisse pour geeks18

**OUTILS**

Les nouveautés du Windows Azure
SDK 1.720
Eclipse Juno : le nouvel Eclipse22
Premiers pas avec Wakanda
(2e partie)24

DÉVELOPPEMENT WEB

CSS3 de A à Z48

CARRIERE

Chef de projet « pompier » (retour d'expérience)52

JE DÉBUTE AVEC...

Test Driven Development avec Visual Studio 2012 ..55
Je débute avec... OpenShift59

CAS D'USAGE

Créer son propre service
de stockage en ligne,
disponible 24/24h62

**DOSSIER**

Choisir son langage de programmation

Le bon langage au bon moment37

Les critères de choix42

Windows 8 de A à W

Sortie mondiale le 26 octobre de Windows 8.
Préparez-vous au nouveau Windows : nouvelles
API, nouvelle interface, nouveaux outils,
Windows Store, version ARM.

Eclipse : Développez rapidement vos plug-ins Eclipse

Webmaster : Adobe CS 6 : quoi de neuf pour les
développeurs et développeurs web ?

Je débute en C++ : C++ c'est beau, performant et
fun. La preuve !

DOSSIER

Domotique,
La maison
connectée
(2e partie)

Piloter sa maison ?

C'est facile !

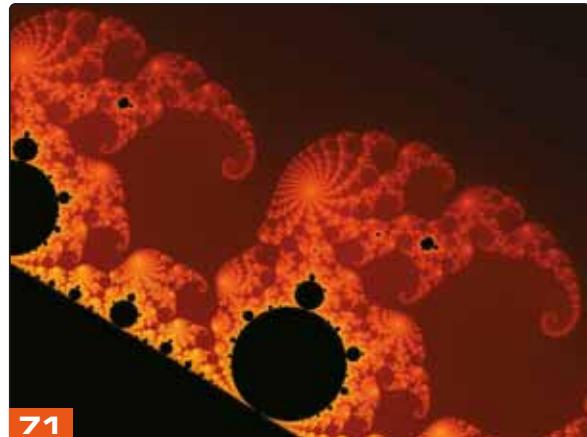
Un peu de pratique 30

PRATIQUE

Renderscript, optimisation CPU/GPU	
pour Android	65
Sharepoint 2010 :	
PowerShell et la brique Search	69

caminc

Réaliser vos fractales.....	71
-----------------------------	----



WEBMASTER

Personnaliser la gestion des produits sous	
Ubercart/Drupal 6	73
Administrer le RAID d'un serveur Linux	77

BUGTRACK

Windows 8 et async/await.....	82
-------------------------------	----

Editeur : Go-O2 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. **Experts :** G. Coat, W. Woiré, X. Seignard, T. Arguillère, A. Morgaut, S. Warin, C. Falguière, M. Levi, J. Ballet, P. Berenguel, D. Girond, N. Kamenhoff, T. Royer, J. Roche, M. Delrieu, G. Bougenière, J-S. Mercy, P. Garry, H. Desaunois, S. Galand, N. Babaci, C. Villeneuve, S. Hertrich, Jason DeOliveira, Fathi Bellahcene.

Illustration couverture : montage comprenant photo de main (© istockphoto C.Stepan Kapl) et fond avec noms de langage (© fotografiedek - Fotolia.com).
Publicité : Régie publicitaire, KNow sarl.
Pour la publicité uniquement : Tél. : 01 41 77 16 03 - diff@programmez.com. Dépot 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 Vaudrelane
Ce numéro comporte 2 encarts jetés pour les abonnés



Tempête ou coup d'épée dans l'eau ?

Alors que la chaleur s'abattait sur la France à la mi-août, l'été n'empêchait pas le monde informatique de s'activer. La rentrée promet d'être pleine de surprises

et de nouveautés. Eh oui, les fêtes de fin d'année sont presque là ! Ne vous inquiétez pas, début janvier, on se préoccupera de l'été ! Décidément, le temps linéaire n'est pas notre ami.

Google a tiré le premier avec la nouvelle version d'Android et une partie graphique boostée pour améliorer les animations et les performances globales en 2D et 3D. Développeurs et utilisateurs apprécieront. Apple a enchaîné avec OS X 10.8 qui est finalement dans la continuité de la 10.7, avec quelques fonctions intéressantes comme l'intégration iCloud ou encore Power Nap et les premiers pas de Siri sur Mac. Cependant, tout le monde attend les nouvelles machines iMac et surtout Mac Pro. iOS 6 arrivera à l'automne avec un nouvel iPhone et peut être un iPad 7".

Microsoft n'est pas en reste avec les tablettes tactiles sous Windows RT, la disponibilité de Windows 8 (le 26 octobre) et de Windows Phone 8. Ce dernier fait beaucoup de bruit à cause d'une compatibilité avec les modèles actuels quasiment inexistante ! Ce faisant, Microsoft fragmente le marché Windows Phone alors qu'il cherchait à l'éviter. La tâche du développeur est d'autant plus compliquée que pour la partie tablette, il faut adapter et coder pour Windows RT et non Windows Phone. Nous verrons si l'approche d'Apple autour de iOS et OS X (qui amorcent clairement une « fusion ») est la meilleure (Google a opté pour la même démarche pour aboutir à une version unifiée téléphone - tablette), ou si la démarche de Microsoft réussit à prendre sur le difficile marché des terminaux mobiles et des tablettes. Début septembre, Nokia doit dévoiler de nouveaux terminaux en espérant sortir du fossé dans lequel se trouve le constructeur depuis presque 2 ans. Il faut dire que iOS et Android font exploser les constructeurs historiques du mobile. RIM n'est pas non plus en grande forme. Et parler d'un rachat pur et simple n'est plus une illusion mais une réalité. Et que dire du flou artistique de la stratégie HP autour des terminaux et systèmes mobiles : un coup on nous tue les tablettes, webOS et Palm. On ferme tout. Un autre jour, on nous annonce une division « mobility » et l'apparition probable de nouvelles tablettes (mais avec quel OS ?). L'avenir de webOS reste incertain même s'il a été donné à la communauté. La question est de savoir : pourquoi faire ?

Nous suivrons aussi de près la situation de HTML 5. Fork ou pas fork. La question est désormais au-delà de cette désunion. Elle est de savoir si la communauté des développeurs et les éditeurs remettent un peu d'ordre dans ce désordre. L'affaire a au moins le mérite de mettre en lumière les carences et lenteurs du W3C.

François Tonic - Rédacteur en chef
ftonic@programmez.com

Abonnement : Programmez, 17, Chemin des Boulanger, 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

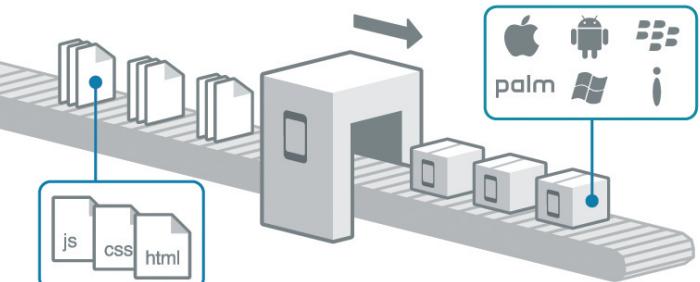
» Microsoft prépare activement Windows 8,

Windows RT, Windows Server 2012 et Visual Studio. Windows 8 est disponible en version RTM (version finale prête pour l'industrie) pour les abonnés MSDN, et depuis le 16 août pour les souscripteurs à la Software Assurance. La version ARM pour tablette est également disponible. L'environnement de développement Visual Studio 2012 est lui aussi en RTM depuis le 4 août. Le lancement officiel est attendu pour le **12 septembre**. A vos codes pour vous préparer à Windows 8 et à toutes les nouveautés. Sortie : 26 octobre.

» Adobe a dévoilé la version 2.0

d'un des outils de développements mobiles les plus utilisés : PhoneGap. Cette v2 propose de belles améliorations : command line tooling, documentation étendue, inspecteur web distance, migration vers Apache Cordova, amélioration de la création d'application iOS, support de Windows Phone ! Le framework est disponible gratuitement. Pour construire les applications, le projet propose PhoneGap :Build, outil pour transformer directement le code en applications mobiles. Pour le développeur, l'outil est gratuit, puis les licences démarrent à 12 \$ par mois. Il est actuellement en bêta et gratuitement disponible. Une conférence PhoneGap aura lieu le 14 septembre à Amsterdam.

Site officiel : <http://phonegap.com>



» Mozilla continue de faire évoluer Firefox.

Pour les développeurs, le projet supporte Pointer Lock API, API pour mieux contrôler la souris (applications et jeux). Il s'agit d'une technique expérimentale (https://developer.mozilla.org/en-US/docs/API/Pointer_Lock_API). Autre amélioration, une transformation de texte améliorée en CSS pour le support du Turc et du Grec. Toutes les nouveautés de la version 14 : <http://www.mozilla.org/en-US/firefox/14.0/releasenotes/>

» Ubuntu 12.10 permettra d'exécuter des applications et sites web en dehors du navigateur,

comme n'importe quelle autre application. Elles seront disponibles sur le desktop. Pour en savoir plus sur cette nouvelle fonction : <http://www.omgubuntu.co.uk/2012/07/ubuntu-unveil-new-web-apps-feature-for-12-10>

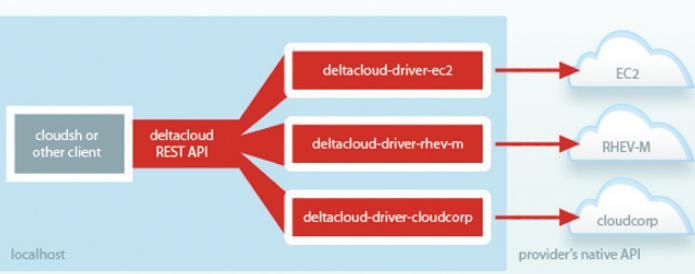
» Facebook a dévoilé mi-juillet une nouvelle version de son SDK pour iOS.

La version 3.0 simplifie la gestion des utilisateurs, la possibilité d'utiliser des contrôles d'interfaces natifs au système et enfin permet de mieux supporter les API Facebook. Ce SDK fonctionne à partir de iOS 4.

Lien : <https://developers.facebook.com/ios/>

» L'API Deltacloud,

crée par Red Hat et donnée à la fondation Apache, est depuis quelques jours en version 1.0. Deltacloud fournit une API serveur et des pilotes pour se connecter à divers cloud comme Amazone EC2. Il s'agit donc d'une API fournissant un niveau d'abstraction et donc indépendante des fournisseurs, des services. Elle doit permettre de gérer les instances cloud, d'avoir une certaine indépendance entre applications et les évolutions des API cloud des fournisseurs. Pour le développeur, cette API fournit une API REST open source écrite en Ruby.



» HP dévoile une nouvelle version de sa solution de gestion du cycle de vie

de l'application, ou ALM, HP ALM 11.5. Cette solution poursuit trois objectifs : l'ALM, le suivi des projets et l'approche DevOps. Cette nouvelle version propose des rapports préconfigurés intégrant des informations à jour sur les processus applicatifs. D'autre part, HP ALM 11.5 s'intègre au logiciel HP Enterprise Collaboration, qui permet des échanges contextualisés sur un mode de réseau social entre responsables clés du cycle de vie des applications. ALM 11.5 intègre aussi le support de l'outil GIT. HP ALM est disponible en SaaS, comme plusieurs offres concurrentes. HP ALM en SaaS est une offre importante pour HP Software car les entreprises n'ont plus à déployer un ALM localement, opération coûteuse et longue. C'est un des arguments de Microsoft avec la version en ligne de la suite Team Foundation Server.

» NetBeans est disponible en version 7.2.

Cette version apporte comme toujours des améliorations et quelques nouveautés. Les performances globales de l'IDE ont été améliorées comme le démarrage et la revue des projets.

Sur la qualité du code, le développeur dispose d'un nouvel outil d'analyse statistique open source : FindBugs. Concernant le futur, la roadmap ne fournit pas encore de précisions sur les prochaines versions. Habituellement, les versions majeures sortent vers mai -juin.

» OS X repose sur des fondations open

source. A chaque évolution, Apple publie le code source correspondant. A peine 3 semaines après la disponibilité de OS X 10.8, le constructeur vient de mettre en ligne les sources ouvertes du système. La liste est assez longue, plus de 200 projets ouverts sont utilisés par Apple dans OS X et les outils de développement. Site : <http://opensource.apple.com/release/mac-os-x-108/>

JS.everywhere(2012) Europe

16 - 17 novembre 2012 • Paris

JS.everywhere() est une conférence internationale entièrement consacrée à l'essor de JavaScript dans le secteur informatique professionnel : Web Apps, Mobiles Apps, JavaScript est désormais partout. Deux jours de rencontres et d'échanges avec plus de 300 experts et passionnés venus de toute l'Europe.

1ère édition européenne

Au programme

- Keynotes & sessions sur : Server-Side JavaScript, NoSQL, HTML5, Model Driven Architecture, sécurité, accessibilité, etc.
- Ateliers "hands on" : fondamentaux du développement Web et Mobile moderne
- Rencontres et networking, repas, goodies, au Tapis Rouge (Paris 10^{ème}), un lieu unique chargé d'histoire réservé pour toute la conférence
- En bonus, ateliers Wakanday : découvrez Wakanda, la nouvelle plateforme 100% JavaScript

Note : la majorité des contenus sera présentée en anglais



Speakers Confirmés



Douglas Crockford



Laurent Ribardière



Michal Budzynski



Estelle Weyl



Dio Synodinos



Alexandre Morgaut



Melinda Gallo

Et bien d'autres à venir ...



Inscriptions ouvertes !

Profitez du tarif "Early Bird" avant le 15 octobre

Réservez votre place sur jseverywhere.eu

Venez en tant que Speaker !

Proposez votre session avant le 15 septembre

jseverywhere.eu/call-to-speakers

Un événement



avec le concours de



#jse2012



HTML 5 va-t-il voir double ?

Depuis juillet dernier, la communauté HTML 5, le W3C et autres WHATWG en tête, s'agitent. Ian Hickson (fondateur du Web Hypertext Application Technology Working Group) a publié sur une mailinglist W3C un message très critique envers le W3C et leur manière de travailler sur HTML 5. Le premier problème évoqué est la décision de découper HTML 5 en plusieurs sous-spécifications (canvas 2D, postMessage, etc.).

Pour lui, cela provoque une confusion croissante dans les versions des spécifications. Chaque sous-spécification avançant « de son côté », il est parfois difficile de savoir où en est chaque groupe et quelle spécificité est réellement prise en compte dans HTML 5. C'est pour cela que le WHATWG a voulu reprendre le HTML 5 actuel pour travailler sur une unique spécification, comprenant tous les éléments formant HTML 5. Ce travail est appelé « HTML Living Standard ». Sur le site officiel du WHATWG, le living standard est déjà là avec l'ensemble des documentations et des travaux. L'équipe évoque aussi la différence entre la spécification W3C et WHATWG. Ce dernier semble regretter l'appellation HTML 5 et non HTML

tout court, et que les exemples d'utilisation ne soient pas inclus par défaut. Il pointe du doigt plusieurs omissions dans les spécifications du W3C, tout en mettant en avant différents éléments uniquement disponibles dans la spécification WHATWG (attribut inputmode, nouveautés sur les fonctions des hyperliens...).

C'est donc un fork qui se met en place ! D'autre part le WHATWG affirme qu'il y a une divergence sur le HTML avec le W3C. Ce dernier veut créer une « version » de HTML 5 selon le processus interne, alors que le WHATWG affirme vouloir développer et définir une description canonique de HTML et des technologies liées, en corrigeant les bugs au fur et à mesure et en rajoutant les nouvelles fonctions nécessaires et viables.

Tout cela ne va pas aider à y voir plus clair. Qui a raison et qui à tort ? HTML 5, au sens

W3C, n'est toujours pas disponible en son entier et cette lenteur de définition a été souvent pointée du doigt. Va-t-il exister deux spécifications HTML 5 ? Visiblement oui. Même si aujourd'hui, celle du W3C (les parties stabilisées) prime.

Mais les éditeurs veulent aller plus vite pour implémenter et utiliser massivement HTML

5. Or, les lenteurs actuelles dans la conception et la validation des spécifications sont de réels problèmes.

Le WHATWG voudrait faire sauter cette contrainte pour mieux suivre le marché et être bien plus réactif. Les éditeurs, et surtout les gros consommateurs de HTML5, vont peser sur cette affaire car ce sont les premiers concernés (implémentations, outillage). Microsoft ne fait pas partie du WHATWG, contrairement à Google, Apple, et Mozilla.



Sources : <http://lists.w3.org/Archives/Public/public-whatwg-archive/2012Jul/0119.html>
Et <http://www.whatwg.org/specs/web-apps/current-work/multipage/>

agenda

► SEPTEMBRE

- 11 septembre 2012, **ParisJug 2012** (voir détail page Ressources) <http://www.parisjug.org/xwiki/bin/view/Meeting/20120911>
- 13 septembre 2012, Paris La Défense, **CodeWay Tour 2012** - Développement Mac pour développeurs Windows ! Embarcadero organise un tour de France pour vous faire découvrir le RAD Studio XE2 et le développement pour Windows et Mac (OSX, iOS). <http://forms.embarcadero.com/forms/EM12Q2FCodewayTour2012> <http://www.parisjug.org/xwiki/bin/view/Meeting/20120911>
- 27 septembre 2012, Paris, Séminaire technique gratuit : **Optimiser et mesurer la performance de vos applications C/C++ et Fortran avec Intel Parallel Studio XE 2013**. Organisé par Microsigma et Intel. www.microsigma.fr/intel/

► OCTOBRE

- Du 11 au 13 octobre 2012, Paris 8e, Eurosites Georges V, **Open World Forum 2012**. L'édition 2012 sera dédiée à l'utilisateur : entreprise, administrations et collectivités. Fidèle à ses origines, l'OWF 2012 apportera un éclairage nouveau sur les approches fondées sur l'ouverture qui transforment à la fois la société et les entreprises. <http://www.openworldforum.org>
- Du 18 au 19 octobre 2012, Forum IBM de Bois-Colombes, **Paris Web 2012**, les inscriptions sont ouvertes. <http://www.paris-web.fr/presse/2012/06/les-inscriptions-pour-paris-web-2012-sont-ouvertes.php>

► Dell aime l'open source et veut le faire savoir.

Le constructeur lance officiellement le projet Sputnik qui était en réflexion depuis 6 mois. Il s'agit de concevoir un ultrabook orienté développeur et basé sur une distribution Ubuntu. Ce projet deviendra réalité en automne prochain ! Le fabricant envisage actuellement de proposer dans certaines régions un portable pour développeur basé sur son Dell XPS 13, avec Ubuntu 12.04LTS préinstallé. « *Le Projet Sputnik est un parfait exemple d'innovation portée par nos collaborateurs et favorisée par le programme d'incubation que Dell a mis en place* », explique Nnamdi Orakwue, chargé du programme d'incubation de Dell et assistant exécutif de Michael Dell. « *Ce projet inaugure toute une série d'idées nouvelles que les salariés de Dell vont tester auprès de nos clients ou de nos partenaires dans le cadre de ce programme. Nous sommes impatients de voir arriver cet automne le premier produit du Projet Sputnik, avec la volonté d'assurer son succès.* »



Site officiel :

<http://content.dell.com/us/en/enterprise/d/campaigns/sputnik.aspx>

Performance et Simplicité Au Bout Des Doigts

Découvrez le nouveau NetAdvantage for .NET 2012 V.1 qui répond au doigt et à l'œil!
Essayez la version gratuite dès aujourd'hui!

www.infragistics.com/.NET



Compatible avec
Microsoft® Visual
Studio® 2012

INFRAGISTICS
DESIGN / DEVELOP / EXPERIENCE

Infragistics Ventes France ☎ 0800 667 307 • Infragistics Ventes Europe +44 (0) 800 298 9055 • Infragistics India +91 80 4151 8042 • [@infragistics](https://twitter.com/infragistics)

Copyright 1996-2011 Infragistics, Inc. All rights reserved. Infragistics and NetAdvantage are registered trademarks of Infragistics, Inc. The Infragistics logo is a trademark of Infragistics, Inc.

PHP Day et Masterclass de Rasmus

Le 4 juin dernier, le « Campus Technologique du groupe IONIS », accueillait le PHP Day : une journée durant laquelle les étudiants de l'EPITECH et de SUP'Internet ont pu assister à des conférences et participer à des ateliers animés par des experts de PHP, dont Rasmus Lerdorf en personne, le créateur du langage PHP.

> Quatre conférences pour un état des lieux

La journée a commencé par un point sur la sécurité des sites web en général, dispensé par Damien Seguy co-auteur du livre « Sécurité PHP » (Eyrolles). Il s'est notamment attardé sur les recommandations en matière de sécurité de l'OWASP (Open Web Application Security Project - <https://www.owasp.org>) qui pointent du doigt les dix risques de sécurité applicatifs web les plus critiques, les principaux étant les failles permettant les injections (SQL), le Cross-site Scripting (XSS), les vols de session et les failles dans les systèmes d'authentification.

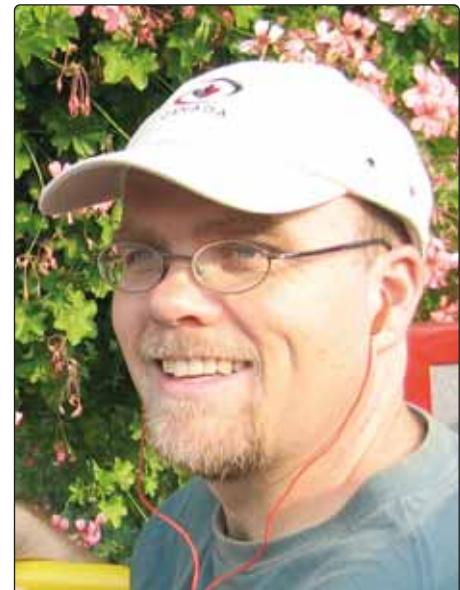
Les études de cas étaient à l'honneur avec

des exemples de failles XSS sur des sites très connus... une vraie leçon d'humilité pour les jeunes programmeurs de l'assistance !

Olivier Dasini auteur de « Audit et optimisation MySQL » (Eyrolles) a fait profiter les étudiants du campus de son savoir-faire sur l'optimisation MySQL. Pour Olivier Dasini, les quatre axes principaux d'optimisation des performances d'une base de données MySQL sont :

- le hardware
- la configuration du serveur MySQL
- l'optimisation du schéma de la BDD
- l'optimisation des requêtes SQL

Ce dernier point étant en général le premier



D.R.

à exploiter dans un système déjà en place car il permet des interventions chirurgicales là où une refonte du schéma aurait un impact bien plus important sur le code source et la solution en général.

Olivier Dasini a par la suite détaillé le fonctionnement interne du serveur MySQL afin de mieux démontrer l'impact de la conception du schéma de la base et de l'optimisation des requêtes SQL sur le coût en mémoire et sur les temps de réponse,



notamment via un typage approprié des champs (pas de VARCHAR(255) pour stocker des numéros de téléphone !) et l'utilisation d'outils internes à MySQL comme PROCEDURE ANALYSE ou encore EXPLAIN. Damien Seguy, a ensuite exposé aux étudiants les problématiques auxquelles peuvent être confrontées les entreprises dans leur approche de la programmation et dans la mise en production de leur travail. Les auditeurs ont pu découvrir les trois « mantras » indispensables pour survivre à l'industrialisation de son code et permettre un suivi à long terme :

- « Se libérer du code », s'assurer de rendre le code assez accessible pour qu'il puisse être maintenu par d'autres, notamment par le respect de conventions dans la structuration du code ou le nommage des variables et l'utilisation d'outils comme PHP_CodeSniffer
- « Capitaliser » par exemple en mettant en place des tests unitaires efficaces, des knowledge bases, des bugs tracker ou d'autres systèmes permettant de garder une trace des problèmes rencontrés et surtout des solutions trouvées !
- « Impliquer les utilisateurs » en recueillant leur feedback et en les sollicitant pour des tests ou pour remonter les bugs rencontrés.

Enfin, Guillaume Plessis, Julien Pauli et Cyril Pierre de Geyer, les auteurs de « Performance PHP » (Eyrolles) ont prodigué des conseils pour optimiser la pile LAMP (Linux Apache Mysql PHP) dans des situations de forte audience ou de traitement de données lourdes, par exemple. Analyse de logs, outils d'optimisation, construction de scénarios de code, techniques pour « stresser » ses applications sont autant de thèmes qui ont été traités durant cette troisième conférence.

> Rencontre avec Rasmus Lerdorf & sa masterclass

Le clou de la journée : l'intervention de Rasmus Lerdorf, le créateur de PHP. Après un rappel de l'histoire du langage, de son évolution jusqu'à aujourd'hui où PHP est considéré comme totalement incontournable. A la

question (ou plutôt au troll) de l'utilité de la création de PHP il nous a rappelé avec malice que PHP sauve des vies... grâce, par exemple, à un site en PHP qui permet de retrouver les disparus du tsunami (<http://sahanafoundation.org>), avant bien sûr de rappeler la philosophie de ce langage : la quête de la simplicité et de l'efficacité. Rasmus nous a aussi assuré de la volonté des développeurs de PHP de continuer, comme depuis maintenant plus de 15 ans, à faire coller PHP aux demandes et technologies du web qui changent chaque année. Il a promis que : « PHP supportera ces nouvelles technologies quelles qu'elles soient ». Rassurant ! L'échange avec l'auditoire fut particulièrement animé et intéressant et nombre d'étudiants n'ont pas hésité à confronter Rasmus sur les performances de PHP comparé à ses concurrents.

Rasmus Lerdorf a conclu son intervention par un tour d'horizon des fonctionnalités de PHP en 2012 : du Support Version Machine (<http://php.net/svm>), aux « Traits » (<http://php.net/manual/fr/language.oop5.traits.php>)



Paloma, étudiante en première année à Sup'Internet nous parle de sa journée.

Tu as participé au PHP Day, quelles ont été tes impressions ?

Déjà, honorée que mon campus reçoive le créateur de PHP. J'ai étudié ce langage toute l'année et c'est très impressionnant de voir l'homme qui est derrière quelque chose d'aussi complexe que PHP se présenter humblement devant nous !

Trouves-tu que ce genre d'événement s'inscrit avec pertinence dans ton parcours d'étudiant ?

Tout à fait. À SUP'Internet, la pédagogie nous pousse à la curiosité. Bien que je sois en bachelor création et design graphique, quand Damien Seguy nous invite à « sortir du code », ça me parle. Le code est un outil, mais doit être considéré dans

en passant par la nouvelle 'short array syntax' (\$a = [1, 2, 3];) ou encore des modifications de la syntaxe générale du PHP permettant une utilisation directe des arrays retournés par des méthodes d'objets ou des fonctions (foo()["bar"][]), la possibilité de faire des closures dans des objets, des closures statiques ou encore l'apparition d'un serveur web de test interne à PHP (attention c'est juste pour tester)... bref, Rasmus a su prouver que PHP regorge de surprises qu'il vous reste à découvrir !

Sur un barbecue géant dans la cour du campus technologique du groupe IONIS, le PHP Day s'est terminé tard dans la soirée, laissant à ses participants, auditeurs comme intervenants un excellent souvenir.

Gaël Coat,

Directeur délégué de SUP'Internet

Au sein de la structure, il se positionne comme le principal référent technique des étudiants et pilote le bachelor Développement et Technologies du Web. Il enseigne le PHP et est responsable de plusieurs modules technologiques majeurs. Il a été durant près de 4 ans le responsable du développement Web d'EPITECH.

un environnement complet. C'est pour ça que le web est à la fois complexe et passionnant.

Créer un langage de programmation, c'est quelque chose qui t'intéresserait ?

Déjà, j'aimerais maîtriser PHP à fond ! Mais en créer un nouveau, pourquoi pas. C'est un défi incroyable à mes yeux. Mais au-delà de cela, c'est l'engagement des intervenants qui m'a inspiré. J'ai envie d'entreprendre aussi et je sais qu'en choisissant une école du web, je suis dans le secteur parfait pour ça.

Tu garderas donc un bon souvenir du PHP Day ?

C'est sûr ! Qui peut dire qu'il a fait un barbecue avec le créateur du langage de programmation qu'il utilise tous les jours ?

L'information permanente

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

www.programmez.com



1&1 HÉBERGEMENT ILLIMITÉ

TOUT POUR RÉUSSIR VOS PROJETS WEB

Vous avez de hautes exigences pour votre site Web ? Optez pour un hébergement 1&1 : avec plus de 11 millions de contrats clients, 2 milliards de chiffre d'affaires, 5000 employés et 5 centres de données haute performance en Europe et aux Etats-Unis, nous comptons parmi les leaders mondiaux de l'hébergement. Avec 20 ans d'expérience dans l'hébergement Web et 1500 développeurs, nous sommes votre partenaire privilégié aujourd'hui et demain. C'est pourquoi, avec les packs d'hébergement complets 1&1, vous profitez de nombreux avantages qu'aucun autre hébergeur ne vous propose.



TOUT INCLUS

Espace Web et trafic illimités, 500 comptes email (POP3, IMAP) et FTP, 100 bases de données MySQL (1 Go).



WORDPRESS, DRUPAL, JOOMLA!...

... et de nombreuses autres applications open source populaires. Vous avez le choix parmi 65 applications 1&1 Click & Build à installer en quelques clics.



SÉCURITÉ OPTIMALE

Votre site hébergé en parallèle dans 2 centres de données 1&1 situés à différents endroits (géo-redondance). Sauvegardes quotidiennes.



SERVICE EXPERT 6J/7

Assistance assurée par des professionnels de l'hébergement, via hotline non surtaxée et email.



DOMAINES | EMAIL | HÉBERGEMENT | E-COMMERCE | SERVEURS

* Offre « 1 an à -50 % » soumise à un engagement de 12 mois et à des frais de mise en service de 9,99 € HT (11,95 € TTC). Offre sans durée minimum d'engagement également disponible. Adobe® Dreamweaver® CS5.5 offert pour toute nouvelle commande d'un pack 1&1 Illimité et téléchargeable depuis l'Espace Client 1&1. Conditions détaillées sur 1and1.fr.

1 AN À -50%*

Hébergement tout inclus

1&1 ILLIMITÉ

- Espace Web **ILLIMITÉ**
- Trafic **ILLIMITÉ**
- Accès **ILLIMITÉ** à 65 applications Click & Build (WordPress, Drupal, Joomla!...)
- 3 noms de domaine **INCLUS**
(au choix : .fr, .com, .info, .net, .org, .eu)
- 500 comptes email (POP3, IMAP) et FTP **INCLUS**
- 100 bases de données MySQL (1 Go) **INCLUSES**
- **NOUVEAU : Adobe® Dreamweaver® CS5.5 d'une valeur de 379 € INCLUS***
- **NOUVEAU : 110 € de crédit publicitaire Google AdWords®, Facebook®, Bing™**
- Outil de newsletter et de statistiques
- PHP5, Perl, Python, Zend Framework

3,49
€ HT/mois
(4,17 € TTC/mois)*

~~6,99~~
€ HT/mois
(8,16 € TTC/mois)

EXCLUSIVITÉ 1&1 : ADOBE® DREAMWEAVER® CS5.5 INCLUS

Logiciel professionnel
de webdesign

- **NOUVEAU : Aperçu sur écrans multiples**
- Prise en charge CSS3 et HTML5
- Affichage en direct
- PhoneGap pour applications natives Android et iOS
- Validation W3C

~~379 €~~
(prix de vente indicatif)



1&1



0970 808 911
(appel non surtaxé)

www.1and1.fr

Notre sélection de livres

JACQUES PRINTZ

Architecture logicielle

Dunod

Comment comprendre l'architecture logicielle ? Comment la concevoir ? L'auteur expose en 500 pages les grands principes de définition, comment relier le réel et le modèle. Cette troisième édition propose un nouveau chapitre consacré à un exemple complet de système complexe de la famille C4ISTAR, ainsi que de nouvelles sections concernant l'influence du cloud computing et des architectures des microprocesseurs multicœurs sur l'architecture et ses différents acteurs.

CHARLES BOP

Traité de robotique volume 3

Ellipses

La robotique est un monde complexe à la multiple réalité. L'auteur poursuit son explication et son exploration d'un monde nouveau. Cette 3e partie aborde l'analyse des

capacités d'adaptation, de surveillance et de contrôle par une personne, comment transposer les gestes à un robot. Toutes les commandes sont expliquées : commandes d'axe, géométrique, stochastique, en vitesse, en force, dynamique, par anticipation, ainsi que par logique floue et fuzzy logic appliquée seule ou en alternance à la gestuelle et aux robots mobiles autonomes.



Ses vastes possibilités en font un langage à la fois dépouillé et complexe, mais incontournable, des systèmes de gestion de bases de données relationnelles (SGBDR). Il a été adopté par tous les éditeurs informatiques

qui proposent un moyen efficace et sécurisé de stocker et manipuler des données sur un serveur. Un livre référence avec les meilleurs experts français !

COLLECTIF

Programmation iOS 5

Yrolles



Cet ouvrage, disponible exclusivement en ebook, est centré sur iOS 5 qui introduit de nombreuses évolutions du système aussi bien au niveau fonction que sur

les API. Les nouveautés d'iOS 5 telles que les storyboards et l'ARC (Automatic Reference Counting) dans Objective-C sont traitées. Les fondamentaux du développement iPhone sont détaillés, de l'Objective-C et sa gestion spécifique de la mémoire aux contrôleurs de vue, en passant par la mise en place des vues et des TableView. En attendant iOS 6.0...

» **septembre**

PARISJUG, 11 SEPTEMBRE

Le ParisJug organise sa conférence de rentrée le 11 septembre avec comme thème principal : Performance JVM. William Louth sera présent pour aborder ce problème si important et répondra à 3 questions majeures : mesurer les performances, optimiser et exécuter sous contrôle la JVM. Site officiel :

<http://www.parisjug.org/xwiki/bin/view/Meeting/20120911>

Les autres conférences et sessions techniques du Java User Group en France :

Bordeaux Jug : projet Ceylon (date non fixée), <http://www.bordeauxjug.org/meetings>

Les JUG à l'heure où nous écrivons n'ont pas encore dévoilé les conférences de septembre / octobre. Consultez l'annuaire JUG de France pour en savoir plus :

<http://www.jtips.info/index.php?title=JavaUserGroups>

LE PLEIN D'OBJECTIVE-C

Vous cherchez une conférence technique sur le langage Objective-C ? Lanyrd en organise partout en Europe : Cologne (11 septembre), Leicester (17 septembre), Nottingham (26 septembre). Site : <http://lanyrd.com/topics/objective-c/>

» **Appel à contributions !**

Vous rêvez d'être speaker dans une conférence technique ? 4D et Wakanda cherchent des intervenants pour la conférence JS.everywhere (16 & 17 novembre à Paris). Les thèmes : développement web, Javascript, mobilité... Déposez votre candidature avant le 15 septembre sur : js everywhere.eu/call-to-speakers.

» **Communautés**

Autour du monde Apple et Objective-C

Objective-C reste mal connu en France, il est pourtant au cœur des terminaux Apple. L'association EpiMac / LabMac est très active pour organiser des formations, des conférences autour du langage Apple. La dernière remonte à juin dernier. EpiMac organise régulièrement des sessions et formations. Elle a ses locaux sur le campus de l'Epita près de Paris. Le site propose aussi les vidéos des dernières sessions !

Site : <http://www.epimac.org/>

La communauté des développeurs Windows 8

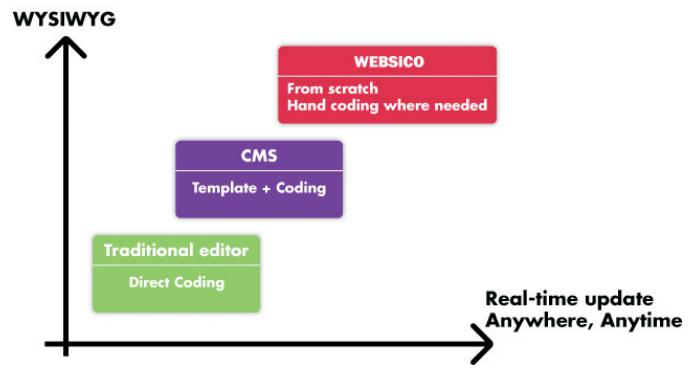


Vous développez sur et pour Windows 8 ? La communauté des développeurs Windows 8 (lancée en juillet dernier) est faite pour vous et entièrement francophone ! Vous aurez accès à des articles techniques, un groupe Facebook avec de nombreuses participations de développeurs, des réunions techniques.

Les liens indispensables : <http://win8dev.fr> et <https://www.facebook.com/groups/192550030818123/>



Websico : un CMS simple comme une souris



Websico est un nouveau logiciel qui s'utilise uniquement à travers internet, pour répondre au mieux aux attentes des utilisateurs qui souhaitent avoir un site de contenu. L'outil respecte les normes et protocoles standard W3C / SEO. Au niveau de l'installation, vous n'avez aucune opération à faire car l'accessibilité s'effectue à distance. L'utilisation de l'outil se fait principalement à la souris, pour plus de simplicité et de légèreté. Ainsi la notion de glisser/coller (Drag and Drop) devient une force, pour déplacer dynamiquement les éléments (textes et images) d'une page. La notion d'éditeur Riche (Wysiwyg) est présente pour travailler directement à travers votre navigateur et obtenir un résultat visuel en temps réel. L'opération s'effectue avec une barre d'icônes, déplaçable dans l'ensemble de votre écran.

Bien entendu, le code source peut être modifié, sous la forme d'une gestion de surcouche, pour répondre aux demandes de certains utilisateurs (designer, graphistes...), c'est-à-dire la possibilité d'insérer vos propres librairies CSS / JS pour personnaliser les pages. Ainsi, chaque page devient unique, tout en ayant la possibilité d'utiliser le multilangage ou l'insertion de widget. Au niveau

de la maintenabilité, vous vous occupez seulement de l'évolution de vos pages et des corrections de votre réalisation. L'outil se charge de gérer le cache, de l'accessibilité des pages et d'effectuer les backups (jusqu'à 30 couches d'archives). La configuration des pages est 100 % modifiable et personnalisable avec un affichage en mode rewriting automatique.

En résumé, même si l'outil est jeune, il se différencie des autres sites web de création en ligne, car il est en constante évolution. Par ailleurs, vous avez l'entièvre liberté de vos réalisations suivant vos envies et sans être tributaire d'une personne externe au projet. Mais dès à présent, vous pouvez aller sur le site de l'application (<http://www.websico.com/fr>) et vous rendre compte des différentes possibilités, (aides, documentation, vidéos) pour découvrir et commencer à utiliser cet outil.



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

1^{er}

intégrateur européen de solutions open source

Livre blanc

Le Guide de l'Open Source



Plus de 150 solutions évaluées et 300 référencées dans 40 domaines d'applications.

A télécharger gratuitement sur www.smile.fr

open
source
www.smile.fr



Jonathan Roquelaure,

« C'est dans l'informatique que j'ai finalement trouvé mon bonheur »

Jonathan, consultant chez Valtech, nous dévoile sa passion pour l'informatique, son travail de consultant mais aussi et surtout, son côté développeur !

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

Enfant, mon grand père me racontait ses voyages en Afrique et en Asie où il avait participé à de grands travaux en tant qu'ingénieur. Toutes ses histoires m'ont donné envie de trouver un travail et de faire une carrière qui me permettrait à moi aussi de voyager, de réaliser des projets concrets et d'envergure. Ayant des facilités en sciences, je suis entré à l'ISEN Toulon ([Institut Supérieur d'Electronique et du Numérique) pour décrocher un diplôme d'ingénieur généraliste. C'est là-bas que j'ai découvert l'informatique parmi de nombreuses autres matières (THS, Physique, Telecom, Electronique,...). J'ai toujours apprécié les jeux de réflexion, les énigmes, bref tout ce qui fait se creuser la tête et, si j'ai hésité à poursuivre vers les

mathématiques appliquées, c'est dans l'informatique que j'ai finalement trouvé mon bonheur. Je me suis aperçu que chaque nouvel algorithme, chaque programme est un problème que l'on peut résoudre de multiples façons et c'est vraiment ce qui me stimulait. J'ai donc décidé de terminer ma formation par une spécialisation en Systèmes Informatiques et un stage de fin d'études au Mexique, alliant ainsi une première expérience du monde du travail et mon désir de voyages.

“ La culture geek fait partie de l'ADN de Valtech ”



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

Comme je l'ai dit, chaque projet, chaque tâche de développement peut se voir comme un nouveau problème, une nouvelle énigme à résoudre. Avec l'expérience, la manière de les aborder est toujours changeante, on prend du recul sur les solutions trouvées précédemment, on trouve des alternatives meilleures ou simplement plus élégantes. Si on ajoute à cela les évolutions technologiques qui arrivent de plus en plus vite et l'intégration du numérique de plus en plus courante dans notre vie de tous les jours, c'est un métier qui est en constante mutation, où l'on peut toujours se renouveler et toujours faire de nouvelles découvertes.

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

Très important, et chez Valtech c'est quelque chose qui est vraiment encouragé et qui fait partie de la culture de l'entreprise. C'est important car cela permet de se renouveler et, comme je l'ai dit, l'informatique est un monde en constante mutation, où l'on a vite fait d'être perdu si l'on ne fait pas l'effort d'observer et d'analyser ces changements. Il suffit pour s'en convaincre de regarder les dernières grosses évolutions, comme la démocratisation des smartphones, l'émergence des réseaux sociaux et sites communautaires, l'arrivée de la domotique, des objets connectés... Aujourd'hui, une entreprise d'envergure à besoin de définir une politique de communication multica-



nal cohérente, et cela passe par l'utilisation des derniers outils, de ce qui fait "le buzz". C'est en étant proactif sur ces évolutions, que l'on peut être force de proposition et permettre à nos clients de s'engager sur les technologies de leur temps et de ne pas prendre un train de retard sur leurs concurrents. Cela est important pour le client mais rend aussi le travail plus passionnant car toujours plein de découvertes.

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 ?

Le job de développeur est de plus en plus éloigné de l'image que l'on avait du barbu dans son garage. Aujourd'hui, je sens une véritable attente des clients pour que l'on soit une vraie force de proposition, tant sur les technos que sur la manière de mener à bien les projets, de modéliser intelligemment le métier afin d'être au plus proche de l'utilisateur final. Cela augmente le champ des responsabilités et peut être considéré comme facteur de pression mais au final cela nous laisse plus de libertés et diminue les frustrations inhérentes à l'exécution bête de tâches sans réflexion d'ensemble. Je crois que le plus difficile aujourd'hui est qu'il faut être en constante veille, savoir communiquer, être à l'écoute du marché bref, ne jamais se reposer sur ses acquis.

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

En dehors du travail j'aime beaucoup voyager. J'ai la chance d'avoir une compagne qui partage ma passion et nous nous en donnons à cœur joie ! Concrètement, on essaie de faire au moins un gros voyage par an et nous sommes revenus en août dernier d'une année sabbatique en Nouvelle-Zélande et Asie du Sud-Est. Il m'arrive aussi de travailler à des projets plus ou moins pro en dehors de mon travail chez Valtech. C'est ainsi que j'ai participé au développement du site Tableonline dont je suis toujours actionnaire. J'ai aussi la chance d'être dans une entreprise qui



Au bureau...

Je travaille presque tout le temps en musique, écouteurs fixés sur les oreilles et internet sur une webradio comme LaGrosseTambouille ou sur le mixcloud. En ce moment je suis assez gâté puisque j'ai un poste très performant avec deux beaux écrans.

essaie de garder des valeurs humaines et d'établir de forts liens entre les employés (sorties régulières subventionnées au ski, canyoning, barbecue, un lipdub qui a été primé...). Ainsi, je passe beaucoup de soirées avec des collègues de bureau qui sont finalement devenus des amis. Sinon, je fais de la musique (trompette, didgeridoo), je sors pas mal (concerts, fêtes, théâtre...), car Toulouse est une ville qui bouge énormément.

week-end on en profite pour aller s'oxygénier ou faire un peu de sport.

Comment vois-tu ton job évoluer ?

J'ai eu la chance, il y a deux ans, d'être Tech Lead sur un projet pour Airbus et j'ai vraiment apprécié le mélange expertise technique et encadrement d'équipe (assurer la montée en compétences des nouveaux, encadrement...), j'aimerais donc m'orienter vers un poste de ce style, prendre plus de responsabilité dans la gestion des projets, les choix... mais sans laisser de côté l'aspect purement technique. D'un autre côté, les entreprises prennent de plus en plus conscience de l'importance d'une véritable stratégie digitale et cela induit certains changements : une approche plus globale des projets qui étaient auparavant purement techniques, un aspect sociologique qui va, selon moi, prendre de plus en plus de poids pour toucher de plus en plus efficacement les utilisateurs finaux.

Des conseils aux étudiants et dévs qui nous lisent ?

Soyez vous-mêmes, suivez les tendances, lisez des blogs, des magazines d'informatique,... mais n'oubliez jamais que vous aurez beau être le meilleur, si vous ne savez pas vous vendre vous et vos idées, vous ne serez jamais considérés à votre juste valeur. Nous ne sommes plus à l'âge d'or de la bulle internet mais l'informatique à encore de belles années devant elle !

“ Le job de développeur est de plus en plus éloigné de l'image que l'on avait du barbu dans son garage. ”

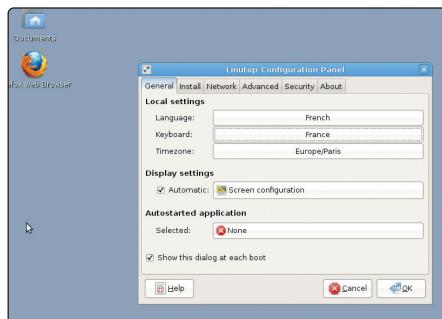
Linutop 4.0, un couteau suisse pour geeks

Linutop est un petit ordinateur sans ventilateur, donc totalement silencieux, propulsé par Linux. Présenté comme un appareil à destination des entreprises pour mettre en place des bornes Internet ou des affichages dynamiques, il est en fait bien plus intéressant que cela. A y regarder de plus près, il se révèle être un magnifique jouet pour geek. Découverte.

Linutop est un petit ordinateur de type PC qui tient dans un boîtier de 20x18x3.5 cm. En façade se trouve un interrupteur marche/arrêt et un bouton de reset, une prise micro, une sortie son et deux ports USB. A l'arrière, nous trouvons une deuxième sortie son, et une sortie vidéo VGA + DVI, résolution maximum 1920x1440. Nous trouvons encore 3 ports USB dont il est à regretter qu'ils soient un peu trop proches les uns des autres. Enfin, à l'arrière toujours, vous trouverez une prise RJ 45 pour vous connecter à une carte Ethernet 1000 Base-T. Encore un regret à ce niveau. Deux ou même trois interfaces au lieu d'une seule Ethernet décupleraient les possibilités du Linutop, par exemple pour en faire un routeur. L'alimentation électrique est dans un boîtier externe

> Prise de contact

Le démarrage du système est très rapide, en raison de sa légèreté et de l'absence de disque. Le système d'exploitation est à la base une distribution Linux Ubuntu adaptée aux usages de cette machine. Au démarrage vous arrivez dans un bureau Gnome et on vous présente d'emblée un panneau de configuration. La philosophie de Linutop est



Le bureau de Linutop

de faire que l'utilisateur soit immédiatement opérationnel. Ainsi avec un clic de souris dans ce panneau, vous transformerez votre Linutop en "Kiosque", c'est-à-dire une machine qui présente un affichage dynamique, par exemple pour faire la démonstration d'un produit dans un centre commercial. Il y a des possibilités inédites et sympathiques dans ce panneau, comme celle de planifier un redémarrage par jour, afin de ne pas encourrir un risque d'arrêt du système à cause d'une application qui présenterait une fuite mémoire. La documentation papier livrée avec le produit est une peau de chagrin. Allez vite dans le menu Linutop et sélectionnez l'aide en ligne qui est beaucoup plus fournie. Se connecter à Internet est très simple, il suffit pour cela de connecter le Linutop à la box de votre fournisseur d'accès, et tout le reste sera automatique grâce à la présence d'un client DHCP pré-installé. Le panneau de configuration vous permettra aussi de sauvegarder votre système, données incluses ou non, au choix, sur une clé USB bootable, y compris sur un PC quelconque. Une opéra-



tion à ne pas négliger de faire régulièrement pour ne pas vous faire piéger par une carte Flash qui fatiguerait.

> Un serveur LAMP

Un autre clic sur le panneau de configuration installera un serveur LAMP sur votre Linutop, c'est-à-dire le trio Apache, PHP et MySQL. Ainsi vous disposez d'un serveur de développement très sympa et même plus. En effet, si la box qui vous relie à Internet supporte DynDNS, vous pourrez mettre en place votre site Internet à la maison, avec une machine qui consomme très peu d'énergie.

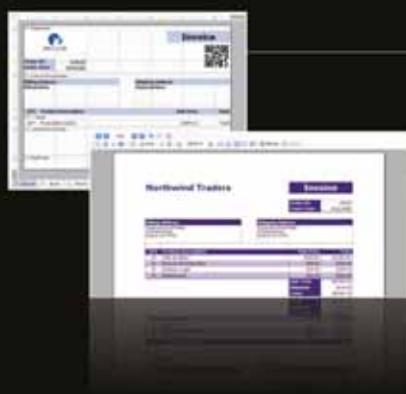
> Et plus

Il est possible d'aller beaucoup plus loin avec Linutop, qui n'est pas du tout une machine que l'on doit considérer comme dédiée, et surtout limitée, à des bornes. On peut, par exemple, en faire un serveur de fichiers ou de contenu multimédia pour le réseau à la maison. Puisqu'il est possible d'installer des applications, on peut très bien imaginer en faire un serveur de courrier centralisé, sur lequel vous rapatrieriez tous les mails de tous vos comptes et que vous pourriez consulter de n'importe où, dans la mesure où vous êtes sur Internet via DynDNS. Simplement dans ce cas, on remplacera la carte Flash par un disque, pour éviter une usure prématuée en raison des opérations d'écriture. On peut aussi faire du Linutop... un ordinateur. On connecte un écran, une souris et un clavier USB, et c'est parti ! Vous disposez de Firefox et d'OpenOffice qui sont préinstallés. Linutop sera alors un poste de travail original, économique en énergie, silencieux tout en étant suffisamment puissant, et surtout pas encombrant. Votre travail terminé, vous le déconnectez de votre télévision et vous le rangez dans un tiroir. Sympathique ! :-)

Site : www.linutop.com

Frédéric Mazué

fmazue@programmez.com



ActiveReports 7 à partir de € 1 292



Moteur de rapports rapide et flexible encore plus performant..

- Rapports à partir de formulaires, des rapports de transaction/analytique
- Nouveau concepteur de pages pour une génération facile
- Plus de flexibilité de présentation avec tables, matrices, listes et graphiques
- Personnalisation étendue avec Flexible .NET API



GdPicture.NET à partir de € 3 233



Outils complets d'imagerie documentaire et de gestion pour les développeurs.

- Capturez, traitez, créez, affichez, modifiez, annotez, composez, fractionnez, fusionnez et imprimez des documents depuis vos applications Windows et Web
- Lisez, écrivez et convertissez les images vectorielles et raster en plus de 90 formats, dont PDF, PDF/A, TIFF, GIF, JPEG, PNG, JBIG2, WMF, BMP, WBMP, ICO, PCX, PNM, XPM, JPEG 2000, HDR, PSD, TGA, PICT, EXR, DDS, PPM, SGI, PBM, PGM, PFM, XBM, IFF et le format graphique RAW



Aspose.Total for .NET à partir de € 2 020



Tous les composants Aspose .NET réunis dans un seul package.

- Programmez la gestion de formats courants, tels que Word, Excel, PowerPoint et PDF
- Incluez des graphiques, des e-mails, des correcteurs, des codes barres, l'OCR, des diagrammes, des images et la gestion de projets/formats dans les applications .NET
- Les principales utilisations sont la fusion de messages, l'ajout de codes barre, la création de rapports Excel dynamiques à la volée et l'extraction de texte des fichiers PDF



dotConnect Data Providers à partir de € 129



Fournisseurs ADO.NET (Oracle), MySQL, PostgreSQL, SQLite et Salesforce.

- Support d'ADO.NET Entity Framework 1 et 4, LinqConnect et NHibernate pour Oracle, MySQL, PostgreSQL et SQLite, concepteur visuel pour modèles ORM
- Intégration totale : Support BIS et Enterprise Library, fournisseurs ASP.NET...
- Intégration Advanced Visual Studio avancée, outils et éditeurs de conception riches, support de fonctions de BD spécifiques et bien plus encore

Les nouveautés du Windows Azure SDK 1.7

Lors de la conférence *Meet Windows Azure* qui s'est déroulée à San Francisco le 7 juin dernier, nous avons découvert les nouveautés de la plateforme Cloud de Microsoft.

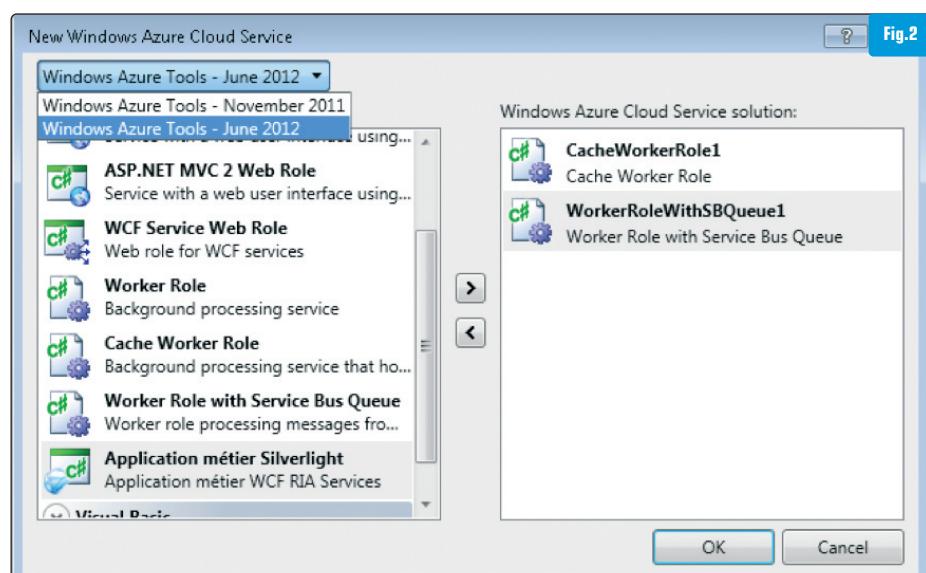
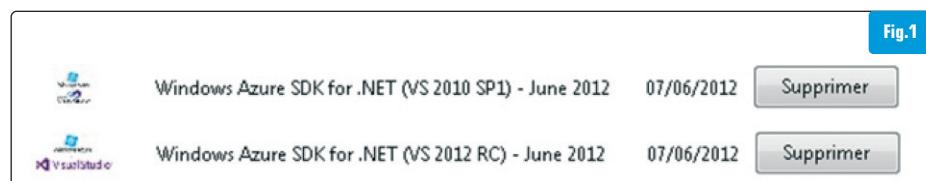
Avant tout, une offre de type « Infrastructure As A Service » (IaaS) grâce au support des machines virtuelles, mais l'offre de type « Platform As A Service » (PaaS) s'est enrichie avec l'arrivée de la fonctionnalité « Web Site » qui permet d'héberger rapidement un site internet soit via une galerie qui contient des applications telles que WordPress ou Drupal, mais aussi via du développement réalisable sur Web Matrix ou Visual Studio. Cependant, lors de cette conférence, le nouveau SDK Windows Azure a été annoncé, nous allons donc nous y intéresser.

Avant de parler des nouveautés du SDK, il faut l'installer. Pour cela il est possible soit d'aller sur les sites de téléchargement de Microsoft, soit via le logiciel Web Platform Installer. Ici, dans notre cas, nous allons installer le SDK sur une machine ayant déjà la version 1.6 du SDK Windows Azure, elle a de plus à sa disposition Visual Studio 2010 Ultimate, et Visual Studio 2012 RC Ultimate, le tout installé sur une machine avec Windows 7 x64.

Installation

Il est possible de trouver les exécutables d'installation via le logiciel Web Platform Installer, ou via le site de téléchargement de Microsoft, comme nous pouvons le voir, il existe deux versions du SDK 1.7, une pour Visual Studio 2010 SP1, et une pour Visual Studio 2012 RC [Fig.1].

Outre le fait qu'on puisse à ce jour installer la dernière version du SDK sur les deux environnements de développement, il est doré-



navant possible d'installer cette version en parallèle avec la précédente sur Visual Studio 2010. A ce jour, il est donc possible de

maintenir les deux derniers SDK sur un seul poste, ce qui vous permettra de réaliser des montées de versions de vos applications tout en gardant la possibilité de réaliser des modifications sur la version précédente.

Il est bon de noter, que malgré le fait qu'on puisse à ce jour installer le SDK Windows Azure sur Visual Studio 2012 RC, la plateforme ne supporte toujours pas nativement la version 4.5 du Framework .Net, il vous faudra donc pour le moment utiliser la version 4.0 pour pouvoir utiliser des projets de type Windows Azure Cloud Service.

Dans Visual Studio 2010, lorsque vous créez un nouveau projet de type « Windows Azure Cloud Service », il est possible de choisir la version du SDK que vous voulez utiliser, soit celle de Novembre 2011 qui correspond à la ver-

sion SDK 1.6, et la version de Juin 2012 qui est la version actuelle. Cette dernière apporte deux nouveaux templates de projet qui sont le « Cache Worker Role » et le « Worker Role with Service Bus Queue » comme on peut le voir [Fig.2].

Ces deux nouveaux templates apportent donc leurs lots de nouveautés, ainsi que le support de nouvelles fonctionnalités, le premier va utiliser le nouveau système de cache qui est actuellement en preview, le second va, quant à lui, fournir un modèle simple d'utilisation des nouvelles queues du Service Bus, ce qui peut être apprécié lorsqu'on débute sur Windows Azure, et que l'on souhaite expérimenter soi-même les diverses fonctionnalités.

Cache Worker Role

Ce modèle de projet, nous permet donc d'utiliser la version de cache de Windows Azure SDK 1.7, à noter qu'elle est différente de celle que l'on pouvait utiliser avant qui s'appelait Windows Azure AppFabric Cache.



Windows Azure™

Comme on peut le voir ci-contre, un nouvel onglet est apparu dans la configuration de nos rôles, celui-ci permet d'activer ou non la fonctionnalité de Cache [Fig.3].

On peut néanmoins remarquer que nous ne sommes pas obligés d'utiliser un rôle dédié pour le cache, mais qu'il est possible d'utiliser un rôle existant afin d'y ajouter une fonctionnalité de Cache. Ce modèle de projet, nous permet donc uniquement d'avoir un début de projet avec un cache distribué pré-configuré par Visual Studio.

Worker Role & Service Bus Queue

Contrairement au modèle de projet précédent, celui-ci propose une implémentation simple de l'utilisation des Services Bus Queue.

En effet, dans la classe WorkerRole.cs, nous avons au sein de la méthode OnStart(), la configuration de la queue à utiliser dans notre application, et dans la méthode Run(), une boucle infinie qui nous permet de lire les messages de la queue configurée au sein du rôle.

> Support de IIS Express

Bonne nouvelle pour ceux qui ne veulent pas installer un environnement IIS sur leurs postes de développement, avec la nouvelle version de Windows Azure, il est possible d'utiliser IIS Express pour tester en local vos développements.

Pour utiliser IIS Express, il vous suffit de l'installer, et ensuite de s'assurer que vos projets Windows Azure l'utilisent. Pour cela, il vous suffit d'afficher les propriétés de votre projet de type Windows Azure Cloud Service, et de sélectionner IIS Express dans la partie Web comme on peut le voir [Fig.4].

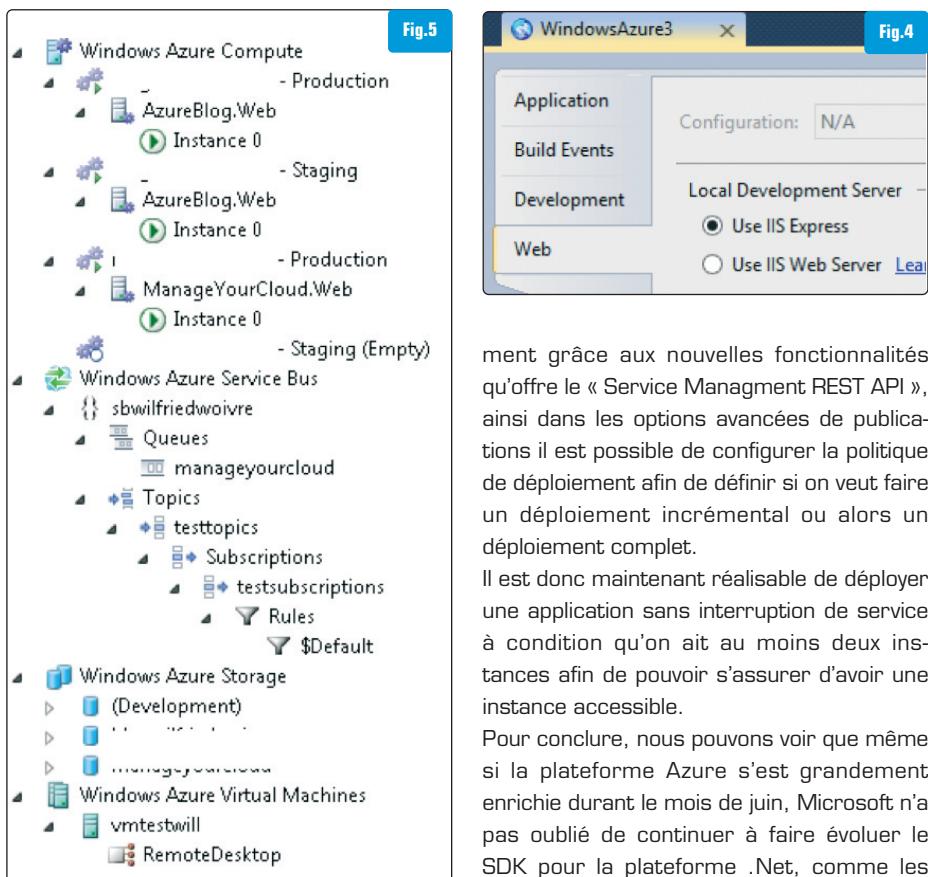
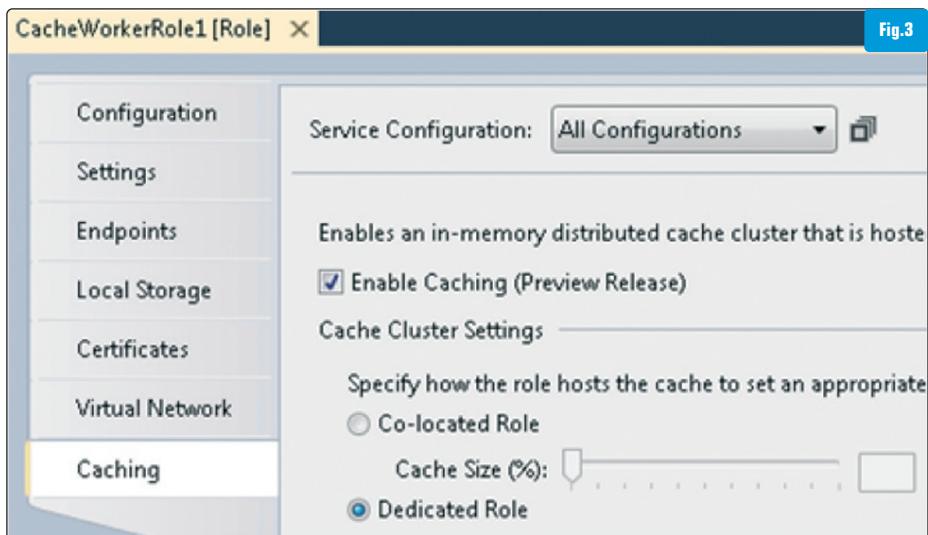
Pour ceux qui ne connaissaient pas IIS Express, le principal avantage de celui-ci est qu'il consomme beaucoup moins de ressource pour son fonctionnement, vous n'aurez donc plus besoin d'une machine très performante pour émuler vos projets.

> Explorateur de serveur

Avec cette nouvelle version du SDK, l'explorateur de serveur a été modifié pour intégrer les nouveautés de la plateforme, comme on peut le voir [Fig.5].

Dans cette nouvelle version, il a donc été intégré le support des machines virtuelles avec la possibilité de rapidement se connecter en connexion à distance.

Et en effectuant un clic droit sur une instance



ce déployée sur Windows Azure, il est possible de se connecter via la connexion à distance.

De plus, la gestion du Service Bus a été améliorée, il est ainsi plus simple de créer de nouvelles queues ou topics, il est possible également d'envoyer et de recevoir des messages directement depuis Visual Studio.

> Processus de publication

Dans cette nouvelle version du SDK, le processus de publication a été amélioré, notam-

ment grâce aux nouvelles fonctionnalités qu'offre le « Service Management REST API », ainsi dans les options avancées de publications il est possible de configurer la politique de déploiement afin de définir si on veut faire un déploiement incrémental ou alors un déploiement complet.

Il est donc maintenant réalisable de déployer une application sans interruption de service à condition qu'on ait au moins deux instances afin de pouvoir s'assurer d'avoir une instance accessible.

Pour conclure, nous pouvons voir que même si la plateforme Azure s'est grandement enrichie durant le mois de juin, Microsoft n'a pas oublié de continuer à faire évoluer le SDK pour la plateforme .Net, comme les autres SDK pour les plateformes telles que Java. Dans cette dernière mise à jour on a pu donc voir une intégration des nouvelles fonctionnalités dans Visual Studio, de plus le support de IIS Express permet d'obtenir un émulateur plus léger, et pour finir la simplification du processus de publication permet de garder la disponibilité de votre application dans le Cloud.

Wilfried Woivré

<http://blog.woivre.fr> - <http://zecloud.fr>

wilfried@zecloud.fr - MVP Windows Azure

Eclipse Juno : le nouvel Eclipse

Comme tous les ans, le mois de juin est marqué par la sortie de la nouvelle mouture d'Eclipse. C'est donc Juno qui succède à Indigo. Le 27 juin dernier a été l'occasion de la sortie simultanée de 72 projets portés par 445 committers pour un total de 55 millions de lignes de code !

Plus de deux semaines plus tard, Eclipse Juno dépassait déjà le cap du million de téléchargements. Intéressons nous alors aux nouveautés apportées par Juno et qui expliquent l'engouement autour de la plateforme Eclipse.

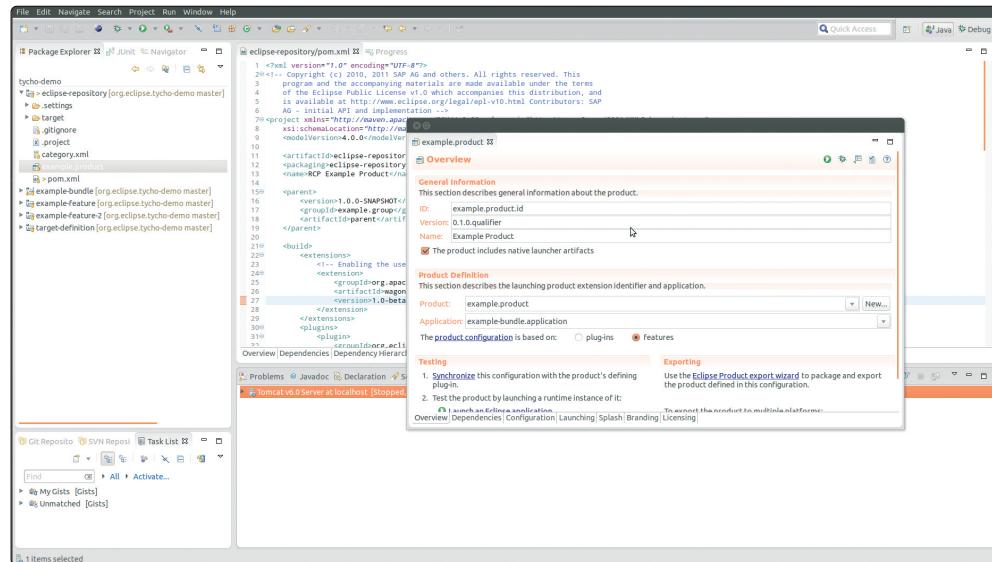
> Eclipse 4.2 ou Eclipse 3.8 ?

Vous l'avez peut-être remarqué, mais la release de Juno est un peu particulière. En effet, depuis la sortie de Eclipse 3.0 en juin 2004, la numérotation a suivi son bonhomme de chemin jusqu'à aujourd'hui et la version 3.8. Cependant, une version 4.2 est aussi disponible. Ce changement de numéro de version majeure reflète une évolution fondamentale de la plateforme Eclipse, autour de laquelle est construite Juno.

> La plateforme Eclipse 4 et Juno

Trois ans ! C'est le temps qu'il aura fallu pour mettre au point la nouvelle plateforme Eclipse : E4AP (Eclipse 4 Application Platform). Bien qu'il n'apporte que peu de changements pour l'utilisateur final de Juno, ce chantier aura été l'occasion de repenser la façon de développer des applications de type RCP (nous aurons l'occasion d'y revenir dans le prochain numéro de Programmez !). Notons cependant un certain nombre d'améliorations telles que le nouveau look'n'feel plus épuré et pensé de manière à ce que l'attention de l'utilisateur se porte sur les parties actives de l'IDE.

Ajoutons à cela la possibilité de placer les éditeurs et les vues sans contraintes, la barre de recherche universelle permettant un accès rapide à n'importe quelle commande ou fichier, la gestion des éditeurs détaillés, ou encore la surprenante possibilité de styler Eclipse au runtime via un inspecteur CSS ! La plateforme Eclipse 4 a donc permis un grand bond en avant en termes d'utilisabilité, et ce n'est que le début : http://help.eclipse.org/juno/index.jsp?topic=%2Forg.eclipse.jdt.doc.user%2FwhatsNew%2Fjdt_whats-new.html



> Les nouveaux projets

A coté du travail titanique abattu par les committers de la plateforme Eclipse 4, d'autres équipes de projet nous ont réservé quelques belles nouveautés.

> Code Recommenders

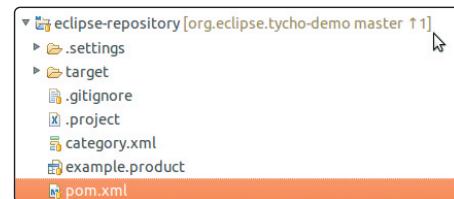
Lauréat du prix du « Nouveau projet le plus innovant » lors de la dernière EclipseCon, Code Recommenders apporte une touche d'intelligence dans l'auto-complétion.

Les frameworks à maîtriser sont toujours plus nombreux, et les bonnes pratiques parfois obscures. Grâce à ses algorithmes d'apprentissage, Code Recommenders a analysé les sources de nombre de projets open source afin d'en extraire les usages les plus répandus des API ainsi que les patterns de bonne pratique. Il faut aussi y ajouter l'enrichissement de la documentation par des exemples de code, des snippets de code contextuels, etc. Imaginez à quel point son usage peut s'avérer productif !

<http://www.eclipse.org/recommenders/>

> Orion, pour le web, sur le web

Orion est un nouveau projet destiné à fournir un environnement de développement sur le web et pour le web. Alors soyons clairs, il ne s'agit pas d'embarquer Eclipse dans un navigateur, mais bien d'un tout nouveau projet



codé en Javascript. Orion est résolument pensé pour exploiter les capacités du navigateur, ainsi chaque fichier ouvert est un nouvel onglet dans le navigateur, pour revenir en arrière il suffit d'utiliser le bouton précédent, etc. La sortie de Juno a vu aussi celle de la version 0.5.0 d'Orion, c'est donc un projet encore très jeune. Mais il est bon de noter qu'il dispose d'un ensemble de fonctionnalités intéressantes telles que l'intégration à Git, l'auto-complétion et la coloration syntaxique pour les langages Javascript et HTML et un système de plugins qui permet d'étendre la plateforme Orion.

<http://www.eclipse.org/orion/>

> Java : support de Java 7 et autres sucqueries

Le projet Java Development Tools (JDT) intègre désormais les structures syntaxiques de Java 7. Ainsi, les diamond operators, le multi-catch, les switch/case sur des chaînes de caractères sont désormais supportés par l'éditeur. Il faut aussi rajouter un meilleur support du surlignage des accolades, une meilleure analyse statique pour

les fuites mémoire, etc. Il est aussi bon d'indiquer que le JDT dispose désormais d'un ensemble d'annotations permettant l'analyse statique du contrôle des valeurs nulles. (@NotNull, @Nullable, @NonNullByDefault). http://help.eclipse.org/juno/index.jsp?topic=%2Forg.eclipse.jdt.doc.user%2FwhatsNew%2Fjdt_whats-new.html

> Des langages, toujours plus de langages

Eclipse étant extensible, de nombreux projets de support de nouveaux langages sont annoncés. Citons tout d'abord le projet Koneki qui offre un environnement de développement pour les développeurs Lua. Lua est un langage de script très utilisé dans l'industrie du jeu vidéo et dans l'informatique embarquée notamment.

<http://www.eclipse.org/koneki/>

Xtend est un des nombreux langages s'exécutant sur la machine virtuelle Java (JVM) dont l'objectif est de pallier certaines lacunes et faiblesses de Java. Contrairement à Scala, Groovy ou encore Ceylon, Xtend ne produit pas directement du bytecode interprétable par la JVM, mais des classes Java qui seront ensuite compilées. Xtend est un

langage typé dynamiquement et qui apporte un ensemble de fonctionnalités manquantes dans Java. Citons pêle-mêle le support des lambda expressions, la surcharge d'opérateurs pour n'importe quel type d'objets et les templates expressions qui permettent de faire du templating simple et concis.

<http://www.eclipse.org/xtend/>

> La fondation Eclipse

La fondation Eclipse est l'organisme non lucratif qui gère Eclipse. Elle a pour but de promouvoir la communauté open source, de s'assurer de la création d'un ensemble d'outils et projets complémentaires et aussi de la conformité des droits de propriété intellectuelle des projets. Elle s'assure aussi de la mise à disposition des infrastructures web et de développement aux projets Eclipse.

> Industry working groups

L'usage de logiciels/frameworks open source s'est largement démocratisé et il n'est pas rare de rencontrer des industriels en faisant un usage intensif.

Pour eux, l'informatique n'est qu'un moyen de créer leurs produits, leurs services. Il est donc primordial qu'ils puissent avoir à la fois

un pouvoir de décision sur l'orientation future des technologies qu'ils utilisent et une assurance d'un support à plus long terme qu'à l'accoutumée en informatique. C'est ainsi que s'est créé un ensemble d'« Industry Working Groups » (IWG) au sein desquels collaborent les membres de la fondation à la promotion et l'adaptation de la plateforme Eclipse pour les besoins spécifiques des industries. Parmi ces dernières, citons l'automobile, l'informatique embarquée et un IWG plus générique: le Long Term Support (LTS). L'IWG LTS a pour but de fournir un support à long terme (années) et à très long terme (décennies) pour les projets Eclipse. En effet, il faut être conscient que du code basé sur les projets Eclipse est en production en orbite autour de la Terre dans des satellites par exemple.

<http://www.eclipse.org/org/industry-workgroups/>

> Common Build Infrastructure (CBI) : buildez votre Eclipse chez vous !

Afin d'assurer le Long Term Support, Eclipse a besoin d'un processus maîtrisé et reproduitible de construction de sa ligne produits (différentes versions, plateformes, etc.). C'est l'objectif de CBI.

CBI est donc un ensemble d'outils et de bonnes pratiques visant à permettre de builder la plateforme Eclipse de manière très simple. Cet outillage se compose, entre autres, de Maven, Tycho et Hudson.

> CVS, la fin d'une ère. Git, le début d'une nouvelle.

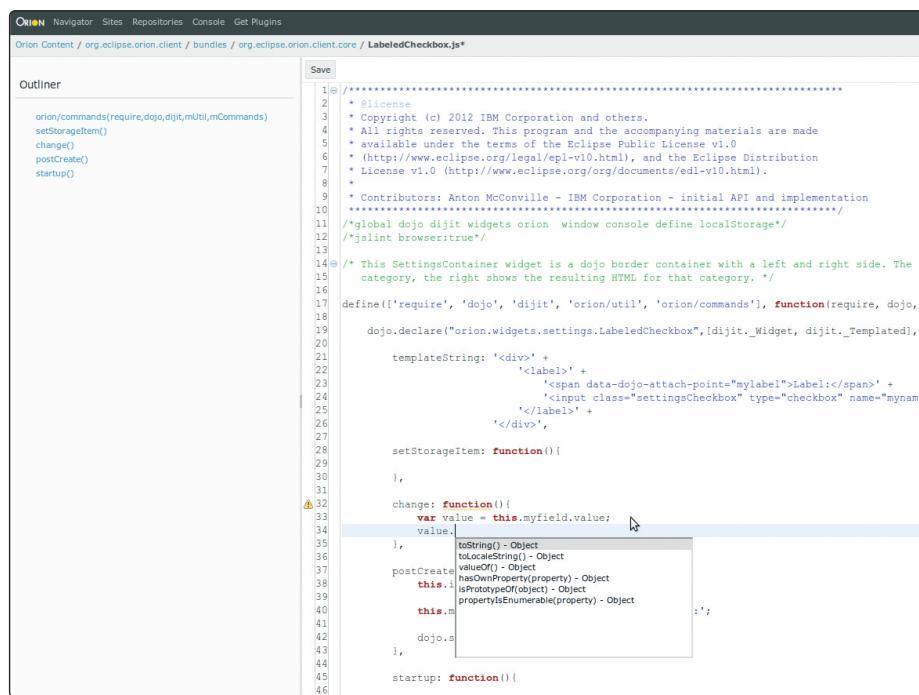
Effectivement le 21 décembre 2012 sera la fin d'une ère. Mais pas celle que vous croyez ! Cette date marquera officiellement la fin du support de CVS comme gestionnaire de version pour les projets d'Eclipse. Mais il ne s'agit pas de laisser les committers sans une alternative viable et bien outillée. Celle-ci s'appelle EGit, le plugin d'intégration du gestionnaire de version Git dans Eclipse. Parmi les nouveautés incluses dans Juno, nous trouvons la décoration des branches afin de voir les divergences entre le repository Git local et le repository distant. Citons aussi le support de la commande de mise en cache (git stash) des modifications non committées. Eclipse nous offre toujours plus de fonctionnalités alors, qu'attendez-vous pour passer à Juno ?

Xavier Seignard

Développeur Eclipse indépendant

xavier_seignard@gmail.com

```
public void createInitialLayout(IPageLayout layout) {
    layout
    ...
    createFolder(String folderId, int relationship, float ratio, String refid) : IFolderLayout - 32 %
    getEditorArea() : String - IPageLayout - 32 %
    ...
    addActionSet(String actionSetId) : void - IPageLayout - 22 %
    addShowViewShortcut(String id) : void - IPageLayout - 21 %
    addNewWizardShortcut(String id) : void - IPageLayout - 19 %
    addPerspectiveShortcut(String id) : void - IPageLayout - 12 %
    setEditorAreaVisible(boolean showEditorArea) : void - IPageLayout - 10 %
    ...
}
```





Premiers pas avec Wakanda

2^e partie

Dans le précédent épisode, nous avons créé notre premier DataModel Objet, observé l'API REST exposée et manipulé les données avec le DataBrowser intégré. Nous allons maintenant créer une interface utilisateur avec Wakanda Studio, pour le desktop et le mobile.

La DataClass « Todo » créée la dernière fois a été un peu modifiée pour l'occasion, nous lui avons ajouté des attributs (priority, creationDate, ...) et du code d'initialisation et de validation. Cet article étant consacré à l'interface, nous ne nous étendons pas sur ces changements que vous pourrez analyser en regardant le modèle et son script (<http://github.com/thibarg/wakanda-todo-programmez>) [Fig.1].

> WAF WAF ! Va chercher les données

Nous avons déjà créé des données, et souhaitons les rendre accessibles. En utilisant le *GUI Designer* du studio, nous allons réaliser une **interface HTML5 / CSS3**, en mode **WYSIWYG**.

Avant cela, parlons du côté **client Ajax du Framework de Wakanda (WAF)**. Automatiquement chargé dans le navigateur de l'utilisateur, il fournit un **dataprovider**, des **datasources**, et des **widgets**. Il crée une couche d'abstraction permettant, littéralement, de casser la barrière qui existe entre une interface cliente et ses données sur le serveur. Le dataprovider utilise l'**API REST de WakandaDB** afin de récupérer les données et leur modèle. Les datasources l'utilisent pour alimenter les widgets en données (le « C » de la pattern MVC) et permettent de partager un état entre plusieurs

widgets. Une datasource peut également être locale en étant associée à une variable JavaScript (un tableau, un objet, type scalaire), mais dans cet article, nous nous concentrerons sur les datasources dites serveur.

> Déclarer, lier : la simplicité par l'exemple

La DataClass « Todo » étant définie (attributs, règles de validation et de sécurité, ...), nous allons pouvoir y lier un widget d'interface. En l'occurrence, nous allons afficher la liste de Todo de l'utilisateur courant dans un widget *Grid*.

Une fois le Studio lancé, un clic sur le bouton « Main page » de la barre d'outils ouvre le fichier « index.html » avec le *GUI Designer*, qui se trouve dans le dossier « WebFolder » [Fig.2].

Le *GUI Designer* est divisé en cinq parties. Une classique barre d'outils, en haut. Une zone d'édition au centre, flanquée à gauche par les widgets disponibles (vues « Widgets » et « Outline ») et les sources de données (vues « Model » et « Datasources »), et à droite par les propriétés, dont le contenu s'adapte en fonction de ce qui est sélectionné dans la zone d'édition.

Nous commençons par glisser-déposer une *Grid* depuis la liste de widgets vers la zone d'édition [Fig.3].

Pour la lier aux données de la DataClass *Todo*, nous procéderons en trois étapes. La première consiste à faire un clic droit sur la DataClass *Todo* pour créer la datasource qui lui sera liée : [Fig.4].

Le *GUI Designer* active alors l'onglet « Datasources », qui affiche la datasource que nous venons de créer. Par défaut, une datasource server prend le nom de la DataClass, en mettant sa première lettre en minuscule. Nous avons ainsi la datasource *todo*, liée à la DataClass *Todo* [Fig.5].

La seconde étape consiste à sélectionner les attributs que l'on souhaite afficher dans la Grid. Nous déployons la datasource *todo*, et sélectionnons (ctrl-clic avec Windows, cmd-clic avec Mac) les attributs « *label* », « *done* », et « *priority* ».

Enfin, la troisième et dernière étape consiste à glisser-déposer ces attributs sur la Grid [Fig.6].

Si l'accès aux données était complètement ouvert, nous en aurions terminé : un clic sur le bouton d'exécution nous permettrait de visualiser tout de suite les données, quelle que soit leur quantité : 10, 100, 10 millions de lignes. Dans notre exemple, le code écrit - côté serveur - pour l'événement *onRestrictingQuery* filtrera les données envoyées. En l'occurrence, puisque nous ne serons pas connectés, la grille restera vide. Nous ajoutons donc le widget « *login* dia-

Fig.1

Fig.2



log », et pouvons tester l'accès aux données. Le but de ce chapitre est de démontrer la force du WAF : nous allons très bientôt nous occuper de créer des pages plus professionnelles, contenant autre chose qu'une grille sur fond blanc. Pour le moment, contentons-nous de cocher la propriété « resizable », de telle sorte que nous puissions retailler la grille dans le navigateur pour afficher toutes les colonnes. Nous cliquons à présent le bouton « Run file » [Fig.7]. Si le serveur n'était pas lancé, le studio s'en charge, puis le navigateur ouvre la page. Cliquez sur le lien « Login ». Les login et mot de passe sont « john » et « pass ». Une fois identifié, rechargez la page (nous verrons plus loin comment afficher les données siège que l'utilisateur est connecté) et admirez les tâches de John [Fig.8].

Nous avons illustré un des points clés de Wakanda. Nous avons simplement déclaré notre modèle, puis l'avons lié à un widget (ici la *Grid*) : *Declare and Bind*. Wakanda s'est chargé de tout le reste : envoyer une requête REST asynchrone au serveur, récupérer les données, les distribuer dans la grille. Il est temps de construire une page qui soit un peu plus jolie.

> Lier et super-lien avec le WAF

Datasources/Dataprovider ne se contentent pas de considérablement simplifier l'accès

à des données sur le serveur et leur affichage dans des widgets. Un système d'abonnement automatique permet de centraliser les actions : si plusieurs widgets sont liés à la même datasource, ils seront tous mis à jour lorsque les données liées à cette source viendront à changer (pattern *Observer*, ou *Pub/Sub*). Un exemple typique est celui d'une page affichant une grille et un formulaire de saisie des détails. Ici on pourrait imaginer une grille ne contenant que « *label* » et « *done* », et à côté de cette grille un formulaire contenant tous les autres attributs. Dans cet autre formulaire, un widget *Text Input* est lié à l'attribut « *label* », et un widget *Checkbox* est lié à « *close* ». Nous avons donc des widgets différents liés à la datasource *todo*. Dans ce cas, la sélection d'une ligne de la grille met à jour automatiquement les données affichées dans le formulaire. Et vice versa : si l'utilisateur modifie le libellé ou coche/décoche la case dans le formulaire détaillé, la ligne sélectionnée dans la grille est automatiquement mise à jour.

> Tout le monde est expert CSS3

Le *GUI Designer* permet d'appliquer des styles aux widgets en cochant des cases, en déplaçant des curseurs ou en choisissant des couleurs dans une palette : un fichier *.css* est automatiquement créé (dans le dossier *styles*) et appliqué à la page html, via le *WAF Loader* (qui sera aussi utilisé pour charger des scripts, ou d'autres feuilles de style). Chaque widget, ou sous-widget, peut être formaté selon ses besoins, ses envies, ou sa charte. Le principe de fonctionnement est simple : On clique sur le widget que l'on souhaite modifier, puis on change son apparence dans l'onglet « *styles* » : avant-plan, arrière-plan, bordure, ombre, gradient, position, ...

Mais avant cela, on peut appliquer un *skin* à la page toute entière. Par exemple, tout en restant sur la page d'index, cliquez dans une zone d'édition vide pour sélectionner le document lui-même, puis activez l'onglet « *Skins* » et testez différents *skins* : vous verrez que le *look & feel* de la grille (et du *login dialog*) change, ce qui vous permettra d'appliquer un style global, par défaut, à chaque élément.

> Monter en grade

Donnons un fond à notre page. Le fond se règle dans l'onglet « *Styles* ». On peut utiliser une image, simplement en glissant-déposant son fichier dans la zone « *image* » : elle sera alors recopiée dans « *WebFolder/images/* », et référencée dans la balise *css* « *background-image* », avec éventuellement ses propriétés de répétition.

Nous allons plutôt colorer le fond de la page avec un gradient bleu. Il s'agit de cliquer le gradient de départ, de choisir une couleur dans le *picker* qui s'affiche, puis de valider la couleur choisie [Fig.9]. On procède de la même façon pour le gradient d'arrivée.

Une fois votre gradient réglé, vous pourrez constater (en regardant le fichier « *styles/index.css* ») que Wakanda a généré le code CSS3 des gradients adaptés aux principaux navigateurs, puisque cette pro-

Fig.2

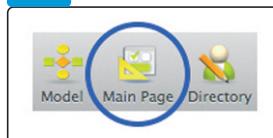


Fig.4

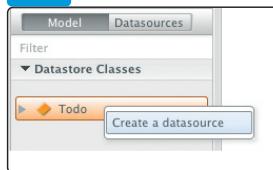


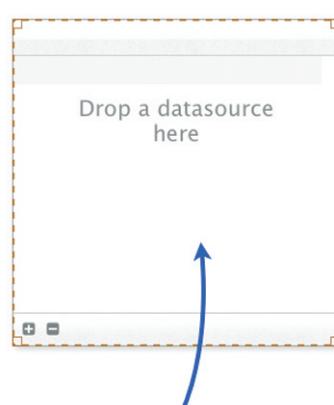
Fig.5



Fig.7



Fig.6



Signed in as John Doe

Logout

label done

Se marier

Faire du sport

Terminer ex. Waka...

Tel. Alan

Fixer mes bugs Top10

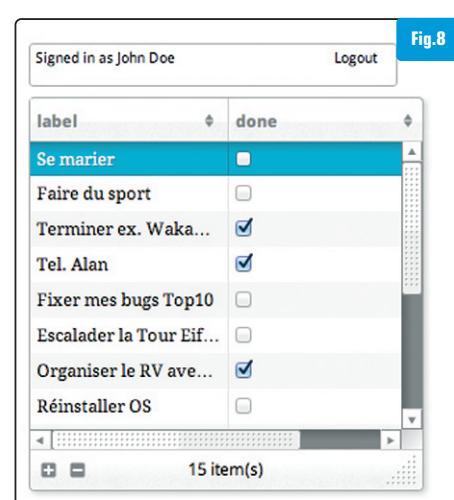
Escalader la Tour Eif...

Organiser le RV ave...

Réinstaller OS

15 item(s)

Fig.8



priété n'est pas encore totalement harmonisée. On peut ajuster le gradient, le découper en sous-gradients, modifier son orientation ou l'inverser.

> Titre et Login

Occupons-nous maintenant du reste de la page en commençant par ajouter un titre. Nous déplaçons un peu le widget *Login Dialog* et glissons un widget *Text* (ne pas confondre avec *Text Input* car nous voulons ici un texte statique) en haut à gauche de la page. Le *GUI Designer* active l'onglet « *Properties* », et nous pouvons entrer le titre dans la rubrique « *Text* ». Par exemple « *Liste de choses faites ou à faire* ». Rendez-vous à présent dans l'onglet « *Styles* » pour changer la taille de la police, sa couleur (blanche, pour que le titre soit bien visible sur le fond bleu).

Dans la partie « *Size & Position* », nous demandons que les propriétés *top* et *left* soient réglées à 20 pixels. Nous pourrions également lui ajouter des effets d'ombres et une bordure, mais après quelques essais aux résultats esthétiques dignes d'un programmeur, et non d'un graphiste, nous avons considéré que du blanc de base sur le fond bleu, c'était bien suffisant.

Le widget de *login* subit également une cure d'esthétisme. Nous le déplaçons sous le titre. Pour l'aligner à gauche, nous maintenons *ctrl* (Windows)/*cmd* (MAC), une grille magnétique nous aide à le positionner. Sa couleur de fond est ensuite réglée sur « *transparent* », sa taille de police est augmentée et mise en blanc [Fig.10].

> Grid

La grille, enfin, est également déplacée, alignée à gauche avec les deux autres widgets, et mise en forme. Le widget *Grid* est un wid-

get complexe, de très haut niveau, contenant tout un ensemble de sous-widgets et de DIV, chaque sous-partie étant elle-même paramétrable individuellement. C'est pourquoi, quand une grille est sélectionnée, l'onglet *style* contient, tout en haut, un popup menu permettant d'indiquer la partie sur laquelle nous travaillons : la grille dans son ensemble, l'entête, le pied, une colonne, une ligne, une ligne paire, une ligne impaire... [Fig.11].

Donnons libre cours à notre imagination débordante quand il s'agit de peindre : les lignes paires et impaires ont un fond bleu un peu différent, l'entête est également passé en bleu, nous avons francisé les libellés, etc. Enfin, nous allons ajouter un peu de code pour gérer l'affichage de la liste en fonction de l'utilisateur connecté. Une fois le *Login Dialog* sélectionné, se positionner sur l'onglet « *Events* », et cliquer sur « *on Login* ». Le *GUI Designer* crée alors un fichier « *index.js* », l'inclut dans le loader pour qu'il soit chargé dans le navigateur, et l'affiche. Compte tenu du code écrit dans l'événement « *on RestrictingQuery* » sur le serveur, nous pouvons nous contenter de faire extrêmement simple : nous demandons à lister *toutes* les tâches. Le serveur filtrera la liste pour nous. Le code est le suivant :

```
WAF.sources.todo.all();
```

Où *WAF* est le *namespace* du Framework, *sources* un objet contenant toutes les datasources de la page (dans cet exemple, il n'y en a qu'une), *todo* est cette datasource et *all()* une API des datasources qui fait ce que son nom indique.

De retour dans la page *html*, nous cliquons maintenant sur le bouton de l'événement « *on Logout* », et insérons ce code :

```
WAF.sources.todo.noEntities();
```

Là encore, le nom de l'API parle de lui-même.

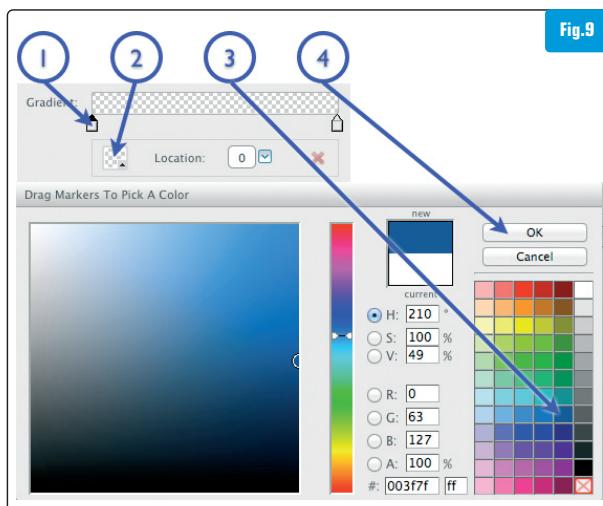


Fig.9



Fig.10

Une note sur l'intérêt du *model-driven* de Wakanda. Puisque les options de sécurité sont exécutées sur le serveur, utiliser *all()* pour l'événement *onLogout* aura le même résultat que *noEntities()*. Un utilisateur tentant de voir la liste des tâches de tout le monde, qui aurait l'idée d'utiliser le debugger du navigateur pour exécuter la requête *all()* en serait pour ses frais.

Après avoir écrit ce code, nous rechargeons la page sur le navigateur. Vous pouvez maintenant essayer de vous identifier en utilisant différents utilisateurs. Dans l'exemple fourni, tous ont le mot de passe « *pass* ». Essayez de vous connecter en tant que « *john* », « *alan* » ou « *admin* ». « *admin* » est le seul utilisateur qui pourra voir la liste des tâches de tous les utilisateurs.

Ici, nous avons laissé la grille en mode « *read-write* ». Vous pouvez donc créer une nouvelle tâche en cliquant le bouton « *+* » en bas à gauche, et en supprimer en sélectionnant une ligne et en cliquant le bouton « *-* ». L'application finale disponible au téléchargement est plus aboutie, utilise plus de widgets (des onglets, une matrice, ...) et affiche une liste de tâches en cours, et une autre de tâches à faire.

> On se téléphone ?

Maintenant que nous pouvons gérer nos tâches depuis une application web desktop, il est temps de créer une interface pour un mobile. Pour l'iPhone dans le cas présent. Fermons les onglets encore ouverts, sélectionnons le dossier « *WebFolder* », dans la liste de gauche, puis depuis la barre d'outils, cliquons le bouton « *New page* », option « *for iPhone* », nommons-la aussi « *index.html* ». Le *GUI Designer* crée, si ce n'était déjà fait, un dossier « *Smartphone* » dans le *WebFolder*, y ajoute la page, et l'ouvre en affichant un iPhone en fond de page [Fig.12]. Tout comme précédemment, il s'agit de glisser-déposer des widgets, les lier à des datasources, les formater et écrire le code



gérant le *login/logout*. Pour changer un peu, nous allons afficher une grille, des détails et des boutons (création, validation, suppression), sans oublier le widget de *login*.

Avant toute chose, dans les propriétés du document, nous déployons « iOS Meta Tags » et cochons « Web App Capable ». Cela permet de créer une icône dans l'écran d'accueil et de ne pas afficher la barre de navigation de Safari, ce qui nous fera gagner beaucoup de place.

Nous créons notre datasource comme nous l'avons fait précédemment. Puis nous disposons nos widgets sur le plan de travail. Nous

allons procéder en trois étapes : brouillon, styles et positions, finalisation.

> PREMIÈRE ÉTAPE : brouillon.

Nous glissons-déposons un widget *Text*, en haut de l'espace. Il s'agit du titre, nous lui donnons donc le libellé « Choses faites et à faire », et le laissons en haut, sans modifier son style pour le moment. Puis nous déposons une *Grid*, la positionnons en dessous du titre et réduisons un peu sa hauteur. Nous déposons maintenant un widget *Container*, sous la *Grid*. Là encore, on réduit un peu sa hauteur. Enfin, nous installons le widget de *login*, tout en bas, sous le container (ne le mettez pas *dans* le *Container*). Le brouillon, le premier jet de notre interface est fait [Fig.13]. Notez que le *Container*, lorsqu'il n'est pas sélectionné, devient invisible. Vous pouvez l'afficher en cliquant le bouton « Show shapes » de la barre d'outils du *GUI Designer*.

> SECONDE ÉTAPE : styles et positions.

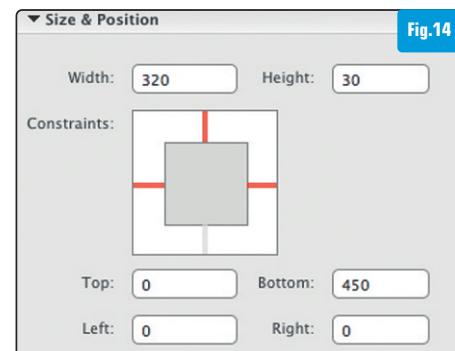
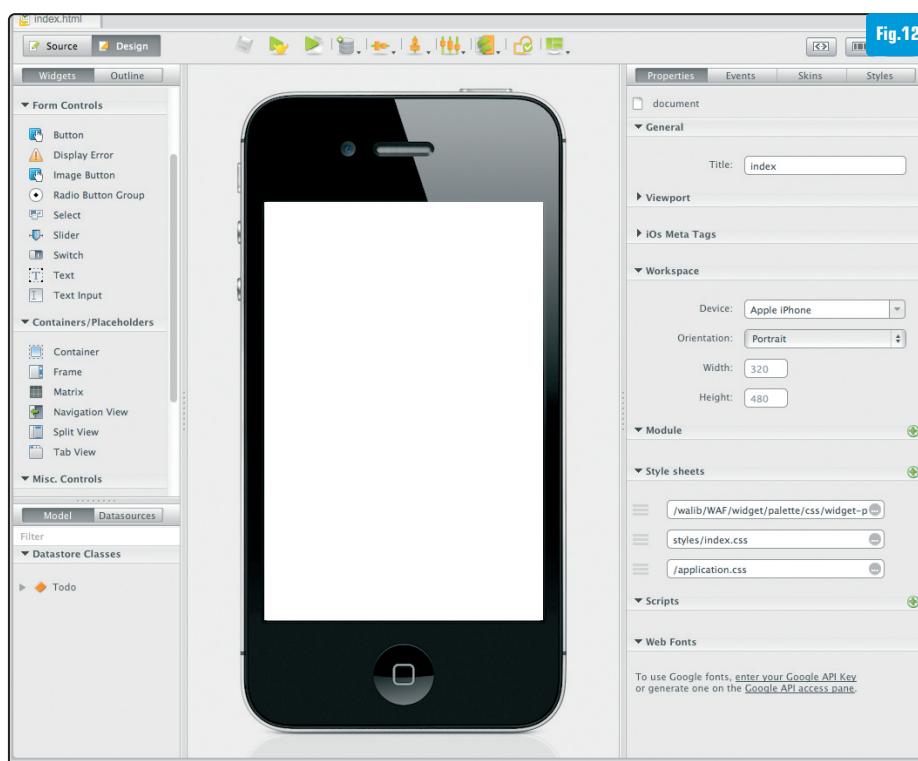
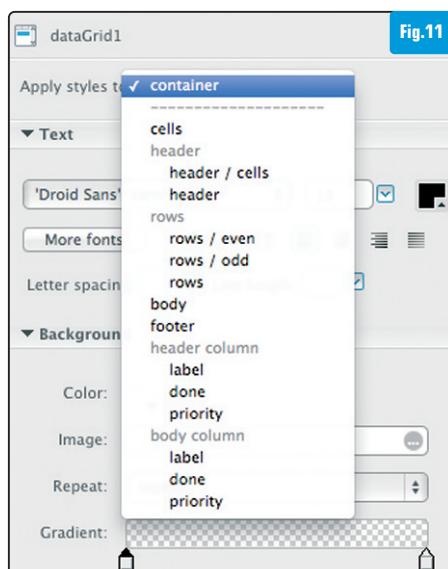
Il s'agit ici de coloriser les widgets et de modifier leur position. Pour gérer automatiquement la tenue du téléphone en mode paysage, nous allons utiliser les propriétés d'ancrage. Commençons par le titre. Après l'avoir sélectionné, nous lui donnons une taille de police de 16, alignement centré, hauteur de ligne 25. Sa taille est réglée à

une hauteur de 30 et des positions *top*, *left* et *right* à 0. Pour qu'il s'élargisse automatiquement lors du passage en mode paysage, nous lui ajoutons une contrainte à droite, en activant le magnétisme en cliquant sur la ligne horizontale droite de la zone « Constraints » [Fig.14].

Nous lui donnons une couleur de fond saisie en hexadécimal dans la zone inférieure du *Color Picker*, 6d85a3, opacité ff. Enfin, la couleur du texte est mise en blanc.

Nous pouvons maintenant formater la *Grid*. Il s'agit ici juste de modifier son positionnement, et de lui associer les attributs à afficher. Nous positionnons la grille de la façon suivante : largeur 320, hauteur 273, top 30, *left* et *right* 0. Inutile de modifier la propriété *bottom*. Pour la *Grid*, nous activons les quatre lignes de contraintes. Nous déployons la datasource *todo*, sélectionnons *label* et *creationDate* et les glissons-déposons sur la *Grid*. Dans l'onglet « Properties », zone « Columns », nous francisons les libellés et modifions la largeur de chaque colonne (200 pour le libellé, 100 pour la date par exemple).

Passons à présent au widget *Container*. Nous lui donnons la couleur de fond usuelle (6d85a3, ff) et modifions ses propriétés de taille et de positionnement : largeur 320,





hauteur 154, top 304, *left* et *right* à 0. Nous voulons ancrer le container en bas (il sera « repoussé » ou « attiré » par la *Grid*),

il faut donc dé-sélectionner l'ancrage supérieur, et activer les ancrages droit et bas. Enfin, nous lui ajoutons des propriétés d'ombrage pour lui donner un aspect encore plus moderne. En l'occurrence, nous travaillons sur les propriétés « Inner Shadow » : couleur blanche, Distance 0, Blur 5 et Size 2. Nous terminons à présent avec le widget *Login Dialog* : couleur de fond usuelle (6d85a3, ff), couleur de texte blanche et posi-

tionnement : hauteur 24, top 456, bottom/left/right à 0. Comme le *Container*, nous voulons que ce widget soit ancré en bas (et à droite, comme pour tous les autres).

> DERNIÈRE ÉTAPE : finalisation.

Ici, nous allons remplir le Container qui est désespérément vide. En vrac, nous déposons dans le container : un *Text Input* en haut, un *Switch* au milieu, et trois *Button* (en bas, et à gauche, au milieu, à droite). Nous déployons la datasource *todo* et glissons-déposons l'attribut *label* sur le widget texte. Puis nous glissons-déposons l'attribut *done* sur le widget *Switch*. Enfin, nous déposons

Wakanda : une plateforme très ouverte

Quand on arrive sur le site Web de Wakanda, on perçoit très rapidement le côté complet de la plateforme. Si on peut imaginer sa puissance, certains se poseront des questions légitimes sur la modularité ou l'interopérabilité : l'architecture de Wakanda a été conçue dans un objectif d'ouverture maximale et de support des standards du Web.

> Un Studio de développement modulaire

• Un support multi-OS

Le studio Wakanda est un environnement « Desktop » tout à fait adapté pour travailler en mode déconnecté. Il tourne nativement (pas de Java) sur **Windows** et **MacOS X**. Il n'y a pas aujourd'hui de version Desktop Linux, mais d'autres solutions sont présentées plus bas pour cet environnement.

• Un éditeur de code sympathique

Le moteur de cet éditeur a en premier lieu été optimisé pour gérer les concepts du langage JavaScript qui fait tourner Wakanda. Il connaît bien sûr les API des objets natifs, mais surtout, comprend les contextes par fonction, les « fermetures » (*closures*), et les héritages par prototype à plusieurs niveaux.

Il supporte d'autres langages Web incontournables et souvent indispensables, comme le HTML, le CSS, ou le XML (et même le PHP), et également les fichiers JSON.

• Des applications Web intégrées

L'environnement du studio, l'explorateur de solutions et de projets, ainsi que l'éditeur de code sont en code natif mais la majorité des outils mis à votre disposition ont été **conçus à partir de technologies Web HTML5 / CSS3 / JavaScript** tournant dans un environnement

Webkit. C'est le cas : du *Model Designer*, du *GUI Designer*, du *Debugger*, du gestionnaire d'utilisateurs et groupes, de la plupart des éditeurs de settings ou de préférences. Il est donc tout à fait envisageable de les rendre utilisables depuis d'autres plateformes telles qu'Eclipse, NetBeans, Sencha Architect, ou Cloud9.

Partagez de votre look à votre squelette

Chaque projet Wakanda peut être enregistré en tant que *template*. Il devient ainsi aisé de réaliser des modèles comprenant des DataClasses, et, comme nous le verrons plus tard, des *Web Components*, modules, ou services, utilisables dans plusieurs contextes.

Le *GUI Designer* permet d'exporter des mises en forme de widgets en tant que *skins CSS* par défaut réutilisables à tout endroit du projet. Ils peuvent être intégrés dans un *template* de projet ou importés dans un projet existant.

• Des extensions réalisées simplement

Au même titre que le studio intègre des applications Web, il permet d'y ajouter **des extensions en JavaScript**, avec ou sans interface graphique HTML/CSS. Wakanda en fournit par défaut quelques-unes dont une extension **JSLint**. Un outil de gestion de versions basé sur **Git** est en développement, et de nombreuses autres extensions ont été envisagées, par la communauté comme par l'équipe

Wakanda, pour intégrer des projets puissants et complémentaires tels qu'Apache Cordova (**PhoneGap**), Pixastic, ou Selenium.

• Vers le Cloud

Ce choix des technologies Web pour le studio n'est pas innocent. Non seulement il permet de les mettre à l'épreuve dans le cadre d'applications métiers professionnelles, mais il autorise également une ouverture significative vers le monde des **Web IDE** tels que Cloud9, Orion, ou Akshell. Une future version pourra ainsi être utilisable depuis n'importe où, y compris depuis des environnements Linux.

> WakandaDB pour tout le monde

• NoSQL pour tous

Le serveur WakandaDB intègre en son cœur, au même titre que CouchDB, un serveur HTTP et un moteur d'exécution JavaScript. Son **API HTTP REST**, proche du standard OData, est facilement accessible depuis n'importe quelle technologie tierce, telle que .NET, Java, PHP, Ruby, Python, node.js, Titanium, mais aussi directement depuis des Frameworks Ajax populaires tels que Sencha/ExtJS, backbone.js, ou Kendo UI. Une base de données Wakanda n'a besoin en théorie, pour être conçue, que d'un simple éditeur de texte, avec idéalement un support de XML et JavaScript. Les premiers



toute la datasource *todo* sur chaque bouton. Nous supprimons les libellés des widgets *Text Input* et *Switch* (onglet « Properties », vider le texte associé à « Label »), et les positionnons correctement. Pour le *Switch*, nous modifions les libellés « *ON/OFF* », pour « *Fait* » et « *À faire* » et le retaillons un peu pour bien voir ces libellés. Le *GUI designer* contient des outils d'alignement et de répartition : nous n'hésitons surtout pas à les utiliser pour centrer les widgets dans leur container, et pour aligner et répartir les boutons. Les boutons, justement, parlons-en. Nous avons associé chaque bouton à la datasource *todo*. Cela nous permet de bénéficier d'actions automatiques, prises en charge par le WAF. Sélectionnons le premier

bouton, à gauche et donnons-lui l'action « *Create* », et le libellé « *Nouveau* ». Le second bouton se voit attribuer l'action « *Remove* », libellé « *Supprimer* ». Enfin, le dernier bouton a l'action « *Save* » et le libellé « *Enregistrer* ». Notre interface est terminée, elle est magnifique ! [Fig.15].

Il reste une dernière chose à faire : ajouter le code listant les tâches lors du *login/logout*, comme cela a été fait précédemment. Vous pouvez maintenant lancer le serveur, et soit utiliser le Simulateur iOS, soit connecter votre téléphone à votre machine si vous êtes dans le même réseau Wi-Fi. N'oubliez alors pas de modifier les réglages du projet pour préciser sur quelle adresse et quel port votre serveur doit écouter. Enfin,

dernière étape : une fois la page chargée dans Safari sur l'iPhone, demandez de suite à créer un bouton dans l'écran d'accueil, puis utilisez ce bouton, modifiez des tâches, changez d'utilisateur... [Fig.16].

> Conclusion

Wakanda contient tous les outils, runtime et développement, pour créer des applications métiers déployées sur le web, en tant qu'application desktop ou mobile. Explorez les widgets du *GUI Designer* et regardez l'application fournie en exemple, qui contient également une interface pour l'iPad.

Thibaud Arguillère

Senior Software Evangelist

<http://github.com/thibarg/wakanda-todo-programmez>

projets Wakanda, alors que le studio n'existe pas encore, étaient réalisés en utilisant TextMate, Notepad++, Eclipse, ou d'autres.

• Remote Debugging protocol

Le code JavaScript s'exécutant sur le serveur peut être débogué à distance via le protocole **CrossFire**, défini par le groupe de travail de Firebug, dans une version étendue pour intégrer une gestion de droits d'accès. Il est aujourd'hui utilisé par le débogueur intégré de Wakanda. Le support de « **Webkit Debugger protocol 1.0** » conçu par l'équipe de « **Chrome Developper Tools** » a été récemment ajouté, permettant ainsi d'interagir avec Wakanda depuis le Web Inspector de Chrome (et potentiellement plus tard, avec celui intégré dans Safari, Apache Weinre, ou Adobe Shadow).

• De CommonJS au W3C

Au delà du support de **ECMAScript 5** fourni par le moteur Webkit JavaScriptCore, la réflexion apportée aux API serveur a été directement orientée vers la question de l'interopérabilité. Wakanda permet donc, en plus des DataClasses du modèle, de rajouter des librairies sous forme de modules **CommonJS** facilement partageables, ou d'utiliser l'API console créée par Firebug. Plus particulièrement, on doit se rappeler que la spécification **W3C HTML5** fut originellement proposée par le groupe de travail **WHATWG**, et co-intitulée « **Web Application 1.0** ». Pour chaque API à fournir, 4D a donc choisi d'implémenter celles du HTML5 en priorité par rapport à d'autres qui seraient propres au serveur. Wakanda Server supporte par exemple nativement les API standard W3C : Web Worker, Web Storage, FileSystem, Timers, XMLHttpRequest, Blob... Certaines d'entre elles ont été étendues, ou réutilisées dans d'autres contextes

comme par exemple SystemWorker qui permet de créer des processus depuis une commande système.

Cette approche permet d'avoir :

- un code serveur facilement compréhensible par un développeur front-end,
- une logique métier implémentée de manière à tourner aussi bien sur le serveur, que sur le client en mode *offline*.
- des librairies/modules fonctionnant selon les besoins, aussi bien de manière synchrone qu'asynchrone.

Note : Compte tenu du nombre de modules existants pour node.js, Wakanda supporte également un certain nombre des spécificités de cet environnement.

• A votre service !

Wakanda définit une notion de service. Les services sont des modules CommonJS administrables par le serveur. Ils peuvent fournir une connexion avec un composant tiers, gérer des tâches récurrentes, ou mettre en place des API au niveau socket ou HTTP. C'est ainsi que le service JSON-RPC est fourni par Wakanda Server.

> Le framework, en français : « Cadre de travail »

• Vos librairies sont nos amies

Le framework Wakanda intègre par défaut des librairies éprouvées telles que **jQuery** et **Raphaël**. Il est aisément de déclarer des dépendances d'autres librairies depuis le *GUI Designer*, permettant ainsi au *loader* du framework de garantir leur disponibilité lorsque le code de l'interface s'exécute.

• Interfaces Web mobiles ou tablettes dédiées

Wakanda permet de définir des interfaces

dédiées pour des environnements *desktop*, *tablet* ou *mobile*. Tous les widgets sont utilisables en mode tactile, et des widgets spécifiques sont disponibles pour mieux répondre aux ergonomies d'environnement tactile ou à écran réduit.

• Pensez Composants Web !

Wakanda est optimisé pour des applications de type SPA (Single Page Application) en mode Web 2.0. Une page créée est considérée comme une interface depuis laquelle vous pouvez charger des composants Web.

Ces composants Web contiennent leurs propres fichiers CSS, HTML, et JavaScript, et peuvent fournir des API aux interfaces qui les utilisent. Ces composants sont donc idéaux pour partager des bouts d'interfaces interactives au sein d'un même projet ou entre plusieurs projets.

• Faites vos propres Widgets

Wakanda vous permet également de créer vos widgets d'entreprise, ou d'intégrer des widgets existants. Ceux-ci peuvent alors être branchés sur le mécanisme *Pub/Sub* des datasources de Wakanda, et être configurés graphiquement depuis le *GUI Designer* du Studio en profitant ainsi de l'éditeur de CSS3.

> Quelle Licence pour conduire Wakanda ?

Wakanda est un projet **Open source** utilisable sous licence **GNU**, au même titre que MongoDB, MySQL, ExtJS, Cloud9 ou Aptana Studio. Les extensions du studio sont souvent fournies sous licence **MIT**. Une licence **commerciale** est également disponible pour un prix tout à fait modeste.

Alexandre Morgaut

Architecte Web et Wakanda Community manager



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

2^e partie : un peu de pratique

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. Passons aux travaux pratiques.

> Cas pratique n°1 : piloter sa chambre en X10

Pour démarrer, il nous faut un contrôleur X10 pour émettre et recevoir des messages X10 sur le réseau électrique et/ou radio. Pour piloter notre réseau depuis un PC, je vous conseille le contrôleur CM15Pro chez Marmitek ou XDom qui propose, via une interface

USB, la possibilité de communiquer en radio ou par courant porteur (disponible pour environ 80 euros). Il existe aussi d'autres contrôleurs comme le CM11 mais il ne gère pas l'interface radio [Fig.1]. Concernant les récepteurs, il existe une multitude d'équipements possibles : modules de prise de courant, interrupteurs, micromodules à encastrer derrière l'interrupteur existant, modules Rail DIN à installer dans le tableau électrique, etc. Pour notre chambre, nous allons simplement utiliser les modules en prise de courant. Il existe le module AM12 (pour Appliance) qui gère le On/Off, et le module LM12 (pour Lampe) qui supporte en plus la variation de l'intensité de la lampe. Ces deux produits se commercialisent aux environs de 30 euros par module [Fig.2].

Fig. 1



Contrôleur X10 (CM15)

Fig. 2



Module Lampe X10 (LM12)

Pour développer avec le CM15, rien de plus simple, il suffit d'écrire un programme en .NET en référençant le composant COM d'ActiveHome. Vous le trouverez après avoir installé le SDK ActiveHome dans le dossier « C:\Program Files (x86)\AHSDK\bin », le fichier se nomme « ahscript.dll ».

Il suffit ensuite de créer une instance de classe « ActiveHome ». Vous disposerez sur cette instance de deux membres fondamentaux :

- Une méthode « SendAction » pour envoyer un message X10 sur le réseau
- Un événement « RecvAction » déclenché quand le contrôleur reçoit un message

Ces deux membres comportent dans leurs arguments le type de l'action (SendPLC, QueryPLC, RecvPLC, SendRF ou RecvRF pour définir l'émission, l'interrogation ou la réception, par courant porteur ou radio fréquence), l'adresse du récepteur (A1 à P16), le type de l'ordre (On, Off, Dim, Bright, AllLightOn, AllLightOff) et des paramètres optionnels (comme la valeur du « Dim » ou « Bright » en pourcentage lorsqu'on fait varier l'intensité d'une lampe).

Pour allumer la lampe A1 :

```
ActiveHome activeHome = new ActiveHome();
activeHome.SendAction("SendPLC", "a1 on", "", "");
```

Pour la réception, il suffit de s'abonner à l'événement « RecvAction ». Par exemple :

```
activeHome.RecvAction += new _DIActiveHomeEvents_RecvAction
EventHandler(activeHome_RecvAction);

private void activeHome_RecvAction(object bszAction, object
bszParm1, object bszParm2, object bszParm3, object bszParm4,
object bszParm5, object bszReserved)
{
    Debug.WriteLine(string.Format("Message recu de l'équipement
'{0}' : Action = {1} | Command = {2}", (string)bszParm1, (string)
bszAction, (string)bszParm2);
}
```

Chez xBrainSoft, nous avons développé en 2008 une API complète pour l'environnement .NET par la définition des objets de base du protocole (décrivant ce qu'est un message, un contrôleur et un device X10 de manière abstraite) et des différents produits X10 : le CM15, les modules AM10, AM12, SW10 mais aussi d'autres équipements comme le détecteur de mouvement MS13, etc. Cette librairie supporte aussi de manière native un mode « proxy » de manière à pouvoir utiliser un contrôleur à distance par l'usage de WCF.

```
// Utilisation du contrôleur CM15 connecté sur une machine distante
X10ControllerProxy cm15 = new X10ControllerProxy("net.tcp://
192.168.0.31:889/CM15.svc");
cm15.Open();
// Ajout des devices
cm15.AddDevice(new ApplianceAM12(X10Address.A1));
cm15.AddDevice(new LampLM12(X10Address.A2));
cm15.AddDevice(new DetectorMS13(X10Address.A3));
```

```

cm15.AddDevice(new LampLM12(X10Address.A11));
// Abonnement au changement d'état
cm15.GetDeviceByAddress<ApplianceAM12>(X10Address.A1).StateUpdated += (object sender, EventArgs e)
    => Console.WriteLine(`ApplianceAM12 Update State = « + (sender as ApplianceAM12).State);
cm15.GetDeviceByAddress<LampLM12>(X10Address.A2).StateUpdated += (object sender, EventArgs e)
    => Console.WriteLine(`LampLM12 Update State = « + (sender as LampLM12).State);
cm15.GetDeviceByAddress<LampLM12>(X10Address.A3).IntensityUpdated += (object sender, EventArgs e)
    => Console.WriteLine(`LampLM12 Update Intensity = « + (sender as LampLM12).Intensity);
cm15.GetDeviceByAddress<DetectorMS13>(X10Address.A1).MotionDetected += (object sender, EventArgs e)
    => Console.WriteLine(`DetectorMS13 Motion Detected @ « + (sender as DetectorMS13).LastMotionDetected.ToString());
// Allumage de A1 et A2
cm15.GetDeviceByAddress<ApplianceAM12>(X10Address.A1).TurnON();
cm15.GetDeviceByAddress<LampLM12>(X10Address.A2).TurnON();
// Variation de l'intensité de la lampe A2
cm15.GetDeviceByAddress<LampLM12>(X10Address.A2).SetIntensity(50);

```

Ainsi, le développement .NET d'une application de contrôle d'un réseau domotique X10 est facile à mettre en œuvre et peu coûteux ! L'idéal pour démarrer ou équiper quelques pièces. Mais comme indiqué plus haut, ses lacunes sur les points de sécurité et surtout de fiabilité (mais aussi de discréption, ne passant pas le label « WAF ») m'ont personnellement freiné quant au déploiement à l'ensemble de l'habitat.

> Cas pratique n°2 : piloter sa maison en Z-Wave

La solution que j'ai retenue donc pour mon habitat est le Z-Wave. Sa topologie en réseau de maillage permet de d'obtenir une bonne fiabilité et rapidité dans la transmission des ordres (quasiment instantanée) et sa capacité bidirectionnelle permet d'assurer la bonne exécution des ordres donnés !

Pour commencer, il nous faut un contrôleur « maître » pour gérer l'ensemble de notre réseau Z-Wave. Nous pouvons utiliser des télécommandes ou interrupteurs comme contrôleurs « maîtres », des interfaces PC avec l'usage d'une partie logicielle, ou bien opter pour une box domotique embarquant un mini pc généralement à base de Linux y associant la partie hardware et software dans une solution packagée. Dans notre cas, les télécommandes et interrupteurs seront considérés comme des contrôleurs « secondaires » et nous choisirons une solution par box ou PC comme contrôleur « maître ».

Utilisation de contrôleur USB Z-Stick S2 comme contrôleur « maître »

Fig.3



Contrôleur Z-Wave (Z-Stick S2)

Une solution consiste à utiliser une interface PC, généralement en USB. C'est le cas notamment du contrôleur Z-Stick S2 d'Aeon Labs commercialisé autour de 65 euros. Ressemblant à une simple clé USB, il comporte une batterie intégrée pour permettre d'inclure ou d'exclure vos périphériques Z-Wave sans avoir à déplacer votre ordina-

teur ! Pour la partie software, vous pouvez utiliser des logiciels comme HomeSeer, mControl ou encore ControlThink (tous payants). Certains d'entre eux mettent à disposition des API pour contrôler votre réseau Z-Wave depuis vos applications [Fig.3].

Autres possibilités, l'achat du SDK directement chez Sigma Design ou développer soi-même son API mais la tâche est beaucoup plus ardue ! La clé Aeon Labs, simple interface COM, est pilotable par un SerialPort en C#.

Dans la pratique, pour contrôler la marche ou l'arrêt d'une simple prise Z-Wave (que nous avons préalablement ajoutée à notre réseau Z-Wave), il faut ouvrir le port COM de notre contrôleur Aeon (dans notre exemple connecté sur le port COM7) pour écrire le message suivant :

```

/*
 0x01, 0x09, 0x00, 0x13, nodeId, 0x03, 0x20, 0x01, state, 0x05,
 checksum
*/

```

Le « nodeId » sera l'ID de notre prise Z-Wave à allumer et « state » l'action à exécuter qui sera 0xFF pour allumer la prise, ou 0x00 pour l'éteindre. Enfin, un checksum sera calculé en faisant un XOR sur les octets du message à l'exception du premier.

```

SerialPort sp = new SerialPort()
{
    PortName = «COM7»,
    BaudRate = 115200,
    Parity = Parity.None,
    DataBits = 8,
    StopBits = StopBits.One,
    Handshake = Handshake.None,
    DtrEnable = true,
    RtsEnable = true,
    NewLine = System.Environment.NewLine
};
sp.Open();
byte nodeId = 0x06; // device ID 6
byte state = 0xFF; // On
byte[] message = new byte[] { 0x01, 0x09, 0x00, 0x13, nodeId, 0x03,
0x20, 0x01, state, 0x05, 0x00 };
message[message.Length - 1] = GenerateChecksum(message);
sp.Write(message, 0, message.Length);
sp.Close();

```

Chaque envoi de message avec le contrôleur doit être associé à un accusé de réception par l'envoi d'un « ACK » dont le code est « 0x06 ». Ainsi dans notre cas, après l'envoi de notre message pour allumer le nœud n°6, le contrôleur écrira sur le « SerialPort » du COM7 l'octet « 0x06 ». En plus de l'accusé de réception entre le client C# et le contrôleur, le nœud accusera lui aussi réception de l'ordre émis. Votre programme C# recevra donc un 2e message du contrôleur pour indiquer le résultat de l'exécution du nœud cible qu'il faudra à notre tour « accuser » en émettant un ACK sur le port série ! Il faudra donc créer un thread séparé pour lire les messages arrivant sur les contrôleurs USB et gérer nativement les « ACK » sur le port série. Il faudra également implémenter un à un les différents formats de messages propres à chaque périphérique, ce qui peut

représenter un très long travail. De plus, nous devrons ajouter le support de la découverte des nœuds sur le réseau, la gestion des messages de configuration pour chaque produit Z-Wave, ainsi que toutes les fonctions d'un contrôleur (gestion du pooling des devices, du routing, des copies du réseau entre les contrôleurs, des mise à jours des SUC/SIS, etc.). Les spécifications du protocole Z-Wave se trouvent sur Internet, bien que le document soit classé « confidentiel » et il est aussi possible d'utiliser le site « ZWave Device Library » pour retrouver tous les détails techniques des différents périphériques Z-Wave du marché.

Il existe aussi des projets open-source d'API Z-Wave, notamment le projet Open-ZWave disponible sur Google Code, écrit en C++ pour plateformes Windows, Mac et Linux, et dispose en plus d'un wrapper .NET en C++/CLI pour une utilisation dans l'environnement .NET.

Utilisation de la Vera Lite comme contrôleur « maître »

Fig.4



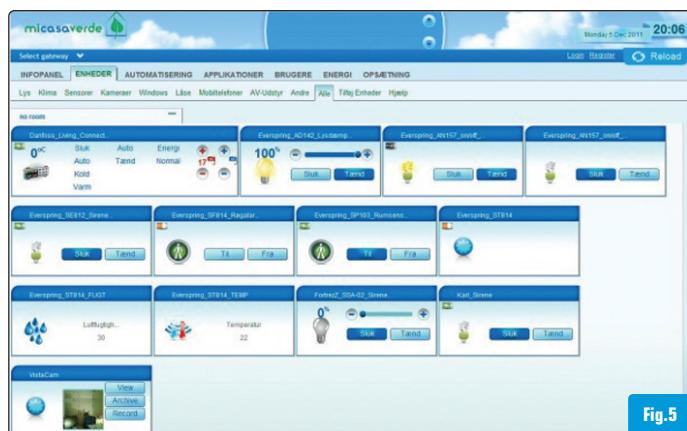
Contrôleur Vera Lite de MiCasaVerde

Pour simplifier son utilisation, nous pouvons nous tourner vers une « box domotique » qui intègre un contrôleur Z-Wave et une interface Web pour piloter et configurer son réseau. Parmi les différents produits sur le marché (eeDomus, les Vera, Home Center, ZiBase, etc.), la Vera Lite de MiCasa-

Verde offre un bon contrôleur pour 175 euros environ. Cette petite boîte (11cm x 9,5cm pour 4,4cm d'épaisseur) dispose d'un connecteur RJ-45, d'une alimentation et d'un port USB. De plus, elle intègre un logement de 4 piles AA pour faciliter les tâches d'inclusion et d'exclusion des périphériques Z-Wave et consomme très peu d'électricité (6W) [Fig.4].

Accessible depuis un navigateur Web, l'interface UI5 permettra de configurer son réseau, de contrôler ses périphériques et de créer des scénarios. Elle intègre aussi une « marketplace » pour ajouter de nouvelles fonctionnalités et propose un langage de script basé sur le Lua [Fig.5].

La Vera Lite expose un service web JSON pour piloter son réseau Z-Wave depuis des applications tierces. Par un simple appel HTTP, nous pouvons récupérer dans un objet JSON, la liste des différents périphériques du réseau ainsi que leurs états. Pour lancer un ordre, il suffira d'appeler une URL en précisant dans l'URL l'adresse et l'action à réaliser. Bien plus simple qu'écrire sur un port COM car ici,



Interface Web UI5 de la Vera Lite



Fig.6
Client sur Windows Phone 7

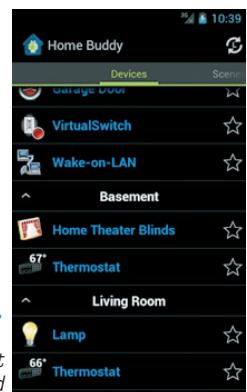
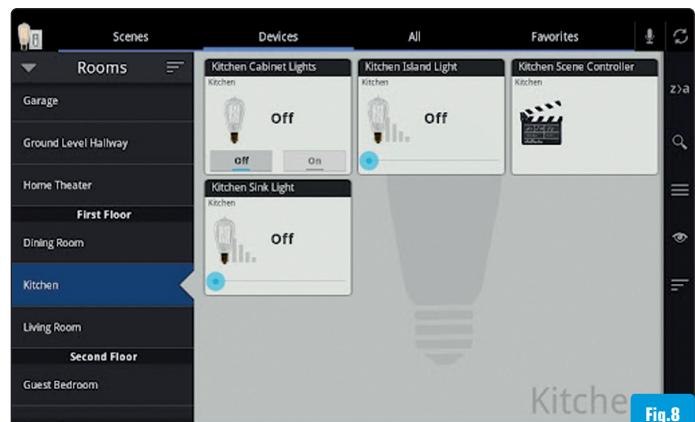


Fig.7
Client Android



Client pour tablette Android

toute la logique et l'implémentation du protocole du Z-Wave et des périphériques, sont encapsulées par la Vera.

Grâce à cela, il est possible de trouver sur les différents marchés (App Store, Google Play ou Marketplace WP7), des applications de pilotage de la Vera permettant le contrôle de son habitat depuis son mobile ou sa tablette, localement ou à distance [Fig.6, 7 et 8].

Contrôler des prises électriques, l'éclairage et le chauffage

Fig.9



Module AN158-6 avec consomé-mètre

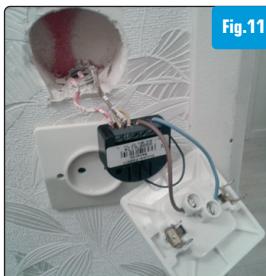
Pour les prises électriques, je vous conseille le modèle AN157-6 chez Everspring (environ 35 euros) pour le contrôle de la marche/arrêt. Il existe aussi le modèle AN142-6 (45 euros) pour la fonction variateur. J'utilise aussi le modèle AN158-6 (45 euros) qui remonte au contrôleur la consommation électrique de la prise [Fig.9].

Pour l'éclairage, le chauffage ou les volets, je vous recommande les micromodules chez Fibaro à encastrer. Il existe quatre modèles : le FGS-211 (relai 3kW), le FGS-221 (double relais 2x1,5kW), le FGR-221 (pour volet électrique) et le FGD-211 (variateur). Comptez environ 60 euros par module [Fig.10].

Pour l'éclairage, j'installe derrière mes interrupteurs des modules FGD-211 qui fonctionnent sans le neutre (en général, dans un habi-



Micromodule Z-Wave chez Fibaro



Installation d'un FGD-211 pour l'éclairage

Fig.11

tat, un interrupteur va-et-vient ne fait que couper la phase, il n'y a donc pas de neutre qui arrive jusqu'à [Fig.11]. Pour le chauffage électrique, j'utilise un double relais (le FGS-221) couplé avec deux diodes de redressement pour contrôler le radiateur par son fil pilote. Le fil pilote permet de commander le mode du radiateur (Arrêt, Hors gel, Eco et Confort) par un courant alternatif. Pour créer ces ordres nous utiliserons des diodes de redressement 1N4007 [Fig.12 et 13].



Diode de redressement 1N4007

Fig.12

Le module Fibaro et le radiateur sont alimentés par le 230V. La 1^{re} sortie du module Fibaro est connectée au fil pilote du radiateur par l'intermédiaire d'une diode 1N4007, et la 2^e sortie est elle aussi connectée au fil pilote mais dans le sens inverse [Fig.14, 15 et 16].

Ainsi, si les deux relais sont sur Off, le radiateur est en mode Confort, si les deux relais sont sur On, c'est le mode Eco. Pour le mode arrêt il faut le relais 1 sur On et le relais 2 sur Off. Enfin, le mode Hors gel est déclenché avec le relais 1 sur Off et le relais 2 sur On. Pour faciliter l'usage du fil pilote, vous trouverez sur le market de la Vera le module « Pilot wire controller » [Fig.17].

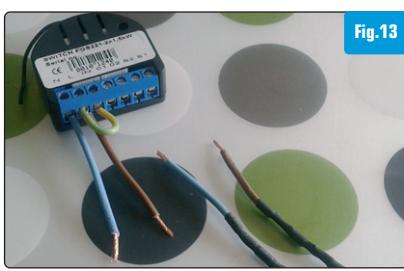


Fig.13

Micromodule FGS221 et ses diodes de redressement

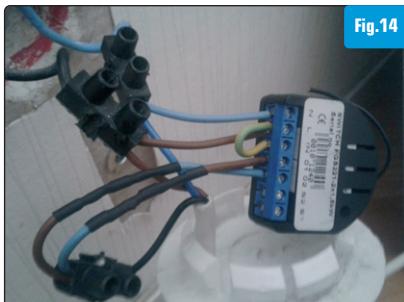


Fig.14

Installation du FGS221 derrière le chauffage



Fig.15

Les micromodules se placent facilement dans les boîtes d'encastrement

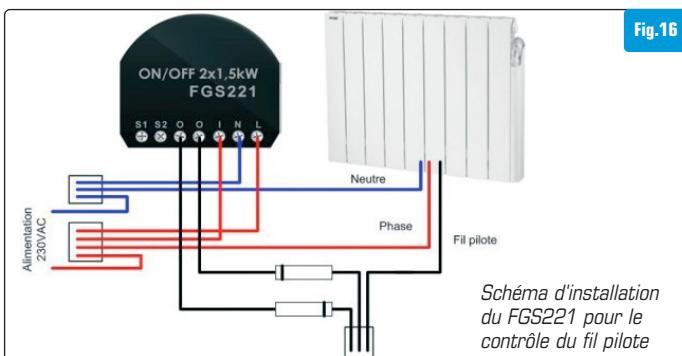


Fig.17

Plugin "Pilot Wire Controller"

Fig.18



Capteur ST814 chez Everspring



Fig.19

Pince ampère métrique connectée sur la phase



Fig.20

Module HEM en dehors du tableau pour communiquer au contrôleur la consommation électrique des pinces.



Fig.21

Détecteur d'ouverture HSM02



Fig.22

Télécommande Minimote chez Aeon Labs

radio avec affichage intégré fonctionnant par piles. Pour les conditions extérieures il est possible d'installer depuis le market Google Weather [Fig.18].

Pour la consommation électrique de l'ensemble du logement, le Home Energy Meter chez Aeon Labs (100 euros) est un module qui utilise des pinces ampère métriques à installer dans le tableau électrique sur la phase, afin de déduire la consommation instantanée et totale (avec une marge d'erreur d'environ 10%) [Fig.19 et 20].

Citons aussi les modules de détection d'ouverture de porte (le HSM02 chez Everspring pour 50 euros) ou de mouvement dont certains incluent un capteur de luminosité (chez Express Controls) [Fig.21].

Ajouter des contrôleurs « secondaires »

En plus des interfaces informatiques (web, smartphone, tablette, ...) nous pouvons rajouter des télécommandes ou « interrupteurs contrôleurs » pour faciliter l'usage au quotidien [Fig.22]. Aeon Labs propose la Minimote (env. 70 euros), petite télécommande qui contrôle jusqu'à 8 scénarios. Et pour ne pas « perdre » le contrôle, il est possible de rajouter ou remplacer nos vieux interrupteurs par des « interrupteurs contrôleurs Z-Wave ». Le ZME 06443 (50 euros) peut s'installer sans boîte d'encastrement et fonctionne avec deux piles, ce qui laisse le choix de l'installer sans aucune contrainte. Il permet de déclencher les scénarios par simple clic ou double clic, avec bouton du haut ou du bas [Fig.23 et 24].

Le ZME 06443 (50 euros) peut s'installer sans boîte d'encastrement et fonctionne avec deux piles, ce qui laisse le choix de l'installer sans aucune contrainte. Il permet de déclencher les scénarios par simple clic ou double clic, avec bouton du haut ou du bas [Fig.23 et 24].

Automatiser et surveiller votre maison

La Vera Lite propose une interface pour créer vos scénarios. Les scénarios sont un enchaînement d'actions (ou de scripts Lua), comme par exemple ouvrir un volet, mettre le chauffage, allumer une lampe, etc. Ces scénarios pourront être déclenchés manuellement, par un événement ou



Fig.23



Fig.24

Interrupteur Z-Wave ouvert

Interrupteur Z-Wave installé

de manière planifiée. Nous voulons par exemple, couper le chauffage quand une fenêtre est ouverte, ou allumer la lumière de l'entrée lorsqu'une présence est détectée et qu'il fait noir. On peut aussi utiliser les tâches planifiées pour publier l'ensemble des variables de l'habitat sur un service comme Cosm (ex-Pachube) et ainsi suivre l'évolution des différentes informations sur l'utilisation de l'habitat. En Lua, nous pourrions planifier le script suivant toutes les X minutes pour envoyer la valeur de nos différents capteurs sur Cosm :

```

local apikey = «xxxxxxxxxxxxxxxx»
local base_url = «http://api.cosm.com/v2/feeds/»
local json_datas = {}
for i,v in pairs(datas) do
  table.insert(json_datas, '{ «id»:' .. i .. '», «current_
value»:' .. v .. '«}')
end
local json_data_str = '{ «version»:»1.0.0», «datastreams»:[ ' ..
table.concat(json_datas, ',') .. ']}'
local socket = require(«socket»)
local http = require(«socket.http»)
local response_body = {}
local response, status, header = http.request{
  method = «PUT»,
  url = base_url .. feed,
  headers = {
    [«Content-Type»] = «application/json»,
    [«Content-Length»] = string.len(json_data_str),
    [«X-ApiKey»] = apikey
  },
  source = ltn12.source.string(json_data_str),
  sink = ltn12.sink.table(response_body)
}

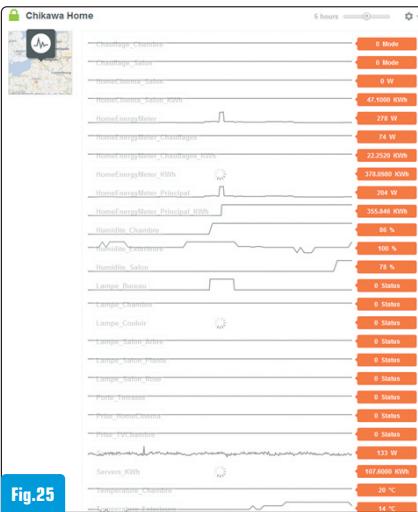
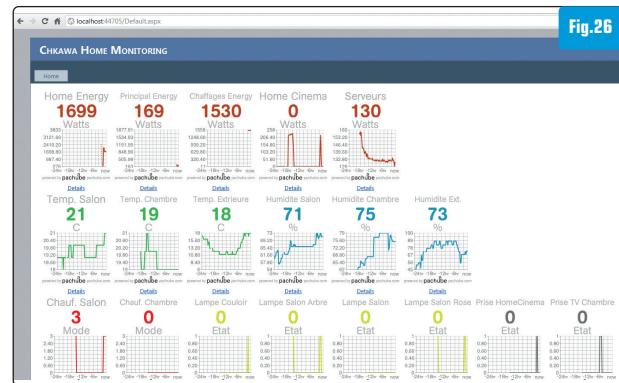
```

Où « datas » serait notre tableau de valeur contenant les différents états de nos périphériques Z-Wave :

```

local datas = {
  - EnergyMetering : Watts & KHH
  [ «HomeEnergyMeter»] = luup.variable_get(«urn:micasaverde-com:
  serviceId:EnergyMetering1», «Watts», 5),
  [ «HomeEnergyMeter_KWh»] = luup.variable_get(«urn:micasaverde-com:
  serviceId:EnergyMetering1», «KWH», 5),
}

```


Fig.25

Figure 25 - Interface Web de Cosm.com
Figure 26- Interface Web personnalisée pour l'affichage des différents "feeds"

- SwitchPowerh

```

[ «Prise_TV»] = luup.variable_get(«urn:upnp-org:serviceId:Switch
Power1», «Status», 3),
[ «Lampe_Bureau»] = luup.variable_get(«urn:upnp-org:serviceId:
SwitchPower1», «Status», 9),

```

- PilotWire

```
[ «Chauffage_Salon»] = luup.variable_get(«urn:antor-fr:service
Id:PilotWire1», «Status», 12),
```

- SecuritySensor

```
[ «Porte_Terrasse»] = luup.variable_get(«urn:micasaverde-com:
serviceId:SecuritySensor1», «Tripped», 11)
```

- TemperatureSensor & HumiditySensor

```
[ «Temperature_Salon»] = luup.variable_get(«urn:upnp-org:
serviceId:TemperatureSensor1», «CurrentTemperature», 26),
```

```
[ «Humidite_Salon»] = luup.variable_get(«urn:micasaverde-com:
serviceId:HumiditySensor1», «CurrentLevel», 28)
```

`}

Le code complet se trouve à l'adresse <http://pastebin.com/mHLzmRDB>.

Développer avec la Vera Lite

Pour finir cet article, voyons de quelle manière développer avec une Vera Lite. Il est possible d'utiliser le Lua, soit pour écrire des scénarios planifiés ou soit pour créer des devices virtuels (comme avec les plugins pour le fil pilote ou la météo Google). Pour contrôler sa Vera Lite depuis sa propre application, nous pouvons utiliser l'API http, en invoquant l'URL http://ip:3480/data_request?id=lu_sdata.

Le service nous retournera un objet JSON qui contiendra la liste et le détail de chaque pièce, la liste des scénarios et bien sûr la liste et le détail des différents équipements Z-Wave de notre réseau. Par exemple, pour le capteur ST817 du salon :

```
{
  «name»: «Température Salon»,
  «altid»: «el»,
  «id»: 11,
  «category»: 17,
  «subcategory»: 0,
  «room»: 4,
  «parent»: 10,
  «temperature»: «21»,
  «humidity»: «66»
},
```

Chaque réponse contiendra dans l'objet JSON les champs « loadtime » et « dataversion ». Grâce à cela, vous pourrez lors d'un prochain appel au service, lui définir quelle version vous avez déjà reçue afin d'obtenir seulement les valeurs modifiées.

Pour invoquer des scénarios ou un device en particulier, comme par exemple allumer une prise On/Off (un SwitchPower), il suffit de lancer une requête sur l'URL :

```
http://ip:3480/data\_request?id=lu\_action&DeviceNum=3&serviceId=urn:upnp-
org:serviceId:SwitchPower1&action=SetTarget&newTargetValue=1
```

Pour expérimenter cela, créons une page HTML avec un peu de



jQuery pour contrôler notre logement de manière visuelle. Pour modéliser votre habitat, rendez-vous sur HomeStyler.com. Une fois les visuels réalisés, nous les positionnerons sur notre page Web et rajouterons par-dessus des images représentant les lumières allumées ou éteintes. Côté Javascript, on charge la première fois les informations de la Vera et on vient définir un évènement sur le clic des images pour allumer ou éteindre la lampe.

```
$(document).ready(function () {
  $.get(`Vera.svc/status`, function (data) {
    categories = data.categories;
    devices = data.devices;
    // Ajout des contrôles des Switch lorsque l'on clique sur
    // l'image du device
    $('#controls').children('img').each(function (i) {
      $(this).click(function (event) {
        var deviceType = this.id.split('_');
        if (deviceType[0] == 'light') {
          if ($(this).attr('alt') == 'Switch Off') {
            $.get(`Vera.svc/setTarget`, { id: deviceType[1], device
Type: 'SwitchPower1', target: '0' });
          }
          else {
            $.get(`Vera.svc/setTarget`, { id: deviceType[1], device
Type: 'SwitchPower1', target: '1' });
          }
        });
      });
    });
    // Initial refresh
    refreshState(true);
  }, `json`);
});
```

La fonction "refreshState" boucle indéfiniment et permet de recevoir de la Vera les changements de valeurs afin de mettre à jour notre interface HTML (états des prises, des lampes ou des chauffages, consommation et valeurs des capteurs de température, d'humidité, de consommation et de sécurité).

```
var refreshState = function (recursive) {
  $.get(`Vera.svc/status`, { loadtime: lastRefresh, dataversion:
```



Fig.27
Vue partielle de la page Web représentant l'habitat (salon + terrasse). Les ampoules sont cliquables pour éteindre ou allumer les lampes. De plus l'interface affiche la consommation électrique, les capteurs météo et d'ouverture de la porte de la terrasse.

```
lastVersion },
function (data) {
  $(data.devices).each(function (index) {
    var categorie = getCategoryNameById(this.id);
    switch (categorie) {
      case «Dimmable Light»:
      case «Switch»:
        // this.status = 1 si allumé ou 0 si éteint
        break;
      case «Power Meter»:
        // this.watts pour la consommation instantanée
        break;
      case «Temperature Sensor»:
      case «Humidity Sensor»:
        // this.temperature et this.humidity
      case «Sensor»:
        // this.tripped = 0 si porte fermé (ou aucun mouvement)
        ou 1 si ouverte (ou mouvement)
        break;
    }
    // Display Refresh DateTime
    lastRefresh = $(data).attr('loadtime');
    lastVersion = $(data).attr('dataversion');
    // Refresh in 2 seconds
    if (recursive) {
      window.setTimeout(function () { refreshState(true); }, 1000);
    }
  }, `json`);
}
```

Conclusion

La domotique est un sujet passionnant qui ouvre beaucoup de possibilités. Ce tour d'horizon des différentes technologies vous donnera matière pour imaginer votre projet en fonction de votre habitat, de vos besoins et de votre budget. Les technologies filaires comme le KNX restent l'un des meilleurs choix mais ses contraintes d'installation et de budget rendent difficile leur démocratisation. Les technologies par courant porteur sont intéressantes mais peuvent souffrir d'un manque de fiabilité sur des réseaux électriques peu récents. Le « sans fil » semble être la solution la plus simple à mettre en place et des technologies comme le ZigBee ou le Z-Wave permettent de répondre à ses principaux défauts (réseau maillé pour une large couverture, retour d'état, sécurité, etc.).

Cependant, peu importe la technologie utilisée, le principal défi reste de privilégier l'usage à la technique. Il faut centrer son projet domotique sur l'objectif souhaité et choisir une ou plusieurs technologies pour y répondre en favorisant la simplicité d'utilisation. De plus, l'utilisation au sein de vos applications s'avère relativement simple comme présentée à travers cet article. Les technologies de développement habituelles combinées à votre installation domotique vous permettront de réaliser une maison communicante et autonome. La prochaine étape vers un habitat intelligent reste l'automatisation des équipements multimédia et l'orchestration de toutes ces tâches par une IA ou un agent intelligent, ce qui donnera une dimension supplémentaire à vos projets.

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

Choisir son langage de programmation



Eternelle question pour le développeur. Quel langage choisir ? Il est souvent imposé, à l'école, dans les projets, dans les entreprises / SSII. Mais malheureusement, pas toujours avec les bons arguments. Est-il toujours pertinent d'utiliser le langage à la mode ? Non, car sa popularité ne signifie pas forcément sa qualité technique et sa performance.

Le choix est très vaste mais finalement, nous tournons depuis 15 ans autour des mêmes langages : PHP, C++, C, Java, C#, éventuellement VB / VB.net. Objective-C connaît un regain de popularité à cause des terminaux iOS d'Apple. Ruby / Rails, Python, Perl sont très appréciés.

Delphi a toujours ses fidèles et certains langages fonctionnels sont très présents dans certains secteurs.

La popularité de tel langage dépend de nombreux éléments : communauté, outillage, médiatisation. Les qualités syntaxiques, les performances ne sont pas toujours des critères de choix. Parfois, le contexte impose de facto le langage. Pour le web, le choix est vaste et varié.

Le développeur ne doit pas se contenter d'un seul langage. Il doit absolument en connaître d'autres et regarder ce qu'il se passe en dehors de son langage de prédilection.



Le bon langage au bon moment

Les premiers langages informatiques sont apparus dans les années 1940. Dans les années 1950, les 3 principaux langages sont le Fortran (FORmula TRANslator) créé en 1953, surtout utilisé pour du calcul scientifique à haute précision dans les milieux universitaires, Lisp (LISt Processor) inventé en 1958, langage de programmation fonctionnelle, paradigme basé sur des fonctions mathématiques pures (théorie des fonctions) et le lambda-calcul. Et bien sûr, Cobol (Common Business Oriented Language) créé en 1959, pour la programmation d'applications de gestion. Les descendants de ces trois langages restent encore très utilisés aujourd'hui.

La tendance actuelle est une remontée en force des langages axés sur la concurrence et la distribution, comme Erlang et des langages fonctionnels avec Groovy, Scala, Clojure, F#. Il apparaît également tous les jours de nouveaux langages, conçus pour améliorer ceux existant déjà : Dart pour Javascript, Go et D pour C, Ceylon/ Fantom/Kotlin pour succéder à Java. Aujourd'hui dans le milieu professionnel, environ une douzaine de langages sont suffisamment matures, robustes, fiables et performants pour pouvoir répondre à pratiquement n'importe quel besoin. Mais lequel choisir ?

Le choix dépend des objectifs que l'on se donne. Apprendre un langage est difficile car chacun apporte des concepts intéressants issus d'un ou plusieurs paradigmes tels que Scala (paradigmes orientés objet et fonctionnel). Il faut donc une motivation réelle : par nécessité, parce qu'un projet commence, pour faire un projet personnel, ou par curiosité.

> Développement rapide de projet Web

Pour cet usage, la disponibilité d'un framework facile à installer et produisant immédiatement un résultat acceptable est déterminante. Les frameworks rapides Rails et Grails ont popularisé Ruby et Groovy. Python dispose aussi d'un framework de développement rapide avec Django. Ils arrivent avec une stack complète adaptée à la création d'applications Web avec, entre autres, la gestion dynamique des pages HTML et des accès à la base de données via un ORM et une base de données embarquée préconfigurée. Ces frameworks permettent de créer en quelques lignes de commande un projet, ses vues et des tables mappées sur la base de données.

Les commandes de création de l'application :

```
$ rails new blog
$ rake db:create
$ rails generate scaffold Post name:string title:string content:text
$ rails generate controller home index
```

Un exemple de formulaire :

```
<%= form_for(@post) do |f| %>
<div class=>field<>
  <%= f.label :name %><br />
  <%= f.text_field :name %>
</div>
<div class=>actions<>
  <%= f.submit %>
```

```
</div>
<% end %>
```

Le traitement du formulaire :

```
def create
  @post = Post.new(params[:post])
  respond_to do |format|
    if @post.save
      format.html { redirect_to(@post,
                                :notice => 'Post was successfully created.') }
      format.json { render :json => @post,
                                :status => :created, :location => @post }
    else
      format.html { render :action => «new» }
      format.json { render :json => @post.errors,
                                :status => :unprocessable_entity }
    end
  end
end
```

Enfin Scala ou Java peuvent être utilisés en développement rapide via Play!. Attention, même en Java, Play! ne fournit pas de framework de présentation Web manipulable, les vues sont des templates HTML. Play! se distingue particulièrement par une bonne intégration du debugging, les erreurs étant affichées directement dans le navigateur.

Un exemple de formulaire :

```
#{extends 'main.html' /}
#{set title:'Home' /}
<form action=>@{Application.sayHello()} method=>GET>
  <input type=>text name=>myName />
  <input type=>submit value=>Say hello! />
</form>
```

Le traitement du formulaire :

```
package controllers;
import play.mvc.*;
public class Application extends Controller {
  public static void index() {
    render();
  }
  public static void sayHello(String myName) {
```

```

    render(myName);
}

}

```

Tous sont des langages objets et implémentent un MVC. Le choix se fait souvent sur la pratique du langage et la disponibilité de librairies pour des domaines spécifiques. Par exemple, Python plus utilisé dans le monde scientifique dispose de librairies de calcul mathématique et de graphing. Ruby est plutôt orienté Web. Grails et Scala peuvent réutiliser les très nombreuses librairies Java.

JavaScript devient aussi plus présent côté serveur via des solutions comme Node.js et présente l'avantage d'utiliser un langage déjà connu des développeurs Web. Node.js est plutôt utilisé pour avoir une implémentation légère de processus asynchrones mais des frameworks commencent à se construire au-dessus de celui-ci.

```

un serveur basique en Node.js
var http = require('http');
function onRequest(request, response) {
  response.writeHead(200, {«Content-Type»: «text/plain»});
  response.write('Hello World');
  response.end();
}
http.createServer(onRequest).listen(8888);

```

Dart se présente comme une évolution de JavaScript pour la réalisation de WebApps en JavaScript côté serveur et pour améliorer le fonctionnement de JavaScript dans le navigateur.

> Construire une application Entreprise

Les applications Entreprise sont la plupart du temps en Java ou C#. Elles intègrent de nombreuses contraintes de conformité à des règles d'exploitation, d'intégration avec d'autres applications de l'entreprise, de sécurisation de traitements et de maintenabilité car elles sont faites pour durer.

C# et Java sont moins rapides à mettre en place, demandent des IDE plus sophistiqués, mais en contrepartie proposent un grand nombre de librairies pour la gestion des protocoles réseau, le reporting, l'intégration de Web Services, la gestion du transactionnel, des protocoles de sécurité.

Ils se ressemblent par certains aspects du langage et divergent par leur architecture. Les deux langages sont orientés objets et disposent d'une gestion automatique de la mémoire par un ramasse-miettes (Garbage Collector). Les deux types de serveurs d'applications fournissent un ensemble de services pour les applications (sécurité, gestion des transactions, gestion des accès à la base de données) pour Java sous la forme de serveurs d'applications distincts de l'OS (tomcat, jboss, glassfish ...), pour .Net sous la forme de services Windows. C# et .Net sont liés à Windows. Si la portabilité est un critère de choix, Java est le langage qui sera privilégié. L'environnement C# est un environnement éditeur qui dispose dans Visual Studio d'outils graphiques plus avancés pour faire des IHM. Visual Studio arrive avec l'ensemble des outils nécessaires, les IDE utilisés en Java tels qu'Eclipse nécessitent d'installer une sélection de plugins, ce qui peut paraître plus compliqué.

Ces IDE apportent des fonctions assez voisines de génération de squelettes d'application et de manipulation du code.

Les librairies et frameworks sont en moins grand nombre dans le

monde .Net et le choix semble plus restreint, mais c'est principalement parce qu'il n'existe souvent qu'une implémentation là où les développeurs Java ont développé des frameworks concurrents. Par exemple, .Net n'a qu'un framework de présentation là où Java en propose une quinzaine.

Java est un environnement plus foisonnant, plus réactif sur certaines innovations (les interfaces REST ont été disponibles très rapidement) mais cette profusion le rend aussi plus difficile à appréhender. C# en revanche a été plus réactif sur les innovations en matière de langage avec le support de syntaxes mixtes (LINQ - Language Integrated Query) pour alléger l'écriture des requêtes et l'évaluation vers F# un langage fonctionnel objet.

Exemple de requête LINQ :

```

var numQuery =
  from num in numbers
  where (num % 2) == 0
  select num;

// Query execution.
foreach (int num in numQuery) { ... }

```

Dès que les fonctions deviennent critiques, il est important d'avoir un code auditable et maintenable.

Les langages utilisés sur ces projets doivent donc posséder des outils de test unitaire (similaires à JUnit), les librairies de mock souvent nécessaires avec les langages objet pour créer le contexte, des outils de test fonctionnel (BDD, intégration à Fitnesse), des outils d'analyse de code. L'intégration dans des outils de suivi de la qualité comme Sonar sera appréciée. Dans ce domaine, les langages récents ou peu utilisés dans ce type d'application ont un outillage moins complet. Les langages qui implémentent la programmation par contrat comme Eiffel permettent une vérification formelle et la génération de tests automatiques via les préconditions (require), les post conditions (ensure) et les invariants.

```

note
  description: «Simple bank accounts»
class
  ACCOUNT
feature — Access
  balance: INTEGER
    — Current balance
  deposit_count: INTEGER
    — Number of deposits made since opening
  do
    .....
  end
feature — Element change
  deposit (sum: INTEGER)
    — Add 'sum' to account.
  require
    non_negative: sum >= 0
  do
    .....
  ensure
    one_more_deposit: deposit_count = old deposit_count + 1
    updated: balance = old balance + sum

```



```

    end
feature {NONE} - Implementation
  all_deposits: DEPOSIT_LIST
    - List of deposits since account's opening.
invariant
  consistent_balance: (all_deposits /= Void) implies
    (balance = all_deposits . total)
  zero_if_no_deposits: (all_deposits = Void) implies
    (balance = 0)
end - class ACCOUNT

```

Réaliser une application mobile

Le développement natif iOS (iPhone et iPad) nécessite l'apprentissage d'Objective-C qui est un langage objet dérivé de C et similaire à Smalltalk. XCode vient avec un outil graphique de conception des IHM qui décharge d'une bonne partie du code. Il ne disposait pas d'une gestion automatique de la mémoire jusqu'à l'introduction d'ARC cette année. L'environnement, entièrement maîtrisé par Apple, est assez homogène. Ce n'est pas un langage nouveau mais sa progression fulgurante dans le TIOBE montre qu'un bon outillage et une API très propre peut contrebalancer la difficulté d'apprendre un langage spécifique et sans gestion automatisée de la mémoire. Comme en C/C++, chaque classe à deux fichiers, un header .h et une implémentation .m. Objective-C utilise le message passing (syntaxe []) pour accéder aux opérations de l'objet.

L'interface d'une classe qui gère un bouton :

```

@interface MyViewController : UIViewController {
IBOutlet UIButton *button;
IBOutlet UILabel *label;
}
@property (nonatomic, retain) UIButton *button;
@property (nonatomic, retain) UILabel *label;
-(IBAction)buttonPressed;
@end

```

et son implémentation :

```

#import <ButtonFun_ViewViewController.h>
@implementation MyViewController
@synthesize button;
@synthesize label;
-(IBAction)buttonPressed {
    NSString name = @»World»
    label.text = [NSString stringWithFormat:@»Hello %@!», name];
}

```

Pour Android, l'implémentation native se fait en Java sur le moteur Dalvik. L'intérêt pour le développeur est que l'on est en terrain connu. Une grande partie des bibliothèques Java ont été adaptées, avec toutefois certaines limites, un smartphone, même récent, n'ayant pas encore la capacité mémoire et la puissance d'un ordinateur.

Un exemple de classe qui gère un bouton :

```

public class MyActivity extends Activity {
  protected void onCreate(Bundle icicle) {
    super.onCreate(icicle);

    setContentView(R.layout.content_layout_id);
}

```

```

final Button button = (Button) findViewById(R.id.button_id);
button.setOnClickListener(new View.OnClickListener() {
  public void onClick(View v) {
    // Perform action on click
  }
});
}
}

```

Les implémentations natives restent nombreuses pour des raisons de performance et d'intégration avec les fonctions de l'appareil. Mais pour éviter le codage sur diverses plateformes, on voit apparaître des frameworks multiplateformes comme Titanium Mobile et PhoneGap. Les deux se programmant en JavaScript même si l'architecture est différente; Titanium fournit un moteur JavaScript natif alors que PhoneGap tourne dans le navigateur.

Exemple de code avec PhoneGap :

```

<!DOCTYPE html>
<html>
  <head>
    <title>Notification Example</title>
    <script type=»text/javascript» charset=»utf-8» src=»phone
gap-1.0.0.js»></script>
    <script type=»text/javascript» charset=»utf-8»>
    // Wait for PhoneGap to load
    ...
    // Show a custom alert
    //
    function alertDismissed() { ... }
    function showAlert() {
      navigator.notification.alert('You are the winner!', alertDismissed, 'Game Over', 'Done' );
    }
  </script>
  </head>
  <body>
    <p><a href=»#» onclick=»showAlert(); return false;»>
Show Alert</a></p>
  </body>
</html>

```

> Ecrire un script

Pour cet usage on veut souvent une syntaxe allégée, un interpréteur (ou un compilateur à la volée) connu sous le nom de REPL (Read-Eval-Print-Loop) dans beaucoup de langages. Les langages à typage automatique ou implicite permettent de faciliter l'écriture en évitant la redondance et la recherche du type à utiliser.

```

Java (typage explicite) : File file = new File("data.txt")
Python (typage implicite) : f = open('data.txt')

```

Beaucoup de scripts manipulent des composants de l'OS ou des commandes et dans ce cas les langages assez proches du système comme Python ou Ruby sont appréciés au même titre que Bash ou Perl qui constituent la majeure partie du stock. Pour les usages courts, les langages sur JVM Java sont pénalisés par le temps de

démarrage de cette dernière. Par exemple, lire une clé dans un fichier de configuration et l'affecter à une variable en Ruby

```
favoritefruit = false
IO.foreach(`fruit.conf`) do |line|
  line.chomp!
  key, value = line.split(nil, 2)
  case key
  when /^([#;]|$/; # ignore line
  when «FAVOURITEFRUIT»; favouritefruit = value
  when /^./; puts «#{key}: unknown key»
  end
end
puts «favoritefruit = #{favoritefruit}»
```

Ruby est assez prédominant dans les outils DevOps car il est utilisé pour implémenter des outils de configuration management comme Chef ou Puppet, ainsi que Capistrano, un outil de déploiement.

Dans des cas plus applicatifs, c'est souvent le langage de développement de l'application qui dirige vers l'un ou l'autre des langages. On trouvera donc ici Python ou Ruby dans ces environnements, et plutôt Groovy dans les environnements Java.

L'existence de Builder pour générer du XML ou du HTML intégrés au langage simplifie énormément le code. Par exemple la génération d'une page HTML avec un MarkupBuilder en Groovy :

```
StringWriter writer = new StringWriter()
def mb = new MarkupBuilder(writer)
mb.html {
  head {
    title «Ma page HTML avec Groovy»
  }
  body {
    h1 «Ma page HTML avec Groovy»
    p {
      a (href:'http://groovy.codehaus.org', «et un lien»)
    }
  }
}
```

Les langages courants ont souvent des implémentations sur différentes plateformes, mais pas forcément pour tous les OS de serveur. A méditer avant de livrer un script d'installation en Lua pour un serveur sous Solaris.

> Mouliner des données

Pour le traitement d'image ou le calcul numérique intensif, les langages proches de la machine comme le C restent à l'oeuvre pour des raisons de rapidité. Plusieurs langages comme D et Go se posent en successeurs, mais sont pour l'instant peu répandus.

Avec l'augmentation de puissance des machines, la différence entre le C et Java est très réduite. Des langages plus lents mais plus faciles à utiliser prennent aussi la relève.

Les usages moins critiques comme les utilisations en analyse de données pour le Big Data ou dans le calcul financier recherchent des langages qui permettent d'exprimer de façon plus directe ce que l'on veut faire. Par exemple, les langages fonctionnels disposent généralement de fonctions permettant d'appliquer un traitement à

chaque entrée d'un tableau. En Java, le processus est assez laborieux et l'on perd vite de vue le processus à implémenter (ci-dessous, un exemple assez bête, calcul de x puissance 2 en Java et dans 2 langages fonctionnels).

Java :

```
interface IntToInt {
  int execute(int x);
}

int[] result = new int[array.length];
for (int i = 0; i < array.length; i++) {
  result[i] =
    new IntToInt() {
      public int execute(int x) {
        return x * x;
      }
    }.execute(array[i]);
}
```

A contrario, les langages qui disposent d'une fonction map permettent une écriture plus explicite.

```
F# : let result = Array.map (fun x -> x * x) [|1; 2; 3; 4; 5|]
Scala : val result = 1.map{i => i * i}
```

L'expressivité peut aussi exister au niveau des types. Clojure dispose d'un type Ratio qui représente des fractions. Scala et Groovy ont le type Range qui permet de définir simplement un intervalle de valeurs. Un autre moyen de rendre le code plus rapide à écrire et également de limiter les opportunités de se tromper est de réduire le texte à écrire en éliminant des redondances (les typages explicites), les symboles peu utiles pour le lecteur (par exemple les ; de fin de ligne) et en utilisant des raccourcis syntaxiques. Certains de ces raccourcis s'avèrent très pratiques comme l'opérateur Safe navigator operator ?. en Groovy, aussi appelé opérateur Elvis. Il permet d'éviter de tester si la valeur de chaque attribut est bien assignée :

```
def user = User.find( «admin» ) //this might be null if 'admin' does not exist
def streetName = user?.address?.street
```

Chaque paradigme oblige à penser le problème d'une certaine manière. Typiquement les langages fonctionnels amènent à découper le problème en fonctions, à les nommer et à se focaliser sur l'algorithme alors que l'objet met plutôt l'accent sur l'organisation des attributs et les opérations du domaine.

> Développer un jeu

Et si on s'amusait un peu ?

La plupart des moteurs de jeux connus sont en C/C++ à l'exception de XNA qui utilise C#. Si vous êtes adeptes de Java, PlayN est un game engine 2D multiplateforme où l'on code en Java.

Et n'oublions pas non plus JavaScript qui avec les Canvas HTML5, l'intégration de SVG et WebGL s'est permis de faire un portage de Quake. Il existe des frameworks dédiés comme JavaScript Game Framework. Unity quant à lui est un game engine 3D multiplateforme qui permet d'utiliser C# ou JavaScript.



> Ecrire un DSL

Pour l'écriture de DSL (Domain Specific Languages) les langages les plus adaptés sont ceux qui ont une syntaxe simple, permettant d'étendre des fonctions de base et de manipuler le code.

Le support des catégories que l'on trouve en Groovy et d'autres langages comme Objective-C, est une caractéristique utile. Cette technique permet d'étendre à la compilation une classe existante en lui ajoutant des méthodes, pour rendre les API plus expressives.

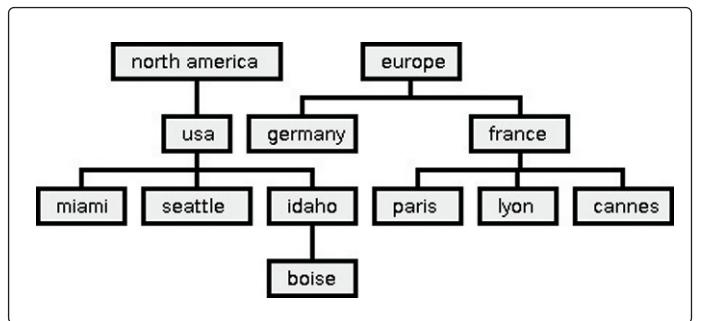
Par exemple, la classe Date est étendue pour supporter des méthodes week, days ...

```
import org.codehaus.groovy.runtime.TimeCategory
now = new Date()
use(TimeCategory) {
    when = now + 3.week - 1.days + 5.hours - 8.seconds
}
println when
```

Des langages comme LISP, Clojure ou Ocaml qui représentent le code sous la forme de structures de données permettent beaucoup de flexibilité dans la manipulation du code.

Par exemple, le DSL vijual permet de dessiner un graphe à partir d'un arbre de données. Il utilise une fonction Clojure draw-tree-image et la syntaxe normale des tableaux. L'usage de macro permet aussi de générer du code à la volée.

```
(draw-tree-image [[:north-america [:usa [:miami] [:seattle]
[:idaho [:boise]]]] [:europe [:germany] [:france [:paris]
[:lyon] [:cannes]]]])
```



> Se mesurer à la concurrence

Certains langages ont particulièrement mis l'accent sur la gestion de la concurrence ou la distribution de processus et sont de bonnes plateformes d'expérimentation.

La concurrence dans un process entraîne souvent des conflits pour l'accès aux objets en mémoire. Clojure est un langage fonctionnel pur, sans état. Il adresses directement ce souci en utilisant par défaut des données non mutables ou persistantes, tout en fournissant des modes de changement d'état des données en mémoire si nécessaire. Les différents modes permettent d'accéder en multi-thread au même état, de manière synchrone (atom), asynchrone (agent) ou transactionnelle (Software Transactional Memory)

Modification transactionnelle de la valeur de r :

```
(def r (ref nil))
(dosync (ref-set r 5))
@r
```

Modification asynchrone de la valeur de a via un agent :

```
(def a (agent 1))
(send a inc)
(await a)
@a
```

Erlang est très spécialisé sur la gestion de processus et leur robustesse. Démarrer deux process et échanger un message en Erlang :

```
Pid = spawn(server, start, [initial]),
register(server, Pid),
server ! hello,
Pid ! bye,
```

> Conclusion

Le choix des langages est actuellement assez ouvert, en particulier pour des applications non critiques. Au-delà de l'usage on doit aussi prendre en compte quelques critères généraux.

La maîtrise d'un langage compte pour beaucoup dans la capacité à répondre rapidement et de manière performante à la demande. La possibilité de suivre des formations et d'avoir le support d'une communauté en ligne fait aussi partie de l'écosystème et doit être évaluée, surtout si les développeurs sont novices.

Les langages n'ont pas tous les mêmes performances pures (voir le benchmark Debian), mais l'adéquation entre le principe de fonctionnement et l'usage compte également beaucoup. Par exemple Java est pénalisé pour la réalisation de scripts simples par le temps de démarrage de la JVM. Un programme Python termine généralement beaucoup plus vite. En revanche, Python gère moins bien le multithreading que Java qui sera plus performant pour un serveur sur une exécution plus longue de par sa capacité à gérer les threads ainsi que la capacité de scale out des serveurs d'application Java EE. Toutes ces informations doivent vous aider à trouver le langage le plus adapté pour un projet donné. Mais en tant que développeur, n'oubliez pas qu'il est bon d'avoir dans sa mallette différents outils pour des besoins différents et pour s'ouvrir l'esprit.

Bibliographie et liens utiles

Les exemples sont tirés de Rosetta Code ou des documentations officielles des langages.

Les langages de programmation

http://fr.wikipedia.org/wiki/Langage_de_programmation

Rosetta Code

http://rosettacode.org/wiki/Rosetta_Code

L'index TIOBE

<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

Comparatif rapide des langages

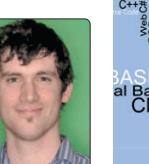
http://en.wikipedia.org/wiki/Comparison_of_programming_languages

The computer language benchmark game

<http://shootout.alioth.debian.org/>



Claude Falguière,
Consultante senior chez Valtech et



Marc Levi,
Consultant chez Valtech

Relecture : Jocelyn Batton, Consultant chez Valtech

Les critères de choix

Choisir son langage de programmation est une question plus complexe qu'il n'y paraît. On peut tenter de se fier à la réputation de tel ou tel langage ou bien encore à l'avis de son entourage. Mais quelle est en réalité la validité de ce choix ?

Le graphique du Tiobe Index (schéma 1) représente les statistiques observées sur l'utilisation des langages. Il montre les grandes tendances, mais ne présente pas de contexte particulier. Or de nombreux critères entrent en jeu : Les capacités du langage, le projet, les ressources, etc. Choisir un langage de programmation c'est tout d'abord en connaître plusieurs et savoir se poser les bonnes questions. Si le Tiobe Index présente ici un intérêt, c'est celui de montrer qu'il n'y a pas de réponse unique.

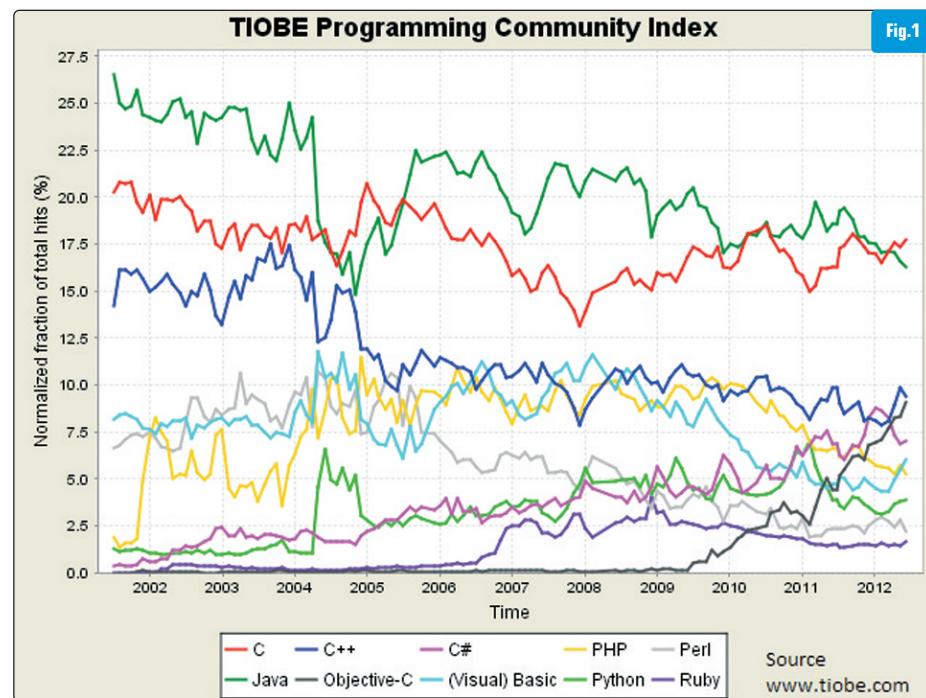
PARADIGMES

Il existe un nombre important de paradigmes de programmation, certains langages en implémentent d'ailleurs plusieurs. Il est inutile de les connaître tous, mais il existe des incontournables.

Par exemple, le paradigme impératif. Ses représentants les plus célèbres sont C, Pascal, Fortran, ... Ce paradigme, proche du fonctionnement de la machine et du modèle Turing, s'appuie sur des séquences d'instructions, des structures de contrôle et des effets de bord, regroupés au sein de procédures. Ce paradigme est implémenté par une écrasante majorité des langages de programmation les plus répandus.

L'opposé du paradigme impératif est le paradigme fonctionnel. Il repose sur le modèle du Lambda Calcul de Church [1] et consiste en une réécriture convergente d'expressions, avec l'utilisation de fonctions de premier ordre, dans un environnement généralement sans effets de bord. Les représentants les plus répandus de ce paradigme sont Lisp (bien connu des utilisateurs d'Emacs), Haskell et OCaml.

En pratique, on peut se permettre de légèrement généraliser en prétendant que tous les langages de programmation que vous rencontrerez appartiendront à l'un ou l'autre de ces deux paradigmes (bien que la réalité soit un peu plus complexe). La tendance actuelle semble être à une progression importante des langages fonctionnels, avec F# de Microsoft par exemple, mais surtout une volonté d'exploiter le meilleur outil au meilleur endroit. C'est pourquoi des lan-



Popularité des langages

gages très impératifs incorporent de plus en plus de traits fonctionnels, tels que Python, Ruby, C++(11) ou tout dernièrement PHP et leurs fonctions anonymes pour ne citer que cela.

LE TYPE DE PROJET

Parmi la myriade de langages qui existent, une première sélection peut être faite en se penchant sur le projet à réaliser. Il faut alors mettre en relation les caractéristiques du langage et ce qui est attendu.

> Des langages pour faire participer des non ou peu initiés

Beaucoup d'utilisateurs de solutions informatiques ne sont pas informaticiens, et encore moins programmeurs. Or, dans plusieurs cas, ces mêmes utilisateurs sont amenés à définir un grand nombre d'éléments, leur permettant d'utiliser ou d'adapter une solution logicielle à leurs besoins. Il est donc nécessaire de leur offrir un langage simplifié. Pour ce faire, les développeurs donnent accès à certaines fonctionnalités de leur sys-

tème via des langages de script et permettent ainsi de masquer la complexité sous-jacente du programme. On parle alors de DSL (Domain Specific Language, Langage Spécifique à un Domaine).

C'est souvent le cas dans le domaine des jeux vidéo : des jeux comme "World of Warcraft" (Blizzard) ou "Baldur's Gate" (Bioware) dont les différents modules des interfaces utilisateurs (boutons, mini-map, barre de statuts, etc.) sont définis en LUA. Il peut même être possible d'apporter des modifications complètes dans la série des "The Elder Scrolls" (Bethesda) grâce au TESScript, un DSL spécifique au moteur de ces jeux.

Lorsque l'utilisateur seul possède les besoins et l'expertise métier nécessaire, il n'est alors pas possible d'écrire le programme à l'avance, mais seulement un environnement de travail lui permettant d'exprimer ses besoins. L'exemple le plus classique est le calcul scientifique, où des outils comme MATLAB, SCILAB ou encore Wolfram définissent des langages de calcul scientifique. Ce genre d'outils permet à l'expert de décrire lui-même ses calculs sans s'occuper de la

VITE, ABONNEZ-VOUS !

jusqu'à **-50%**

Code, gestion de projets, développement web, mobilité, Programmez ! est à la fois votre outil pratique, des articles de code par les meilleurs experts et 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 : _____

Tél : _____ (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

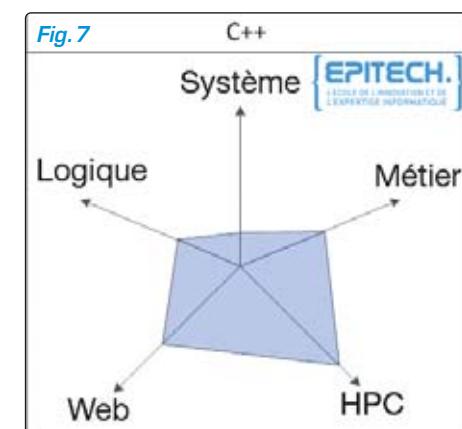
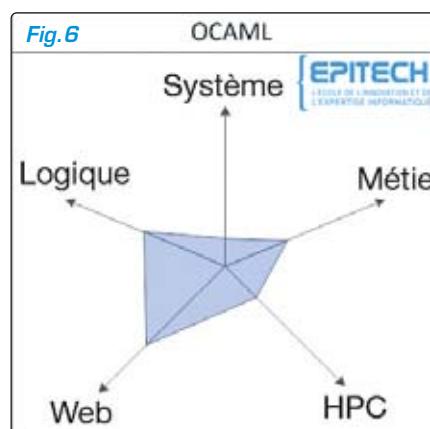
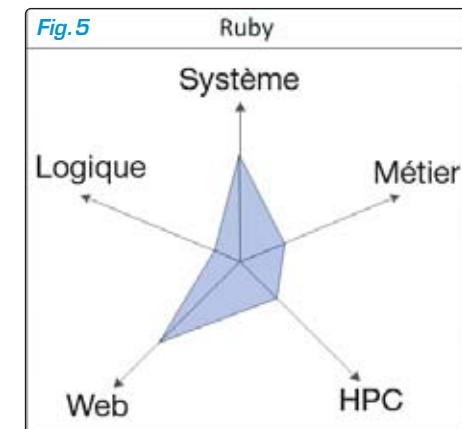
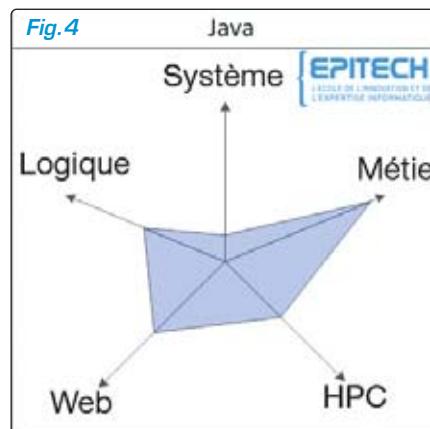
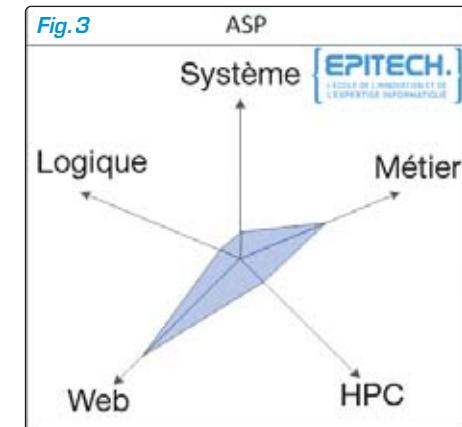
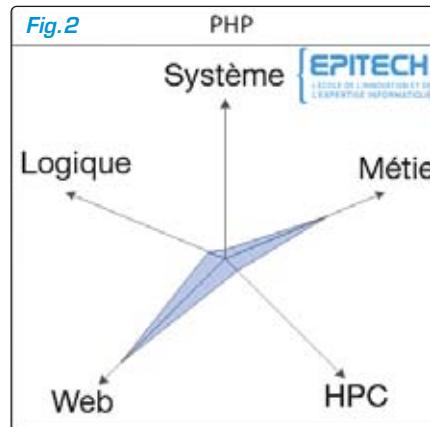
partie technique de l'exécution du calcul. L'Advanced Computer Science Epitech Laboratory (ACSEL - <http://acsel.epitech.eu/>) travaille actuellement sur la mise en place d'un langage spécialisé pour la recherche astrophysique. Ce langage devrait permettre aux chercheurs de décrire aussi simplement que possible les calculs nécessaires, le système s'occupant alors de les distribuer sur un grand nombre de machines afin d'obtenir un résultat dans des délais corrects.

En tant qu'utilisateur, le choix est posé par le système que vous serez amené à utiliser. Le véritable choix se pose ici si vous souhaitez définir une telle fonctionnalité dans votre projet. Comme nous pouvons le remarquer, les langages de script, notamment le LUA, sont régulièrement utilisés. Ils permettent en effet d'être rapidement intégrés aux projets et répondent à de nombreux cas. C'est dans le cas de contraintes et de besoins particuliers en expressivité, en style d'écriture, en fonctionnalités que l'on s'orientera vers le développement et la mise en place d'un langage spécifique. Cette dernière méthode étant nettement plus coûteuse et complexe.

Des langages pour le web

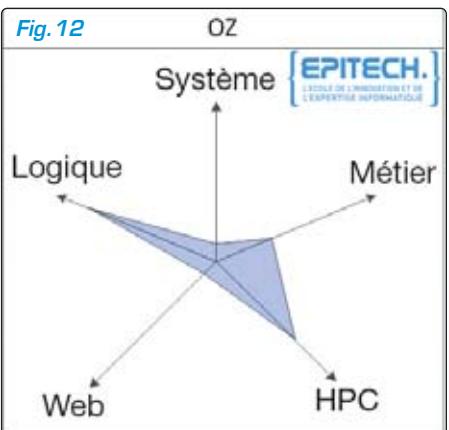
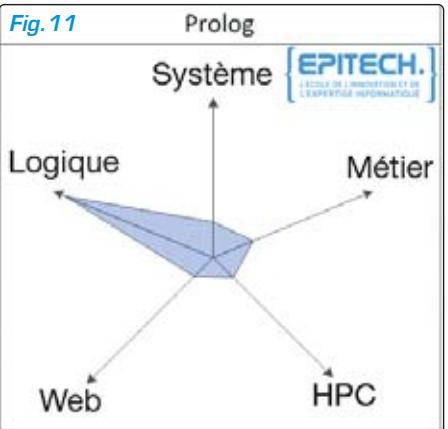
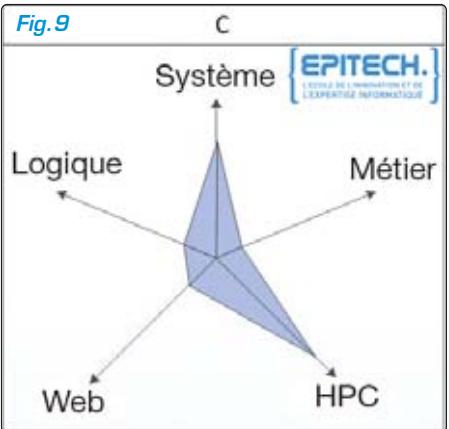
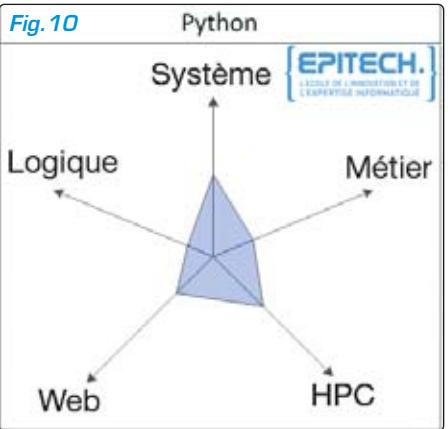
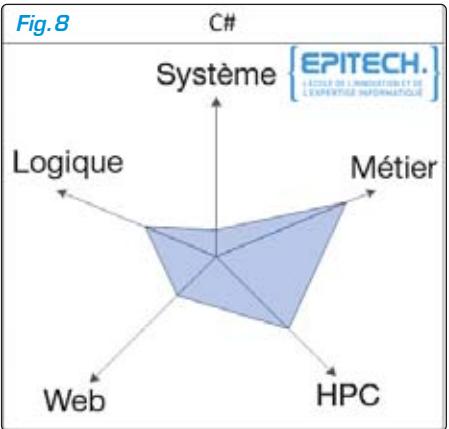
Le monde du Web est un peu à part dans le cœur des informaticiens. Enfermé dans un navigateur, distribué par des serveurs et donc rarement présent de manière durable sur le poste de l'utilisateur, l'écosystème de langages et de techniques du Web n'en est pas moins extrêmement riche, et apporte son lot de plaisirs aux développeurs... Mais aussi et surtout de problématiques bien spécifiques à ce monde entièrement en ligne. Lorsqu'un site ou une application web nécessite d'être dynamique, tel qu'un site de vente en ligne, un réseau social voire un CRM, on se retrouve face à deux problématiques: la capacité d'accueil et les fonctionnalités. Si l'architecture système et logicielle gère la première, le second problème pose la question des technologies logicielles.

Plusieurs environnements et langages viennent immédiatement à l'esprit en ce qui concerne la partie serveur du Web, celle chargée de répondre aux attentes du visiteur exigeant. PHP (schéma 2), ASP (schéma 3), Java avec JEE (schéma 4), Ruby avec Ruby on Rails (schéma 5) ou encore OCaml avec Ocsigen (schéma 6). Chacun a ses spécificités techniques et un environnement de prédilection, mais ce n'est pas la seule chose à prendre en compte. Chacun d'entre eux est soutenu par une communauté bien



spécifique, et offre tout un ensemble de modules répondant à de nombreux besoins. Un point commun entre tous ces langages, celui qui est le plus souvent critiqué, est que ce sont des langages interprétés. On leur reproche tantôt leur lenteur due au fait qu'ils sont interprétés, tantôt le laxisme de leur fonctionnement interne. Ces points ne seront pas discutés ici, cependant nous pouvons mentionner un fait souvent oublié: tout langage capable de servir du texte, et donc ce qui est attendu par un navigateur, est susceptible d'être utilisé du côté du serveur web. Bien que n'étant pas spécifiquement prévus pour cette utilisation, et pouvant

donc ralentir le développement, on peut trouver plusieurs outils facilitant leur utilisation. On mentionnera par exemple HipHop, conçu par Facebook, permettant de compiler du code PHP en C++ (schéma 7), C++ Web Toolkit, Ocsigen avec OCaml, etc. Il est ainsi possible de diminuer considérablement les ressources consommées par un site web reposant sur un grand nombre de transactions entre le navigateur et le serveur. Le terme d'application web, dernier aspect que nous allons aborder, n'est pas clairement défini. Ce terme passé dans le langage technique courant désigne un site hautement dynamique, mais surtout, destiné à



offrir le même type de services qu'une application avec un client lourd. On pourra ainsi mentionner les grands classiques, tels que la suite Office Live, ou encore le tentaculaire réseau applicatif de Google. Certaines technologies commencent à se positionner sur ce secteur, en particulier Ocsigen développé par l'Irill, dont les applications web sont le cœur de métier.

Tout ceci a notamment été permis par le développement d'AJAX (Asynchronous JavaScript And XML), qui rend capable une page web de demander du contenu au serveur via JavaScript, sans avoir besoin de la recharger entièrement. JavaScript est un

langage de script objet orienté prototype avec des traits fonctionnels importants, ce qui explique peut-être qu'il soit le plus souvent mal maîtrisé. Bien que souvent considéré comme lent, ou encore mal conçu, il reste une référence.

> Des langages pour développer rapidement des clients lourds

L'informatique a toujours été au cœur des sociétés organisées, sous une forme ou une autre, en tant que technique de traitement de l'information.

Citons à ce titre l'excellent TLF1 [2] qui nous dit qu'il s'agit de la "Science du traitement rationnel, notamment par machines automatiques, de l'information considérée comme le support des connaissances humaines et des communications dans les domaines technique, économique et social." Nous venons de souligner un aspect majeur, la finalité de l'outil, que l'on oublie parfois, dans un élan d'enthousiasme technique, au profit de la beauté de l'outil lui-même.

Les applications clientes lourdes (installées sur l'ordinateur de l'utilisateur et utilisant en grande partie ses ressources pour fonctionner) se voient qualifiées d'applications métier

du fait qu'elles ne sont conçues que pour répondre à un besoin professionnel particulier. Elles n'ont pas fondamentalement vocation à servir à développer de nouvelles technologies, faire avancer l'informatique en général ou être réutilisées en tout ou partie dans d'autres logiciels. Au sein d'une entreprise, la branche métier veut :

- des outils fonctionnels et pertinents
- rapidement
- à moindres frais

Plusieurs langages ont émergé qui sont capables de répondre à de telles contraintes. Certains ont probablement été conçus dès le départ avec ces objectifs en tête. D'autres ont évolué et de par leur adoption par une certaine communauté de développeurs, se sont vus dotés d'écosystèmes adaptés aux attentes métier. Les langages préconisés sont orientés objet, gérant de façon automatisée la mémoire, capables de s'exécuter sur toute plateforme disposant d'une machine virtuelle pour prendre en charge ladite exécution. C'est le cas de langages comme le Java, le F# et le C# (schéma 8) avec le Framework .NET. Ces langages se sont vus dotés au fil des années d'un écosystème absolument foisonnant de bibliothèques et d'applications prêtes à l'emploi.

Avec la productivité supérieure de ces langages par rapport à d'autres de plus bas niveau et la relative sécurité de développement qu'ils offrent (un développeur est moins susceptible de faire des erreurs simples de gestion de la mémoire par exemple), ils se distinguent en termes de rapidité de production et de mise sur le marché d'une application donnée.

Ils ont donc pu naturellement s'imposer là où le retour sur investissement, la rapidité de mise sur le marché et la facilité à trouver des développeurs priment sur la performance ou la qualité technique du langage, bien que celle-ci puisse rester tout à fait valable.

> Des langages pour la haute performance

Parmi les utilisations de l'outil informatique, l'une des catégories les plus obscures pour le grand public est certainement l'informatique haute performance (High Performance Computing, HPC). Particulièrement développée dans le monde scientifique et par certaines branches de l'industrie, le HPC vise à exécuter des calculs intensifs sur un nombre très important de données. La complexité est telle que certains calculs mettent plu-

sieurs jours, et même plusieurs mois, pour s'exécuter sur des machines. L'approche naïve consiste à multiplier le nombre de machines pour exécuter ces tâches dans des centres de calcul tels que l'IDRIS [3] en France. La multiplication des ressources, quand elle est possible, reste une solution très coûteuse. Une première étape consiste à utiliser au maximum le potentiel des machines disponibles, et d'utiliser le matériel adéquat. Dans cette optique il est important de choisir un langage offrant un accès complet aux ressources matérielles.

Le besoin de performance dans le monde du HPC demande donc l'utilisation de toutes les ressources disponibles. Les processeurs modernes permettent de nombreuses possibilités qui sont intéressantes à exploiter telles que les jeux d'instructions SIMD (Simple Instruction Multiple Data) et l'hyper-threading. D'autre part il est important d'avoir un grand contrôle sur l'action précise du logiciel. En effet, un gain substantiel de performance peut être obtenu si l'on fait attention aux problématiques de cache processeurs et aux optimisations de compilation (vectorisation, etc.). On évitera donc au maximum les langages de script (hors DSL réservés aux utilisateurs). On retrouve régulièrement les langages Fortran, C (schéma 9) et C++.

D'autre part, le matériel évolue et la course au « green computing » pousse les constructeurs et les grands centres de calcul vers de nouveaux types de processeurs spécialisés possédant un meilleur rapport GFLOPS/Watt [4]. Les plus médiatisés étant les GPGPU (General Purpose Graphical Processing Units) qui se pilotent via CUDA (pour les cartes NVIDIA) ou OpenCL (toutes cartes, soutenu par AMD). Il existe des passerelles permettant d'appeler ces morceaux de code depuis C++, Java, C#, Python. On notera que l'OpenCL a été conçu pour les architectures hétéroclites (par exemple, l'Intel 80-coreTeraScale) et pas seulement les GPGPU.

> Des langages pour le système

Au niveau du système, on peut différencier deux types de programmation : la programmation d'outils pour la gestion et l'administration et la programmation du système lui-même.

Dans le deuxième cas, le choix est relativement évident. Votre système d'exploitation a pour principale fonction d'être une couche

de médiation entre votre matériel (processeurs, mémoires, périphériques) et vos programmes. Pour effectuer cela, un contrôle absolu de la mémoire est indispensable, tout autant que la capacité à utiliser directement les instructions les plus spécifiques de votre processeur. En somme, le langage que vous utilisez doit être le plus proche possible de la langue naturelle de votre processeur, il doit être le plus bas niveau possible. L'assembleur a été créé afin de fournir une représentation textuelle plutôt que numérique des opérations du processeur. Il existe naturellement autant d'assembleurs que de processeurs et pour simplifier l'écriture et la lecture du code, un langage commun permettant l'accès aux fonctionnalités bas niveau. Il s'agit du langage C, langage impératif permettant de manipuler précisément la mémoire, l'inclusion de code assembleur spécifique à un processeur lorsque nécessaire mais aussi la possibilité de développer des niveaux d'abstraction très élevés. Le langage C reste LE langage pour le développement système.

De nombreux travaux sont menés pour essayer de proposer des constructions plus modernes et en particulier des fonctionnalités de programmation orientée objet pour la programmation système. C'est par exemple le cas du CBang [5], du projet Rathaxes[6] et du C++ dans le micro-noyau L4. La plupart de ces efforts restent encore du domaine de la recherche et n'ont pas encore de reconnaissance industrielle, voire de systèmes en production.

Si l'on revient à l'autre cas qui nous intéresse, à savoir programmer « proche du système », l'éventail de possibilités se fait déjà beaucoup plus large en termes de choix de langage. Si un besoin de contrôle direct de la mémoire se fait impérieux, alors on se tournera naturellement vers un des membres du binôme C/C++. Cependant, l'accès direct au matériel et à des routines de traitement performantes peut être encapsulé dans des bibliothèques logicielles. Celles-ci peuvent de plus être mises à disposition des autres langages, au travers d'interfaces spécifiques à chacun. Il convient de citer l'exemple de la communauté du langage Perl qui, autour de la plateforme CPAN, dispose d'une extension ou d'une bibliothèque pour à peu près tout ce qui est imaginable sur un ordinateur, des protocoles aux formats de données, des périphériques aux algorithmes, en passant par l'indispensable bibliothèque de simulation de processeurs quantiques. Les langages Ruby

et Python (schéma 10) héritent de cette tradition de langages interprétés visant l'universalité. Néanmoins, cette capacité d'interfaçage avec les bibliothèques C/C++ (natives) étant très répandue pour ne pas dire évidente, il est possible de communiquer efficacement avec le système dans la plupart des langages.

> Des langages pour la logique

Il existe un dernier type de projet. Fortement lié à l'intelligence artificielle et appliquée en grande partie à la problématique du traitement automatique des langues (TAL). Il s'agit de la programmation logique. Ces langages sont souvent basés sur un moteur d'inférence permettant à l'utilisateur de se concentrer sur la représentation des processus qu'il cherche à décrire. On parle souvent à cette occasion de programmation déclarative et non impérative. Le langage le plus connu pour la programmation logique reste le Prolog (schéma 11), pensé pour la réalisation d'un système de traduction automatique. Encore très restreint au cadre académique, cette mouvance a progressé, apportant de nouveaux langages tels que le OZ (schéma 12)[7] (et son framework Mozart) et le Gödel.

Les besoins de cette communauté sont principalement axés sur la recherche et ils ont donc besoin d'un système à la fois simple, c'est-à-dire leur permettant de se concentrer sur l'objectif. Il s'agit donc de langages interprétés, peu typés, permettant un cycle d'écriture et de test très courts. On note tout de même qu'après avoir été fortement décriés à cause de leurs performances, ces langages disposent pour la plupart de compilateurs, permettant d'une part une mise en production des travaux plus simple, et d'autre part un apport de performance permettant des tests sur des jeux de données plus importants.

Par la suite le Prolog a eu de nombreuses versions, comme le Prolog II, permettant d'aborder les opérations sur des logiques d'ordre supérieur. Ou encore le Prolog IV dont les nouvelles fonctionnalités permettent de travailler directement sur les problèmes de satisfaction de contraintes. Le OZ, de son côté, s'est orienté vers le calcul distribué de par ses fonctionnalités de "lazy execution" (permettant comme Haskell, la résolution de certains calculs au moment où l'on en a l'utilité si le programme n'a pas eu le temps avant).

FAIRE SON CHOIX !

Cet article ne peut pas être exhaustif. Nous avons abordé quelques langages mais il en existe quantité d'autres : ADA, Kobold, Go, D, etc. L'important est de se poser les bonnes questions et de rester flexible. D'une manière générale, il est conseillé de rester ouvert sur les langages. Il est important d'être capable de changer et de s'adapter à de nouveaux langages, afin de répondre au mieux aux besoins d'une entreprise ou d'un projet. Il arrive, bien trop régulièrement, que le langage soit choisi par rapport aux habitudes de l'équipe de développement. A notre sens, la maîtrise de l'outil informatique ne peut pas se limiter à un langage, mais à une compréhension plus complète et plus profonde de l'outil en lui-même et des paradigmes de programmation. Un langage n'est plus alors qu'une manière d'exprimer, de façon codifiée, ce que l'on souhaite obtenir. Cette liberté nous offre alors la possibilité de toujours choisir un langage adapté à la situation, un bon outil qui nous assurera les moyens techniques pour obtenir un résultat de bonne qualité.

Bibliographie

- [1] Alonzo CHURCH. Introduction to Mathematical Logic. Princeton University Press, 1996.
- [2] Le Trésor de la Langue Française Informatisé : <http://atilf.atilf.fr> - Définition consultable ici: <http://atilf.atilf.fr/dendien/scripts/tlfiv5/visuse1.exe?11;s=1303565100;r=1;nat=;sol=0;>
- [3] <http://www.idris.fr/>
- [4] Aaftab MUNCHI, Benedict R. GASTER, Timothy G. Mattson, James Fung, Dan GINSBURG. OpenCL programming guide, Addison-Wesley, 2011.
- [5] <http://blog.lse.epita.fr/articles/12-c---system-oriented-programming.html>
- [6] <http://code.google.com/p/rathaxes/>
- [7] Peter VAN ROY, Seif HARIDI. Concepts, Techniques, and models of computer programming. The MIT Press, 2004.



- # Julien Ballet
(Responsable LabFree/Epitech)
- # Philippe Berenguel
(ACSEL, Lab. Astek / Epitech)
- # David Girond
(Responsable Koalab / Epitech)
- # Nicolas Kamennoff
(Coordinateur rédaction / Responsable ACSEL / Epitech)
- # Thibaut Royer (ACSEL / Epitech)

Des langages pour débuter

Il est possible que vous soyez à la recherche de votre premier langage de programmation pour démarer en solitaire. Et dans ce cas précis, l'approche devrait être légèrement différente. Dans un premier temps, le choix du langage importe moins que la méthode et celle-ci doit être centrée autour d'une idée: de la pratique, encore de la pratique, toujours de la pratique. Résistez à l'envie de ne tester tout cela qu'à la fin du livre que vous avez acheté ou des articles que vous avez trouvés : même si chaque concept semble simple, l'intégration de ceux-ci dans un tout cohérent et leur utilisation instinctive passe par l'expérience. Soyons honnêtes, il est probable que vos premiers scripts ou programmes ne soient pas (très) fonctionnels. Néanmoins, nous devons tous passer par cette étape, vous n'y échapperez pas. Ne nous décourageons pas et revenons au choix de notre langage. Puisque nous avons vu que la méthode importe plus que le langage lui-même, il s'agit ici de considérablement réduire les possibilités de choix d'un langage plutôt que de l'élargir et de l'approfondir. Les paradigmes impératif et objet ayant tendance à être les plus répandus et les plus naturels pour ce qui touche aux interactions homme/machine, on aura tendance à se tourner vers eux pour démarer et à réserver le fonctionnel une fois les bases acquises. Il existe une notable exception pour les gens ayant un long parcours scientifique académique, pour qui le choix opposé pourrait

s'avérer plus naturel. Un premier langage devrait être relativement universel, à savoir utilisable pour une vaste majorité d'applications sans être forcément le langage parfait pour toutes : client lourd,



Web, script système. De plus, une large communauté ayant développé, documenté et éprouvé le langage, mais aussi un grand nombre d'outils et de bibliothèques logicielles est très important. Pour finir, on aura tendance à utiliser un langage interprété, ceux-ci permettant des cycles de développement courts, et ainsi pouvoir réaliser plus de tests en un temps donné. Ainsi, parmi les langages qui satisfont à ces différents points, nous citerons Ruby et Python qui sont de très bons langages pour, entre autres, démarrer la programmation. Rappelons qu'il ne s'agit pas d'un engagement sur le long terme mais d'un tremplin pour démarer. Vous reviendrez par la suite à cet article avec, espérons-le, une première bonne expérience en poche et l'envie de continuer l'aventure avec d'autres langages.

PROCHAIN NUMÉRO

Octobre 2012 - **n°156**, parution le 29 septembre 2012

Windows 8 de A à W

Sortie mondiale le 26 octobre de Windows 8. Préparez-vous au nouveau Windows : nouvelles API, nouvelle interface, nouveaux outils, Windows Store, version ARM.

Eclipse : Développez rapidement vos plug-ins Eclipse

Webmaster : Adobe CS 6 : quoi de neuf pour les développeurs et développeurs web ?

Je débute en C++ : C++ c'est beau, performant et fun. La preuve !



CSS3 de A à Z... ou presque

Ça y est, le CSS3 est parmi nous. Il permet de créer des sites bien plus stylisés qu'autrefois, avec beaucoup de facilité et de possibilités. Ce que nous allons voir, ce n'est pas une liste exhaustive de toutes les fonctionnalités de CSS3, mais plutôt ce qu'il faut en retenir

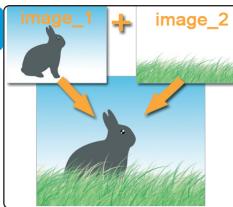
A COMME « ARRIÈRE-PLAN »

Arrière-plans multiples

CSS permet depuis longtemps d'ajouter un arrière-plan aux éléments HTML grâce aux attributs « background ». Mais il est à présent possible de superposer plusieurs fonds sur un même élément ! Pour cela, il suffit de déclarer chaque image l'une après l'autre en les séparant par une virgule. Les images déclarées en premier apparaissent au premier plan. Pour illustrer cela, créons une boîte « div » et appliquons-lui ce style :

```
.multiple-bg{  
    width: 403px;  
    height: 311px;  
    background-image:  
        url(images/image_2.png),  
        url(images/image_1.png);  
}
```

Fig.1



Nous obtenons alors cette superposition : [Fig.1].

Il est par ailleurs toujours possible de gérer la position et la répétition des images pour chacun des arrière-plans définis. Par exemple :

```
background-repeat: repeat-y, repeat-y;  
background-position: top left, top right;
```

Positionnement des arrière-plans

Fig.2



Autre bonne nouvelle, la propriété « background-clip » vous permettra de définir le point d'ancrage de l'arrière-plan à partir de la bordure (border-box), du padding (padding-box) ou du contenu (content-box) sans avoir à compter les pixels !

Voici un petit exemple pour illustrer cela : [Fig.2].

```
background-clip: border-box;
```

Taille des images d'arrière-plans

CSS3 nous donne la possibilité de redimensionner l'image de fond avec la propriété « background-size ». Cette propriété accepte des valeurs en pixels, en pourcentages mais également des valeurs pré définies :

- **auto** est la valeur par défaut
- **cover** redimensionne l'image de façon à ce que toute la place disponible soit couverte, tout en conservant les proportions de l'image. De ce fait, si l'image et la zone à couvrir ne sont pas proportionnelles, on ne verra pas toute l'image de l'arrière-plan.
- **contain** redimensionne l'image de façon à ce qu'elle soit entièrement visible, tout en prenant le plus de place possible.

Appliquons un arrière-plan à une boîte « div » de taille fixe :

```
div {  
    background-image: url('image_3.png');  
    background-repeat: no-repeat;  
    width: 200px;  
    height: 200px;  
}
```

Puis, faisons varier la taille de l'arrière-plan : [Fig.3].



B COMME « BORDURE »

Vous rêviez de bordure ultra personnalisée avec des coins arrondis pour les boîtes et boutons de votre site ? Le W3C a entendu vos prières !

Les coins arrondis

La propriété « border-radius » permet ainsi d'arrondir les coins des boîtes. Ce code applique le même arrondi pour tous les coins :

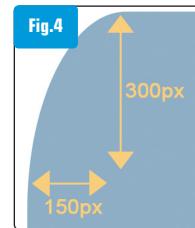
```
border-radius: 5px;
```

Celui-ci applique des arrondis différents pour chaque coin (la première valeur correspond au coin haut gauche, puis on tourne dans le sens des aiguilles d'une montre) :

```
border-radius: 5px 10px 8px 15px;
```

On peut également définir chaque coin indépendamment :

```
border-top-left-radius: 1px;  
border-top-right-radius: 2px;  
border-bottom-right-radius: 3px;  
border-bottom-left-radius: 9px;
```



Il est en outre possible de faire varier le rayon horizontal et vertical des ellipses afin d'obtenir des courbes plus ou moins allongées [Fig.4].

```
border-top-left-radius: 150px 300px;
```

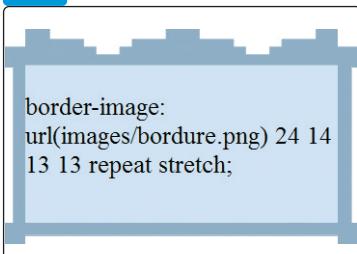
Les images de bordures

Vous pourrez désormais affecter des images aux bordures de vos boîtes en utilisant la propriété « border-image » [Fig.5 et 6]. Analysons le code suivant :

```
border-width: 24px 13px 13px 13px;  
border-image: url(images/bordure.png) 24 13 13 13 repeat stretch;
```

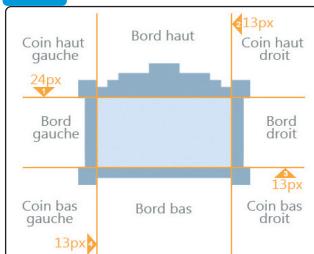
On commence par indiquer la taille des bordures souhaitées avec « border-width ». Puis, dans « border-image », on spécifie l'url de

Fig.5



```
border-image:
url(images/bordure.png) 24 14
13 13 repeat stretch;
```

Fig.6



l'image, les positions des lignes de découpe (vis-à-vis du bord haut, droit, bas puis gauche), et enfin, l'ajustement horizontal et vertical. Quatre valeurs sont disponibles pour l'ajustement :

- **Stretch** : l'image est étirée ;
- **Repeat** : l'image est répétée ;
- **Round** : l'image est répétée, mais si un nombre entier de morceaux ne rentre pas dans la zone, le morceau est redimensionné ;

C COMME « COULEUR »

CSS3 vous donne l'embarras du choix pour déclarer vos couleurs. Ainsi, pour écrire du texte en rouge, on peut utiliser une couleur nommée :

```
color: red;
Utiliser un code hexadécimal :
color: #ff0000;
```

Ou un code RGB décimal. Chaque paramètre (Red, Green, Blue) indique l'intensité de la couleur avec un entier entre 0 et 255 ou un pourcentage :

```
color: rgb(255, 0, 0);
```

Ou encore préférer un code HSL (Hue, Saturation, Lightness). Les valeurs indiquent respectivement la teinte (entre 0 et 359), un pourcentage de saturation et un pourcentage de luminosité :

```
color: hsl(0, 100%, 50%);
```

L'avantage des couleurs en RGB décimal et HSL, c'est que l'on peut faire varier l'opacité en utilisant rgba et hsla. L'opacité se code avec un décimal allant de 0 (entièrement transparent) à 1.0 (entièrement opaque) :

```
color: rgba(255, 0, 0, 0.5);
color: hsla(0, 100%, 50%, 0.5);
```

Notez que la propriété « opacity » permet d'appliquer de la transparence directement sur un bloc :

```
opacity: 0.5;
```

D COMME « DÉGRADÉ »

On parle parfois aussi de gradient. Il s'agit le plus souvent de faire un dégradé de couleur. Jusqu'à présent, on devait à chaque fois utiliser une image. Maintenant, on peut soit :

- Faire un gradient linéaire, via « linear-gradient »
- Faire un gradient « radial » via « radial-gradient » [Fig.7]

Cette « valeur » va ensuite pouvoir être utilisée sur les propriétés comme « background-image », « border-image » et « box-shadow ».

Fig.7



Linéaire

Radial

Fig.8



```
button {
background-color:#222;
background-repeat:repeat-x;
background-image:
url(..../img/gradient.png);
/* Fallback */

background-image:
-moz-linear-gradient(
top,
rgba(255,255,255,.75),
rgba(255,255,255,0));
background-image:
-webkit-gradient(
linear,
0% 0%, 0% 100%,
from(rgba(255,255,255,.75)),
to(rgba(255,255,255,0)));
background-image:
linear-gradient(
top,
rgba(255,255,255,.75),
rgba(255,255,255,0));

/* ... */
}
```

Comme on peut le voir, la déclaration pour Webkit diffère des autres. Heureusement, certains sites nous viennent en aide pour la déclaration d'un gradient : <http://gradients.glrzad.com/>

F COMME « FONT-FACE »

On peut importer des nouvelles polices de caractères via « @font-face ». Ce qui va nous permettre d'utiliser les « Google Fonts » disponibles ici : <http://www.google.com/webfonts>. Voici un exemple très simple pour associer la police « Buda:light » :

```
@font-face {
font-family: 'Buda';
src : url('http://fonts.googleapis.com/css?family=Buda:light');
```

On pourra alors utiliser sur nos éléments la police de la manière suivante :

```
div {
font-family: 'Buda', arial, serif;
}
```

K COMME « KEYFRAMES »

Le développeur Anthony Calzadilla propose sur son site une petite animation du célèbre AT-AT, ces grosses machines que nous pouvons voir dans les films de Star Wars (vous pouvez la visionner à cette adresse : <http://www.anthonycalzadilla.com/css3-ATAT/>). [Fig.9].

Fig.9



Le plus bluffant est que cette animation ne nécessite aucune ligne de JavaScript ! C'est du pur CSS3 ! CSS3 permet en effet de tordre, déplacer et changer l'orientation des éléments. Mais comment répéter un événement ? Ou comment enchaîner les événements ? Tout se joue avec la propriété «@ keyframes ». Celle-ci permet de préciser la valeur de certaines propriétés des éléments au fur et à mesure de l'animation. Essayons le code suivant (spécial WebKit, mais en changeant les propriétés, on peut le faire fonctionner aussi sur les autres navigateurs) :

```

<!doctype html>
<html>
<head>
  <meta charset=>utf-8>
  <title>Test</title>
  <style>
    .box{
      width: 200px;
      height: 200px;
      background: rgba(0,0,255,0.5);
      position: absolute;
      top: 100px;
      left: 90%;
      margin-left: -100px;
    }

    @-webkit-keyframes movingbox{
      0%{ left:90%; }
      100%{ left:10%; }
    }

    .box{
      -webkit-animation:movingbox 2s infinite alternate;
      /* Du moment que cette propriété est activée, l'animation est lancé */
    }
  </style>
</head>
<body>
  <div class=>box<></div>
</body>
</html>

```

On va voir alors un cube bleu se baladant de gauche à droite sur l'écran. Le mouvement ne dure que deux secondes, mais l'animation boucle de manière infinie en alternant le sens de l'animation (permettant ainsi cet aller-retour). A noter que l'animation s'exécute du moment que la propriété « animation » est affectée.

M COMME « MEDIA QUERIES »

Désormais, On doit adapter nos applications Web à une très grande variété de supports : cela va de l'ordinateur de bureau au smartphone, en passant par la télévision. Ce qui nous amène à modifier nos styles pour un meilleur agencement. Ainsi, le W3C nous propose l'utilisation de Media Queries. Il s'agit de déclarer des conditions afin d'appliquer une portion de styles. Ces conditions peuvent être liées au type de média (screen, tv, print, etc.), à la taille d'écran, mais également à l'orientation, au ratio et bien d'autres.

Pour illustrer cela, changeons la couleur de fond en fonction de l'orientation. En rouge pour le paysage, en bleu pour le portrait :

```

body {
  background-color: red;
}

@media all and (orientation: portrait) {
  body {
    background-color: blue;
  }
}

```

Pour en savoir plus sur ce sujet, vous pouvez consulter cet article d'Ethan Marcotte publié sur le site A List Apart : <http://www.alistapart.com/articles/responsive-web-design/>. A lire également, le livre « Responsive Web Design » du même auteur qui est vraiment très bien écrit.

O COMME « OMBRE »

Au rayon « nouveauté qui fait gagner du temps » on trouve également les ombres portées sur les boîtes et le texte.

Box-shadow

Ainsi, la propriété « box-shadow » prend jusqu'à six paramètres. Les deux premiers indiquent le décalage horizontal et vertical (peut être négatif). Le troisième correspond à l'intensité du « flou ». Le quatrième permet d'agrandir (valeur positive) ou de diminuer (valeur négative) la taille de l'ombre. Le cinquième définit la couleur. Enfin, il est possible de rajouter le mot-clé « inset » pour créer une ombre interne. Ainsi, le code suivant produit une ombre externe sur une boîte :

```
box-shadow: 20px 5px 12px -5px rgba(0,0,0,0.4);
```

Et celui-ci une ombre interne :

```
box-shadow: 20px 5px 12px -5px rgba(0,0,0,0.4) inset;
```



Fig.10



Fig.11

Voyez-le résultat : [Fig.10].

Text-shadow

Avec le texte, nous sommes limités aux ombres externes, mais on peut les superposer. Comme pour les boîtes, on peut définir le décalage horizontal et vertical, l'agrandissement de la zone et la couleur. Un petit exemple qui fait mal aux yeux : [Fig.11].

```
text-shadow: 3px 3px 3px red, 3px -3px 2px yellow, -3px 3px 3px blue;
```

P COMME « PERSPECTIVE »

Vous êtes fan de Star Wars ? Vous aimeriez recréer le fameux texte en perspective au début de chaque film ? Avec la perspective, rien de plus facile ! Il suffit de déclarer de la façon suivante :

```

.perspective {
  -webkit-transform: perspective(300px);
  -moz-perspective: 300px;
  perspective: 300px;
}

```

Nous passons alors de ceci : [Fig.12]. A cela : [Fig.13]. Plutôt sympa non ? Pour en voir plus : <http://www.seanslinsky.com/demo/star-wars-crawl-css3/>



Fig.12

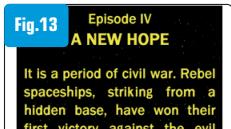


Fig.13 Episode IV A NEW HOPE



Fig.15

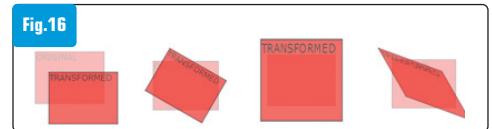


Fig.16

Fig.14



Un autre petit effet sympa consiste à faire des reflets de nos images comme dans un miroir. La propriété « box-reflect » permet de dire dans quelle direction va apparaître le reflet et à quelle distance. On peut également ajouter l'url d'une image ou un gradient qui va nous permettre de faire un masque (pour faire un effet de flou par exemple). Imaginons, on veut faire cela : [Fig.14]. Il nous suffit de faire ceci :

```
<!doctype html>
<html>
<head>
  <meta charset=>utf-8>
  <title>Test</title>
  <style>
    img{
      -webkit-box-reflect:
        right
        1px
      -webkit-gradient(
        linear,
        left top,
        right top,
        from(transparent),
        color-stop(
          0.4, transparent
        ),
        to(white)
      );
    }
  </style>
</head>
<body>
  <img src=>css3.png />
</body>
</html>
```

Plutôt sympa, mais pour l'instant, cette fonctionnalité est restreinte à Chrome.

T COMME « TRANSFORMATION »

W3C a mis un accent tout particulier sur la possibilité de déformer nos éléments ! En effet, de base, nous pouvons :

- Déplacer un élément
- Faire une rotation de notre élément
- Agrandir ou rétrécir notre élément
- Pencher, obliquer
- Faire un mélange des points précédents
- Appliquer les points précédents en prenant en compte la 3D !

Et si vous voulez vous rendre compte, regardez ce lien : <http://www.zachstronaut.com/lab/isocube.html>. [Fig.15].

Voici quelques exemples simples pratiques :

```
div.translate { /* Déplacer mon élément */
  transform: translate(50px, 100px);
}

div.rotate { /* Faire une rotation sur mon élément */
  transform: rotate(30deg);
}

div.scale { /* Zoomer sur mon élément */
  transform: scale(2, 4);
}

div.skew { /* Obliquer mon élément */
  transform: skew(30deg, 20deg);
}
```

Et ce qui nous donne tout ceci : [Fig.16].

T COMME « TRANSITION AUSSI ! »

Souvent, on change les propriétés CSS de nos éléments à la volée. Mais cela se traduit par un changement brutal de notre affichage. Du coup, on a tendance à appliquer du JavaScript afin de faire une transition plus douce. C'est l'objectif de cette nouvelle propriété ! Il est possible de définir sur quelles propriétés on doit détecter un changement, le temps de la transition ainsi que la répartition dans le temps de cette transition (par défaut, linéaire). Un exemple tout simple :

```
div {
  transition: left 2s linear;
}
```

Lorsque La propriété « left » est modifiée, une transition linéaire de 2 secondes est appliquée.

W COMME « WORD »

CSS3 nous propose enfin d'améliorer l'affichage de nos textes. Notamment avec la césure du texte. Placer une propriété « word-wrap » avec la valeur « break-word » permettra de « casser » : [Fig.17]. Il existe une propriété s'appelant « white-space », qui joue sur les espaces blancs pour obliger le retour à la ligne (comme le mode justifier dans nos éditeurs de texte). Mais quand un mot est plus grand que la zone le contenant, cela ne marche pas, et on voit alors le texte déborder. On peut aussi ne pas afficher entièrement le texte, et afficher trois petits points à la fin. Cela se fait via la propriété « text-overflow » avec la valeur « ellipsis ». Mais cette propriété ne fonctionne que sur des zones dont la taille est figée [Fig.18].

This paragraph has long words thisisaverylongwordthatisreallyoneword and again a longwordwithnospacesint

This paragraph has long words thisisaverylongwordthatisreallyoneword and again a longwordwithnospacesint

Fig.17

POUR FINIR

On vient de faire un tour d'horizon sur ce que peut nous apporter CSS3 dans le quotidien de nos développements des sites Web. Si vous souhaitez en voir plus, ou en savoir plus, regardez les liens suivants :

<http://www.smashingmagazine.com/> - <http://css-tricks.com/>

<http://html5live.fr/> - <http://www.w3schools.com/css3/>

Julien Roche, ingénieur d'étude Web / Web Mobile chez Objet Direct.

Marjorie Delrieu, ingénieur d'étude Web chez Objet Direct

Lore ipsum dolor sit amet, ...

Fig.18

Chef de projet « pompier »

Au secours d'un projet de site de jeux en ligne condamné à l'échec

Programmez ! vous propose pour ce numéro un témoignage exceptionnel, celui de la reprise d'un projet après échec. L'occasion de revenir sur un aspect moins connu et pourtant parfois essentiel de la mission d'un chef de projet, le rôle du « pompier ».

Lorsqu'en 2010 le marché du jeu en ligne s'ouvre, JOA, troisième groupe de casinos avec 21 établissements en France, entend couvrir l'ensemble de l'offre autorisée par la loi, paris sportifs, paris hippiques et poker. Après réflexion, il confie en 2009 le développement de son projet à une agence web, avec en objectif la mise en ligne du site joa-online.com en juin 2010. Un cahier des charges est élaboré, il reprend la complexité et les conditions drastiques de régulation, d'information et de contrôle imposées par l'ARJEL, l'Autorité de régulation des jeux en ligne. Complexité amplifiée par la volonté de JOA de n'offrir qu'une seule expérience d'inscription à ses joueurs, qu'ils fréquentent ses casinos ou qu'ils jouent en ligne. Si le joueur d'argent joue à tout, les jeux légaux comportent un vrai risque, c'est pourquoi tant sur la traçabilité des actes des joueurs, que sur la prévention de l'addiction ou la sécurité des données, l'ARJEL impose une règle du jeu incroyablement lourde !

Concernant l'informatique, l'affaire est également complexe. JOA n'est pas et ne souhaite pas devenir un éditeur, choix qui a été pris par certains de ses concurrents qui ont développé leur application. Chaque domaine de prestation auquel se confronte l'opérateur et ses clients – hébergement, banque, jeux, CRM, fidélisation, coffre-fort électronique, ARJEL, etc. - fait donc appel à un partenaire externe qui doit être connecté à la plateforme de jeu. Côté régulation, l'Etat a imposé une infrastructure technique qui permet de disposer d'une traçabilité à 100% de la totalité des actions du joueur. Quant au cahier des charges de l'ARJEL, il impose aux opérateurs de se placer sous une sphère technique, de contrôle et de contraintes, beaucoup plus que la loi elle-même.

Si au début tout se passe bien, des problèmes apparaissent dès la mise en ligne. Par exemple, lorsqu'un client alimente son compte, il est bien débité sur son compte



“ Nous ne pouvions pas ne pas proposer une refonte complète. ”

bancaire, mais la somme versée n'apparaît pas sur son compte JOA ! Pour peu qu'il renouvelle l'opération, il sera une nouvelle fois débité... sans effet sur son compte. Ajoutons à cela qu'un bug sur la plateforme peut impacter de nombreux programmes. Et les gestionnaires de JOA se retrouvent parfois contraints d'intervenir manuellement pour réaliser certaines opérations, un comble pour un site web... L'affaire dérape, le projet prend du retard, les coûts sont dépassés. JOA rate l'ouverture des marchés !

> L'audit mesure les dégâts... et l'ampleur de la tâche

JOA prend conscience de l'échec programmé de son projet. L'opérateur de jeux en ligne constate l'incapacité de son prestataire

à faire fonctionner sa plateforme et se voit obligé de trouver un relais. En septembre 2010, le groupe lance une consultation « pompier » et mandate Itelios - SSII de 80 personnes créée en 2003, et historiquement proche des solutions de Microsoft – pour réaliser l'audit de l'existant, en particulier de la plateforme applicative et de l'infrastructure physique en place chez l'hébergeur. L'objectif est de déterminer l'avancement du projet et le ratio de fidélisation par rapport au cahier des charges. Il est également qualitatif, avec la détermination de ce qui marche, de ce qui est à refaire, etc. Et l'évaluation de la tâche à accomplir pour rattraper tout cela.

« C'est un projet très technique, nous confie Fabrice Parizel, Directeur de projets chez Itelios. Le cœur de projet, c'est la plateforme de services qui doit fédérer des partenaires, des applications, l'ARJEL, des banques, etc., avec la difficulté de synchroniser et d'apporter du SSO. Mais le projet n'avait pas été abordé sous cet angle. Il aurait fallu concevoir une solution orientée vers la plateforme de services, avec échange, dialogue, transactions, workflow, site web, etc. De plus, il n'était pas sécurisé, ce qui est très mauvais pour la perception des utilisateurs finaux ». Devant l'ampleur de la difficulté, JOA se trouve contraint de mettre fin à sa communication, car à chaque fois qu'il essaie de faire venir un joueur sur son site, il prend le risque de le perdre définitivement...

> Du débogage au fonctionnel

« Lorsque nous avons audité le module de transaction des opérations bancaires, nous ne pouvions pas ne pas proposer une refonte complète, affirme Antonin Brachet, chef de projet Itelios en charge de JOA. Simplement, il faut l'expliquer au client en parallèle de ce qu'il veut ». Décision est donc prise de refondre la plateforme transactionnelle. JOA

fait de nouveau le choix de retenir Itelios pour la reprise du projet. A la SSII d'assurer la refonte, la récupération des données et la finalisation des développements, notamment de la couche transactionnelle. Les services ne sont pas interrompus, la phase d'analyse et de transfert des compétences est lancée. Itelios devient indépendant début décembre 2010. Puisque l'accumulation de bugs vient altérer la capacité fonctionnelle des outils, la priorité est donnée au débogage, avant de s'attaquer aux nécessités fonctionnelles. « *JOA fait appel à de nombreux prestataires et partenaires extérieurs, constate Fabrice Parizel. C'est pourquoi la plateforme en ligne est un assemblage de nombreuses briques. Et c'est toujours et par définition une source de problèmes* ». Ce que confirme Antonin Brachet : « *C'est un projet fonctionnellement riche en termes de règles métier auxquelles s'ajoutent les contraintes de l'ARJEL et un nombre important de partenaires à piloter. Il impose une montée en charge et une masse d'informations sur les applicatifs, les paramètres et les intervenants du projet. Ce n'est pas intuitif et cela nécessite du temps* ». Puisque le système existe et doit tourner, Itelios va désormais procéder par petites touches, repérer les bugs, identifier les modules qui posent problème et les remplacer par quelque chose de plus solide. Dans quel état d'esprit se retrouve le chef de projet en charge de JOA ? « *Ma première*

“ Ce qui est important au démarrage, c'est surtout de bien connaître son client. ”

anxiété a porté sur la récupération d'un existant riche, et qui d'un point de vue technique présentait un risque important en faisant intervenir une base de données Oracle avec des technologies Microsoft. Ma seconde appréhension a porté sur l'éventuelle impossibilité de maîtriser la richesse fonctionnelle de l'existant, malgré une phase de transfert de compétences de trois semaines. Mon troisième questionnement a porté sur les interactions avec les différents partenaires face au changement de prestataire principal... allaient-ils maintenir le même état d'esprit ? ». S'ajoute à cela la contrainte d'un client qui a déjà investi beaucoup d'argent et auprès duquel il faut justifier de nouveaux développements, comme la refonte complète du module transactionnel, ce qui n'est pas négligeable ni en charge ni en coût.

Comment Antonin Brachet allait-il aborder la chose ? Dans un premier temps, il va s'efforcer d'assimiler un maximum d'informations, d'un point de vue technique comme fonctionnel. Il va profiter du transfert de compétences, « *tout en reconnaissant qu'il ne sera pas possible de tout connaître dès le départ* »,

avec des fonctionnalités que nous avons continué de découvrir des mois après. Il est impossible de tout connaître ! ». En seconde approche, il va construire une équipe solide en termes de compétences, apte à intervenir sur une combinaison de technologies sur lesquelles Itelios n'a pas l'habitude d'intervenir. Placer des profils experts va lui permettre de sécuriser le projet.

> Technologies et méthodologies

Tout d'abord, s'agissant d'un projet avec récupération de développement, il n'a pas été nécessaire de mettre en oeuvre des méthodologies de design. Récupérant également l'existant sur la partie modèle conceptuel de données, Itelios n'a pas non plus rencontré l'occasion de le manipuler. En revanche la SSII va mettre en place des technologies basées sur l'outil collaboratif de Microsoft, Team Foundation Server. Pour la partie code source, le pilotage du projet est couplé à MS Project. « *Pour le reporting et la facturation, cela simplifie considérablement les choses, nous confie Antonin Brachet. Microsoft offre le déploiement et le build automatisés, ce qui permet de limiter les problèmes annexes* ». Itelios se dote également de l'outil collaboratif de Microsoft, Team System, un tableau de suivi accessible de l'équipe jusqu'au client, principalement organisé autour de deux entités : les bugs pour le correctif et le 'requirement' pour l'expression de besoin sur l'évolutif.

Itelios va commencer par récupérer l'intégralité du code et le réorganiser. Les process passent par une phase d'implémentation, refonte et développement des évolutions, puis une phase de test en interne. Elle s'appuie sur les solutions Microsoft pour réaliser du build automatique, puis le déploiement sur un environnement de recette.

Le client réalise une phase de test et de validation, afin d'éviter que les tests soient biaisés par ceux qui ont réalisé les développements. Puis vient le moment de donner le go pour la phase de validation finale sur un environnement de pré-production et enfin le passage en production. « *Au début, les tests étaient réellement compliqués à réaliser avec les effets de bord* », nous confie Fabrice Parizel. Autre difficulté, l'infrastructure initialement déployée par JOA s'appuie sur des bases de données Oracle... (lire l'encadré).



L'expérience du « pompier »

Fabrice Parizel, Directeur de projets chez Itelios

« Ce n'est pas la première fois que l'on se retrouve dans ce contexte, que l'on nous appelle à la rescoussse sur un projet où le client est en contradiction avec son fournisseur. Il n'y a pas spécialement de process qui sorte du cadre de nos pratiques habituelles.

Les notions de projet et de pilotage sont dans les gènes du directeur de projets. Mon job, c'est d'avoir des projets, des process, du planning et du respect du client. L'une des problématiques, c'est que par définition nous allons avoir un relationnel plus tendu avec le client. Cela dépend beaucoup des gens qui sont en face de lui et, au delà de la compétence

du savoir refaire, de notre capacité d'affronter des problèmes en cascade. Ce qui est important au démarrage, c'est qu'il faut prendre connaissance du现实, connaître les gens avec lesquels nous travaillons, et surtout bien connaître son client. Il est plus difficile de refaire que de faire. C'est un élément fondamental, car lorsque l'on reprend, découvrir et refaire un travail est toujours compliqué. Nous avons d'ailleurs fait plus que découvrir les problèmes, il nous a fallu descendre très profondément pour les comprendre dans un univers que l'on ne connaît pas. Si l'on reprend l'existant et que ça ne marche pas dans les six premiers mois, c'est de la gestion de problèmes. Nous les accummons et les traitons les uns derrière les autres. Si un client est dans l'inconfort, il faut que ce soit réglé tout de suite ».

Au départ, les phases de mise en production sont assez élevées, quasiment une mise en production par semaine. La démarche est également agile afin de répondre à la demande du client de stabiliser le plus rapidement possible une plateforme sur un processus plutôt itératif. Antonin Brachet nous décrit les processus de production : « Chaque entité a un cycle de vie. Elle passe successivement entre les mains du chef de projet pour la validation et la ré-affectation à un membre de l'équipe de développement, puis chez un développeur. Un fois réalisée, l'entité revient au chef de projet pour test, puis rejoint l'environnement client pour continuer le processus de test. Et de nouveau le chef de projet pour rejoindre l'environnement de recette pour test chez le client ». Une fois la correction testée sur l'environnement de production, le client revient au dashboard et transforme le statut de l'entité. Il décide alors soit de rouvrir le problème et l'expression de besoin, et à ce moment là de partir sur une ré-itération, soit la correction ne lui convient pas, soit elle enrichit l'entité si les tests sont significatifs et entraîne une clôture. « Puis processus classique de mise en validation en pré-production et de mise en production finale. »

N'oublions pas non plus les réunions hebdomadaires... Sur un projet de cette ampleur, la mise en place d'un comité de pilotage avec le client s'impose.

Surtout qu'il nécessite beaucoup de correctifs, avec un client qui dans le même temps éprouve le besoin de faire évoluer son service d'un point de vue fonctionnel. « Au client de préciser ce qui est prioritaire en termes correctif et évolutif, et en termes de réorganisation. A lui de nous fournir des deadlines en correspondance à ses attentes », confirme Antonin Brachet.

En juin 2011, après un semestre de travaux forcés, la partie transactionnelle, en particulier le transfert d'argent, est remise en ligne. Et le déploiement de la plateforme est stabilisé en juillet 2011. « Nous sommes sortis de la phase de risque de régression en

octobre 2011, et aujourd'hui de la phase des correctifs, qui sont enfin stabilisés. Elle a nécessité un an de processus. Nous n'avons ralenti réellement qu'en mars 2012 », conclut Fabrice Parizel.



La problématique de marier Microsoft et Oracle

Antonin Brachet, chef de projet Itelios

« L'expression problématique est bien adaptée. Oracle n'est pas le SGBD le mieux intégré avec .NET... Ce n'était pas n'importe quel projet et cela a été particulièrement compliqué sur certains points. Nous avons dû interagir directement avec le support d'Oracle. D'un point de vue technique, faire intervenir une base de données Oracle avec des technologies full Microsoft, qui consomment plutôt du SQL Server, représente un risque important. Utiliser Oracle dans le contexte de JOA, avec la contrainte d'un système quasi intégralement transactionné, présentait un gros point d'interroga-

tion. L'interaction entre les environnements Oracle et Microsoft n'est probablement pas le plus compliqué, des providers existent des deux côtés. Le plus compliqué a été de projeter cela dans le contexte transactionnel avec les transactions distribuées. Surtout qu'initialement et historiquement sur la base Oracle, il y avait un DB Link en place, qui contribuait assez fortement à ralentir les performances, et entraînait énormément de problèmes. Nous avons fait sauter DB Link et cela a retiré une bonne partie de nos problèmes. Nous avons épluché un certain nombre d'erreurs, surtout dans ce contexte transactionnel ».

> La satisfaction du chef de projet

Que retire Antonin Brachet, le chef de projet Itelios qui a porté le projet JOA, de cette expérience ? « C'est clairement un motif de satisfaction important. Nous sommes arrivés dans un contexte difficile pour de nombreuses raisons : un client qu'il fallait rassurer car il restait sur des échecs, un client qui avait des problématiques business fortes face aux utilisateurs finaux, avec des problématiques de jeux qu'il a fallu régler en parallèle de la phase de stabilisation. Nous avons réussi à jongler entre le correctif et le fonctionnel afin de stabiliser d'un point de vue technique, et avec des technologies qui n'étaient pas les plus compatibles ». A un moment donné, Antonin Brachet s'est retrouvé à la tête de 15 personnes intervenant sur le projet, ce qui n'est pas négligeable en termes d'équipe de développement et de process à mettre en oeuvre. « Mais

ma principale problématique a été de piloter le projet dans son ensemble, avec tous les partenaires, même si JOA est resté présent, tout en traitant les problèmes

techniques en termes de fonctionnalités, en répondant à toutes les exigences légales, et en mettant en place en parallèle un monitoring de la plateforme. Cela n'a pas été une mince affaire ! »? Et à titre personnel ? Anto-

Conseils de chef de projet « pompier »

A la suite de l'expérience JOA, Antonin Brachet, chef de projet Itelios, nous donne quelques conseils :

Etre pragmatique. Toujours revenir à la réalité et être terre à terre. Procéder module par module, ce qui revient à la démarche agile. Recentrer les besoins sur le client. Ce n'est pas à nous de définir ce dont le client a besoin, c'est à lui, client, de prioriser ce qu'il veut et de définir l'ordre de ce qui doit sortir en fonctionnalités. Etre à l'écoute du client, c'est juste indispensable et à la base, nous sommes là pour l'aider et le conseiller. Utiliser les bons outils tout en sachant s'en servir. Ne rien céder sur la qualité de réalisation, des tests et du monitoring.

nin Brachet éprouve la satisfaction d'avoir su gérer un projet particulièrement conséquent, tant en équipe qu'en budget. Et à connotation internationale avec des partenaires étrangers.

Sur des technologies qu'Itelios n'a pas l'habitude de mettre en oeuvre ! Et de faire le constat à l'arrivée de la satisfaction du client. « Au départ du projet, nous ne nous attendions pas à réussir aussi bien. Nous sommes arrivés au final à un processus quasi exclusivement évolutif sur une base stable, avec la satisfaction d'avoir travaillé sur des technologies intéressantes. »

Yves Grandmontagne

“ Il est plus difficile de refaire que de faire. C'est un élément fondamental. ”

Test Driven Development avec Visual Studio 2012

Dans nos précédents articles, nous nous sommes intéressés à certains aspects liés à la qualité du code. Nous avons abordé les principes S.O.L.I.D. Dans ce nouvel article, nous souhaitons continuer en abordant la qualité du développement ; un sujet qui est plus que jamais d'actualité avec l'utilisation des méthodologies Agile et particulièrement le Test Driven Development.

L'objectif de cet article est d'être pragmatique et orienté projet : nous souhaitons montrer que les nouveautés apportées dans la nouvelle version de Visual Studio 2012 permettront aux développeurs de faciliter et de rendre naturel le développement TDD. Il s'agit également de simplifier et limiter l'intégration des différents outils dans un processus d'intégration continue.

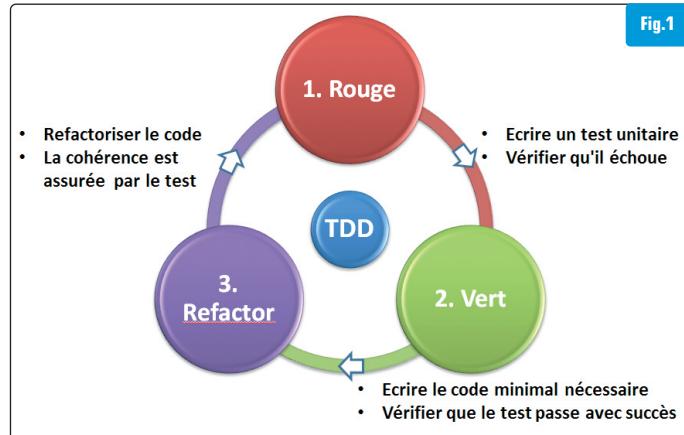
Le Test Driven Development (développement piloté par les tests) décrit une technique de développement de logiciel qui consiste à écrire des tests unitaires avant l'écriture du code source.

Voici les différentes étapes du Test Driven Development [Fig.1] :

- Ecriture du premier test pour une nouvelle fonctionnalité puis vérification de l'échec du test (l'implémentation de la nouvelle fonctionnalité n'étant pas encore réalisée)
- Implémentation du code minimal nécessaire pour passer le test puis vérification du succès du test (l'implémentation fournit alors le comportement attendu)
- Refactorisation et optimisation du code, les tests assurent la cohérence des fonctionnalités

L'utilisation du Test Driven Development permet ainsi l'obtention d'un code source très fiable, prévisible, robuste et, après refactorisation, hautement optimisé. Les tests assurent un comportement correct de celui-ci indépendamment des situations auxquelles il pourra être exposé. Le code source devient alors valide en toutes circonstances. Pour créer de bons tests unitaires, il faut d'abord réfléchir à la conception du programme et à la façon dont il va être utilisé. Il faut éviter une précipitation dans l'implémentation avant d'avoir défini les objectifs. Des erreurs de conception peuvent ainsi être identifiées et résolues plus vite. L'implémentation passe seulement après la validation de la conception complète via les tests unitaires. Les tests unitaires deviennent une sorte de spécification générale décrivant les fonctionnalités du programme de manière unitaire.

Quant à la refactorisation, l'optimisation et la restructuration du code peuvent se faire sans risque car ils sont vérifiables ; les tests unitaires assurant alors la non-régression technique et la cohérence du comportement. Le comportement étant exprimé dans les tests unitaires, ils valident que le programme se comporte toujours de la même façon si les tests passent avec succès. De plus, en associant la méthode d'Extreme Programming (XP) et la programmation en binôme, on obtient un code de très bonne qualité.



Contrairement à d'autres méthodes où les tests sont abordés en fin du cycle, le Test Driven Development met donc le focus sur les tests dès les premières étapes du cycle de développement. Aucun développement ne peut commencer avant que les tests soient conçus et implémentés. Cette méthode a donc un large impact sur l'organisation de l'équipe de développement.

Support des frameworks de tests

Les versions précédentes de Visual Studio permettent l'utilisation d'autres frameworks de tests, mais cela implique certaines limitations comme par exemple :

- L'utilisation d'une application tierce pour jouer les tests unitaires comme avec Gallio : l'utilisation de l'application Icarus Runner est indispensable pour pouvoir exécuter les tests unitaires.
- La couverture de code native ne fonctionne pas sans l'utilisation d'autres plugins (NCover).
- L'exécution des tests unitaires diffère en fonction des plugins : un test utilisant MSTest ne s'exécute pas de la même manière s'il est lancé depuis Visual Studio 2008 ou depuis Resharper, ceci peut poser problème lors de l'utilisation d'un processus d'intégration continue.

Au final avec l'ancienne version de Visual Studio 2010, on finit par multiplier les plugins et les outils pas toujours compatibles les uns avec les autres, ce qui ne permet pas vraiment de tirer avantage des fonctionnalités avancées des frameworks de tests spécialisés. Dans un environnement où le nombre de projets est conséquent, on a besoin de pouvoir utiliser un framework de tests plutôt qu'un autre en fonction des spécificités du projet et des fonctionnalités de celui-ci. Il est donc important de pouvoir proposer aux équipes de développement un large choix de frameworks de tests avec un coût d'intégration limité (dans les outils de développement comme dans les outils d'intégration continue). L'une des belles surprises de cette nouvelle version de Visual Studio est justement le support de plusieurs frameworks de tests unitaires tels que :

Pour .NET :

- NUnit (<http://nunit.org>)
- xUnit.net (<http://xunit.codeplex.com>)
- MBUnit (<https://github.com/Gallio/Gallio-VS2011-Integration>)

Pour Javascript/HTML :

- QUnit & Jasmine (<http://chutzpah.codeplex.com>)

Pour C++ :

- MSTest Native (<http://aka.ms/mstest-native-vs11>)

Prenons, par exemple l'intégration du framework de tests NUnit. Après l'installation de NUnit via NuGet [Fig.2], l'adaptateur de tests est téléchargeable sous la forme d'un plugin via le gestionnaire d'extension [Fig.3], disponible dans le menu « Tools/Extensions and Updates... » de Visual Studio 2012 [Fig.4].

Cette fonctionnalité est complètement intégrée à Visual Studio et les adaptateurs de tests sont disponibles gratuitement sur internet. Ensuite, vous pouvez écrire vos tests unitaires en utilisant les fonctionnalités intégrées et les exécuter directement dans Visual Studio 2012. Vous pouvez ainsi obtenir les résultats des tests et d'autres informations comme l'analyse de la couverture du code [Fig.5].

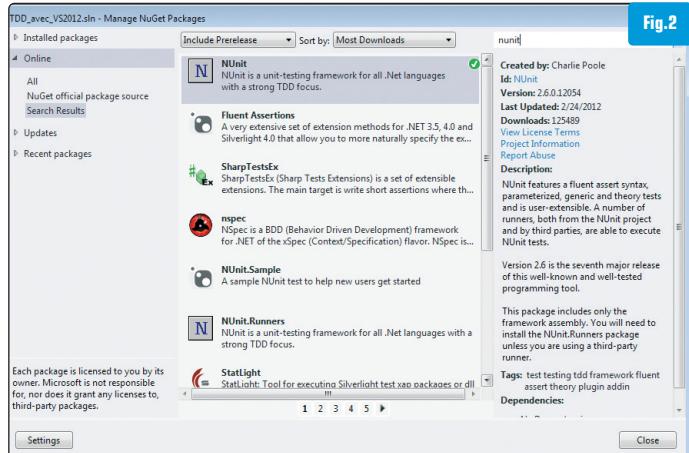
Le support des tests unitaires pour les programmes en C++ est également une nouveauté non négligeable [Fig.6].

Les développeurs C++ pourront ainsi se mettre au TDD en utilisant directement MSTest Native. Voici un exemple d'implémentation des tests unitaires en utilisant C++ et MSTest Native :

```
#include <stdafx.h>
#include <CppUnitTest.h>
#include <Calculator.h>

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace NativeUnitTests
{
    TEST_CLASS(UnitTests)
    {
        public:
            TEST_METHOD(Calculator_AdditionTest)
            {
                int value1, value2, expectedResult, actualResult;
                Calculator::Calculator calculator;
                a=1, b=2;
                expectedResult = 3;
                actualResult = calculator.Addition(value1, value2);
                Assert::AreEqual(expectedResult, actualResult);
            }
    };
}
```



Support de « async » et « await » dans les tests

Avec l'arrivée du framework .Net 4.5 et surtout de Windows 8, il est évident que la programmation asynchrone est l'une des nouveautés les plus importantes et incontournables ; Visual Studio 2012 et son framework de tests unitaires supportent parfaitement cette nouvelle approche. Les mots clé `async` et `await` introduits par .Net 4.5 peuvent désormais être utilisés pour faire des tests unitaires sur des méthodes asynchrones. Voici un exemple d'une méthode asynchrone à tester :

```
public async Task<int> AdditionAsync(int value1, int value2)
{
    // Simulate computing time
    await Task.Delay(3000);
    return value1 + value2;
}
```

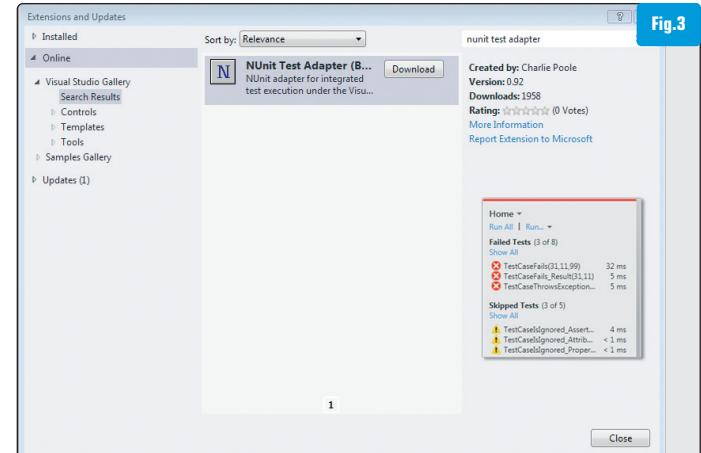
La méthode de test correspondante, implémentée en utilisant NUnit, pourrait être la suivante :

```
[Test]
public async void AdditionAsync_PositiveValues()
{
    var calculator = new Calculator();
    var result = await calculator.AdditionAsync(1, 2);
    Assert.AreEqual(3, result);
}
```

Gestion des tests via Test Explorer

La première impression à l'ouverture de la fenêtre « Test Explorer » est très positive. Voici les principaux changements apportés :

- Les panneaux « Test View » et « Test Results » ont été supprimés et remplacés par le « Test Explorer », ceci améliore l'interaction entre le développement (le code) et les tests.
- L'interface est simple mais efficace : toutes les informations sont accessibles par simple clic de la souris de manière parfaitement intuitive.
- Les tests sont regroupés en fonction de leur statut (failed, passed,...) et les premiers tests visibles sont ceux en échec : ceux qui intéressent en priorité les développeurs. De plus, il n'y a plus besoin de passer par le panneau « Test Results » pour accéder aux sources du test, un simple double clic permet d'atteindre les sources.





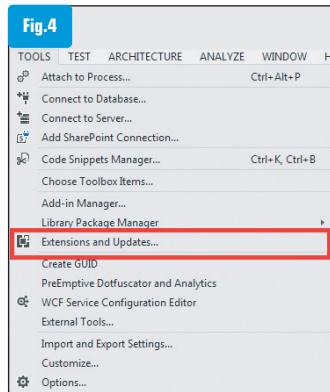
- L'exécution de l'analyse de la couverture du code est simplifiée. Dans les versions précédentes, lancer une couverture de test était certes simple mais pas très intuitif : vous deviez créer un fichier de configuration, le configurer, le lancer depuis le menu Visual Studio et ouvrir la fenêtre adéquate pour obtenir le résultat. Avec Visual Studio 2012, tout cela est plus facile car tout est intégré dans l'interface du « Test Explorer ». Le lancement d'une analyse de la couverture du code se fait directement avec la souris.

Post Build Test Runs

Une bonne habitude à adopter lors de l'application du TDD est d'exécuter les tests unitaires le plus souvent possible pour s'apercevoir au plus tôt d'un dysfonctionnement ou d'une éventuelle régression. Il existe dans la nouvelle version de Visual Studio 2012 la possibilité d'activer l'exécution des tests unitaires après chaque compilation. Ceci est disponible dans le menu « Test/Test Settings » [Fig.7], mais aussi directement depuis le « Test Explorer ». De plus, les tests unitaires tournent sur un Thread d'arrière-plan, la productivité des développeurs n'étant donc pas impactée.

Compatibilité ascendante entre Visual Studio 2010 et Visual Studio 2012

Ceux qui, comme nous, ont déjà effectué la migration des tests unitaires de Visual Studio 2008 vers Visual Studio 2010, ont pu rencontrer quelques difficultés et quelques bugs car la migration n'était pas transparente et parfois complexe. Cette opération était fort pénible, dans certains cas, la migration des applications devait même être forcée vers le framework .NET 4.0 ! Rassurez-vous, avec Visual Studio 2012, vous n'aurez aucun problème de ce type, car les composants utilisés sont les mêmes que ceux utilisés par Visual Studio 2010. En effet, le composant actuel est encore « Microsoft.VisualStudio.QualityTools.UnitTesting.dll » dans la version 10.0.0.0 et utilise toujours le runtime .NET v2.0.50727.

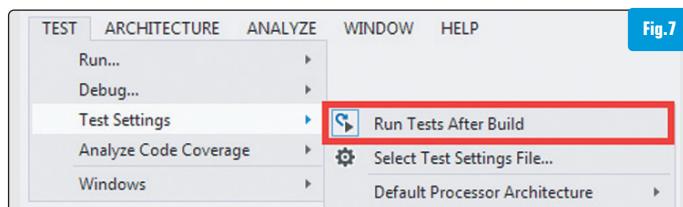


Fakes Framework (Stubs et Shims)

Visual Studio 2012 a également intégré le « Fakes Framework » issu du projet « Moles » créé par l'équipe Microsoft Research. Ce framework sera disponible uniquement avec la version Ultimata de Visual Studio 2012.

Le but de cet outil est de per-

Code Coverage Results				
Hierarchy	Not Covered (Blocks)	Not Covered (% Blocks)	Covered (Blocks)	Covered (% Blocks)
Jason_JASON-PC 2012-06-05 23_19_52.cover	2	16.67 %	10	83.33 %
nunittests.dll	2	25.00 %	6	75.00 %
tdd_avec_vs2012.dll	0	0.00 %	4	100.00 %
{1} TDD_avec_VS2012	0	0.00 %	4	100.00 %
Calculator	0	0.00 %	4	100.00 %
Addition(int, int)	0	0.00 %	2	100.00 %
Multiplication(int,...	0	0.00 %	2	100.00 %



mettre aux équipes de développement de produire rapidement et facilement des tests unitaires. Pour cela, le « Fakes Framework » introduit deux notions :

- Les Stubs : ce sont des implémentations d'interfaces ou de classes abstraites automatiques pouvant être utilisées par les tests afin d'isoler la partie à tester unitairement.
- Les Shims : ce sont des mécanismes qui interceptent des appels de méthodes au run-time et les remplacent par d'autres. Les Shims peuvent être utilisés pour isoler les appels vers des méthodes contenues dans des objets qui ne peuvent normalement pas être « mockés ». Par exemple, il est impossible de tester unitairement des méthodes faisant appel à certains objets du framework .NET. Grâce aux Shims, il est possible de rediriger les appels vers ses propres implémentations.

Les classes du framework .NET incluses dans les namespaces « mscorelib » et « system » ne peuvent pas avoir de « Fake Assembly ». On ne pourra donc malheureusement pas créer de Shims pour la classe « System.Configuration.ConfigurationManager » par exemple. Le « Fakes Framework » apporte un réel plus par rapport à des framework de tests existants (comme RhinoMock) qui poussent souvent les développeurs à modifier leur code fonctionnel pour pouvoir effectuer des tests unitaires.

Test Driven Development avec Visual Studio 2012

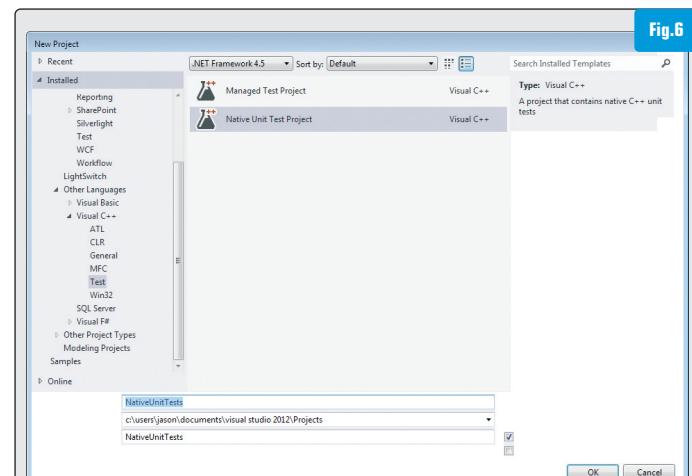
Prenons un exemple concret dans Visual Studio 2012 pour implémenter des fonctionnalités d'un calculateur.

Phase 1 : Ecriture du premier test pour une nouvelle fonctionnalité

En respectant le Test Driven Development, il faut d'abord créer les tests unitaires pour les méthodes « Addition » et « Multiplication ». Pour cela, il faut ajouter un nouveau projet de type « Unit Test Project ». Voici un exemple d'implémentation des tests unitaires en utilisant NUNIT :

```
using System;
using NUnit.Framework;
using TDD_avec_VS2012;

namespace NUnitTests
{
    [TestFixture]
    public class UnitTests
```



```

{
    [Test]
    public void Calculator_AdditionTest()
    {
        var calculator = new Calculator();
        Assert.AreEqual(calculator.Addition(2, 3), 5);
    }

    [Test]
    public void Calculator_MultiplicationTest()
    {
        var calculator = new Calculator();
        Assert.AreEqual(calculator.Multiplication(2, 3), 6);
    }
}

```

A noter que la définition de la classe « Calculator » et de ses méthodes « Addition » et « Multiplication » n'existent pas encore à ce stade. Visual Studio 2012 permet la génération du code manquant de manière automatique. Pour cela, il faut cliquer droit sur la définition « new Calculator » dans le projet de tests unitaires et choisir de générer la classe via l'option « Generate/New Type » [Fig.8]. Un assistant s'ouvre permettant de configurer le type (classe, struct, interface, enum), l'accès (public, internal), le projet de destination et le nom du fichier pour la création de la classe manquante.

L'étape suivante consiste donc à générer automatiquement les méthodes manquantes au sein de la nouvelle classe « Calculator ». Ceci se fait quasiment de la même manière, en cliquant droit sur les appels « calculator.Addition(...) » et « calculator.Multiplication(...) » dans le projet de tests unitaires et en choisissant « Generate/Method Stub ». Voici ce qui est généré :

```

public object Addition(int p1, int p2)
{
    throw new NotImplementedException();
}

public object Multiplication(int p1, int p2)
{
    throw new NotImplementedException();
}

```

La dernière étape de cette phase comporte le lancement de « Test Explorer » et l'exécution de tous les tests unitaires via « Run All » (ou utiliser l'option d'exécution des tests unitaires après compilation). Comme attendu, les tests échouent car l'implémentation du code n'a pas encore été faite.

Phase 2 : Implémentation du code minimal nécessaire pour passer le test

Il ne reste plus qu'à écrire le code qui implémente les fonctionnalités attendues. L'idée est de programmer le code le plus simple répon-

dant aux besoins. L'optimisation et l'amélioration interviennent dans un deuxième temps dans la phase suivante (la refactorisation).

```

public int Addition(int value1, int value2)
{
    return value1 + value2;
}

public int Multiplication(int value1, int value2)
{
    return value1 * value2;
}

```

Suite à l'implémentation, il faut alors relancer « Test Explorer » et exécuter tous les tests unitaires via « Run All » (ou utiliser l'option d'exécution des tests unitaires après compilation) et valider que les tests passent avec succès.

A ce stade-là, le code répond aux besoins attendus. La structure finale inclut un projet avec l'implémentation et un autre avec tous les tests unitaires. Par contre, le code n'est peut-être pas optimisé et sa qualité peut laisser à désirer. Il va falloir donc l'améliorer. Les tests unitaires servent dans ce cas comme filet de sauvetage : la refactorisation du code peut se faire sans avoir peur des régressions, car celles-ci seront détectées par les tests unitaires.

Phase 3 : Refactorisation et optimisation du code

La refactorisation décrit le processus d'amélioration de code après son écriture en modifiant sa structure interne sans modifier son comportement extérieur. On transforme un code qui fonctionne en un code qui fonctionne de manière optimale. Souvent, il en résulte un code plus rapide, utilisant moins de mémoire ou simplement présentant une implémentation plus élégante. Cela consiste à :

- Déetecter et éliminer toute duplication de code
- Limiter la complexité et le nombre de classes
- Simplifier et optimiser l'algorithmique des méthodes
- Relocaliser, renommer et harmoniser les méthodes
- Améliorer la lisibilité du code
- Supprimer le code non utilisé (dit code mort)
- Ajouter des commentaires sur des sections complexes

Dans l'exemple, il n'y a rien à refactoriser, car il n'y a pas encore beaucoup de méthodes ni de classes implémentées. Cette dernière étape doit quand même être réalisée en fin de chaque itération du cycle. Le cycle incluant d'autres fonctionnalités recommence ensuite depuis la phase 1.

Conclusion

Visual Studio 2012 apporte un grand nombre d'améliorations autour des tests unitaires, aussi bien au niveau du framework que des outils. De plus, la flexibilité de pouvoir utiliser son framework de tests unitaires de préférence est un grand avantage. L'association entre TDD et Visual Studio 2012 se fait très facilement. N'hésitez donc pas à l'utiliser avec vos projets agiles !



Jason DeOliveira



Fathi Bellahcene

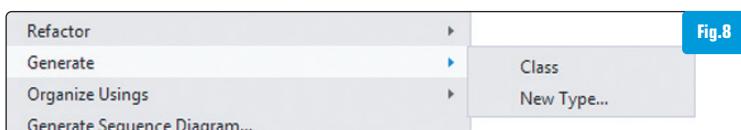


Fig.8

Je débute avec... OpenShift

2^e partie

Dans le précédent article, nous présentions l'offre PAAS de Red Hat, OpenShift, et proposions la création d'une première application de démonstration en PHP. Mais la plateforme propose d'autres technologies et nous nous intéressons aujourd'hui plus particulièrement aux outils mis à disposition des développeurs Java.

Cet article présente dans un premier temps les choix techniques retenus par Red Hat et propose dans un second temps de guider l'utilisateur à mettre en place les outils adéquats pour bien démarrer.

PRÉSENTATION TECHNIQUE DE LA PLATEFORME JAVA

> Le cœur du système

Le serveur d'applications est l'élément central d'une architecture web, la plateforme Java proposée par OpenShift s'appuie naturellement sur JBoss AS 7.1 de Red Hat. Cette version respecte la spécification JEE 6 Web Profile mettant à disposition les principaux standards tels que JSF 2, Bean Validation, CDI avec Weld, JPA, etc.

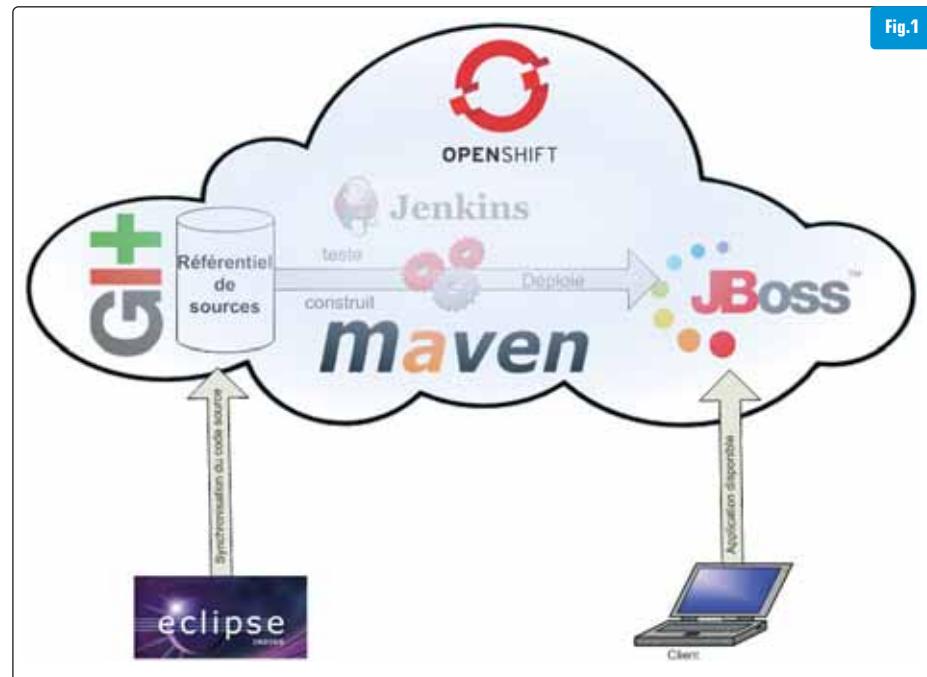
Une des forces de cet opus est de s'appuyer sur OSGi afin de disposer d'un serveur d'applications modulable et performant étant donné qu'il ne charge que ce dont il a réellement besoin.

> Les outils annexes

Il est envisageable de déployer directement son application dans l'instance du JBoss fourni mais OpenShift propose un certain nombre d'outils bien connus des développeurs Java permettant de faciliter la création, le développement, le déploiement et l'intégration continue d'une application JEE. Tout d'abord, il existe un plug-in spécifique OpenShift pour l'outil de développement Eclipse afin d'être en mesure de créer un projet et de réaliser les développements de son application.

Maven est pris en charge, en paramétrant correctement le fichier pom.xml à la racine du projet, afin de simplement pouvoir définir les dépendances et packager son application. Ainsi, il est possible de configurer les librairies nécessaires à son application ainsi que les modules du serveur d'applications à charger.

La mise en oeuvre d'une plateforme continue est possible grâce à Jenkins qu'il est pos-



sible d'activer afin de réaliser des tests avant de déployer automatiquement l'application après un commit dans le référentiel de source.

De ce fait, juste en synchronisant les sources depuis votre environnement de développement avec votre référentiel GIT d'OpenShift, votre application sera construite, testée et déployée ! [Fig.1].

BIEN DÉBUTER LE DÉVELOPPEMENT JAVA EE POUR OPENSHIFT AVEC ECLIPSE

Nous présentons dans cette section les différentes étapes de création d'une application JEE pour OpenShift, à l'aide d'Eclipse et de son plugin OpenShift. Le choix de l'IDE Eclipse est ici lié à l'existence d'un plugin simplifiant la gestion des applications OpenShift, mais l'utilisation de tout autre outil est bien évidemment possible. Pour l'utilisation des outils en ligne de commandes (ssh, rhc et

git), on peut se référer au précédent article, ou encore à la documentation de Red Hat en ligne [1].

> Installation des outils nécessaires

L'installation d'Eclipse est simple et rapide. Elle consiste à récupérer l'archive de la version souhaitée sur le site de l'éditeur [2] – la version « Eclipse IDE for Java EE Developers » dans le cadre de cet article – puis à décompresser le contenu à l'endroit souhaité. Un répertoire Eclipse est alors créé qui contient l'exécutable permettant le lancement de l'IDE. A la première exécution, il est demandé de choisir le répertoire de travail. Nous validons ici le choix par défaut, Eclipse est maintenant prêt à être utilisé.

Pour l'installation du plugin pour OpenShift, on utilise l'interface de gestion des plugins intégrée à Eclipse et accessible depuis le menu le menu « Help → Install New Software ».

Dans la fenêtre qui s'ouvre, il faut ajouter le

[1] Documentation OpenShift : http://docs.redhat.com/docs/en-US/OpenShift/2.0/html/Getting_Started_Guide/

dépôt des outils JBoss, qui contient entre autres, le plugin qui nous intéresse, en cliquant sur le bouton « Add... » [Fig.2].

On choisit un nom pour le nouveau dépôt, par exemple « JBoss Tools », puis on en précise l'adresse, à savoir « <http://download.jboss.org/jbosstools/updates/development/indigo/> » [Fig.3].

Après validation, de nouveaux éléments sont disponibles. On étend le noeud « JBoss Cloud Development Tools » et on sélectionne l'élément « JBossOpenShift Tools » [Fig.4].

Après installation du plugin, il faut redémarrer Eclipse. L'IDE est alors prêt à gérer les projets OpenShift.

> Préparation et configuration

Pour pouvoir créer des applications OpenShift via le plugin Eclipse, il faut disposer d'un compte OpenShift valide. Avoir installé les outils en ligne de commande peut également être un plus. Pour la création d'un compte ou la configuration des outils, il est possible de se référer au précédent article, ou de s'aider de la documentation en ligne.

Il est également nécessaire de disposer d'un accès SSH. Si le système n'est pas encore configuré pour communiquer avec les serveurs OpenShift via SSH, comme cela a été configuré dans le précédent article, il est

possible de générer une paire de clés depuis l'interface d'Eclipse. Pour cela, on accède aux préférences d'Eclipse via « Eclipse → Préférences ... », puis dans « General → Network Connections → SSH2 » on utilise l'onglet « Key Management » qui nous permet de générer une nouvelle paire de clés [Fig.5]. Il ne reste alors qu'à ajouter la clé publique générée au trousseau d'OpenShift, via la console d'administration Web.

Si une paire de clés a déjà été générée, et que la clé publique est déjà dans le trousseau d'OpenShift, il faut vérifier que la clé privée correspondante est bien listée dans l'onglet General.

Le système est maintenant prêt pour se lancer dans la réalisation d'une première application.

> Création et modification de l'application

On procède à la création d'un projet de type « OpenShift application », accessible via le menu « File → New → Other ... » [Fig.6].

Sont alors demandés les login et mot de passe d'accès au compte OpenShift. Le choix est donné entre les différents types d'applications disponibles pour la nouvelle application OpenShift. C'est ici le choix « jbossas-7 » que nous faisons [Fig.7].

Remarque : on peut noter que si l'application existe déjà sur les serveurs d'OpenShift, il est possible de la sélectionner ici. On peut également sélectionner à cette étape des Cartridges (modules d'une application OpenShift) si nécessaire, comme un serveur de base de données notamment.

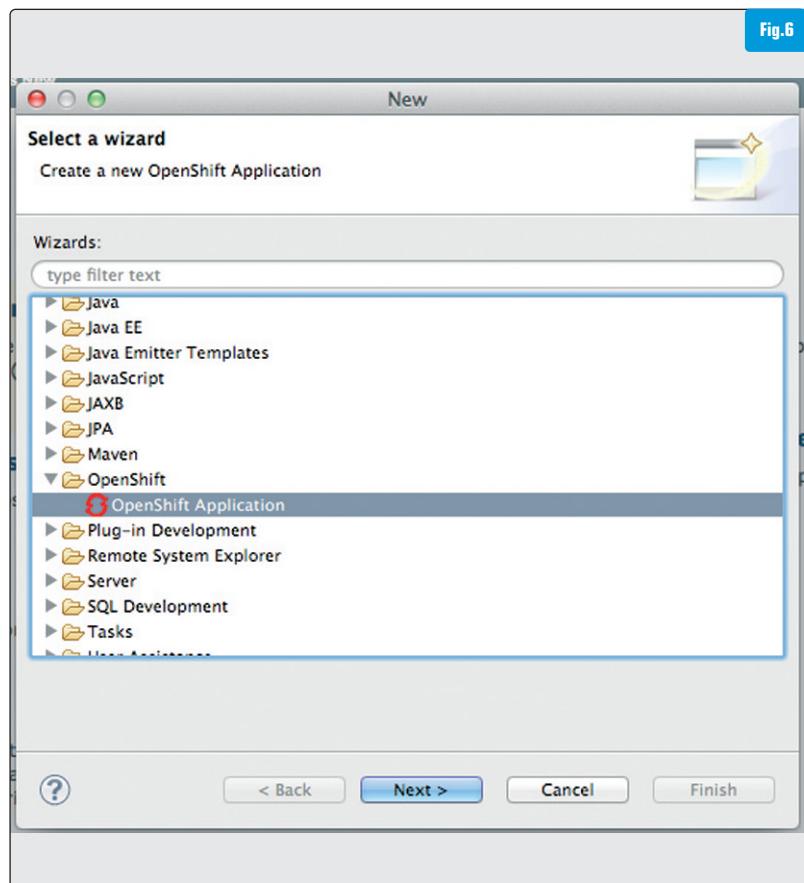
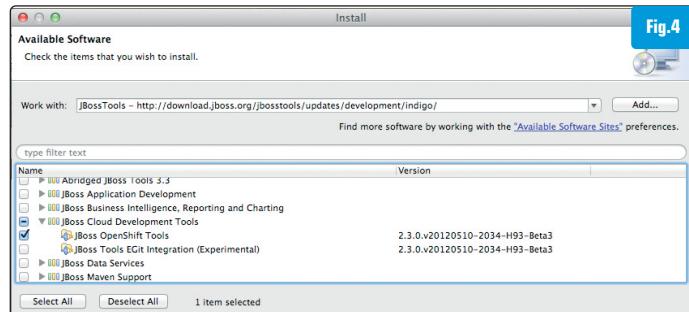
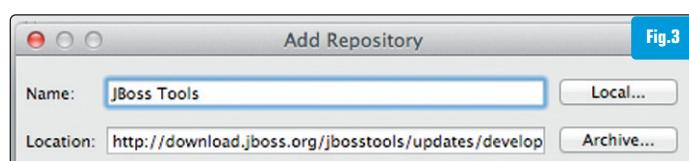
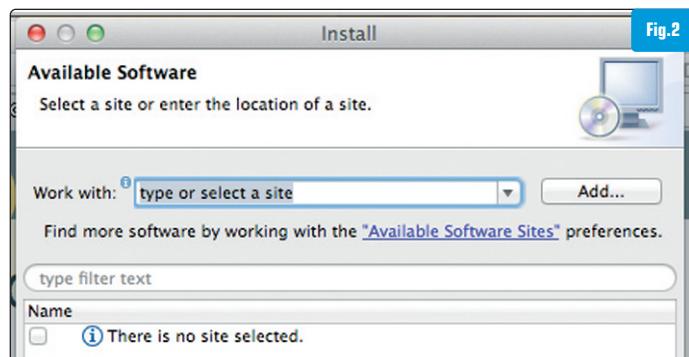
Après validation, la création de l'application peut prendre quelques minutes. En effet, l'application est créée et configurée sur les serveurs d'OpenShift, avec notamment la création du dépôt Git indispensable à la mise à jour de l'application. Le dépôt est automatiquement cloné localement et le projet Eclipse est créé et ouvert dans l'IDE.

Une fois créée, l'application est accessible en ligne [Fig.8]. Pour rappel, les applications OpenShift sont accessibles via une url de la forme : <http://nomapplication-nomdomaine.rhcloud.com>

Pour le développement de l'application, il n'y a rien à préciser, il s'effectue comme à l'accoutumée sous Eclipse.

Une fois les sources modifiées, la mise à jour du dépôt Git se fait simplement par un clic droit sur le projet dans l'explorateur de projet suivi d'un clic sur « Team → Commit » pour mettre à jour le dépôt local, et « Team → Push to Upstream » pour mettre à jour le dépôt distant OpenShift.

La mise à jour du dépôt OpenShift prend un peu de temps. En effet, après la simple mise





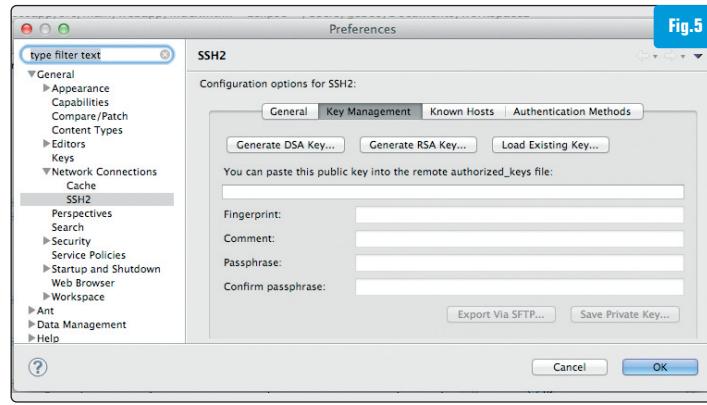
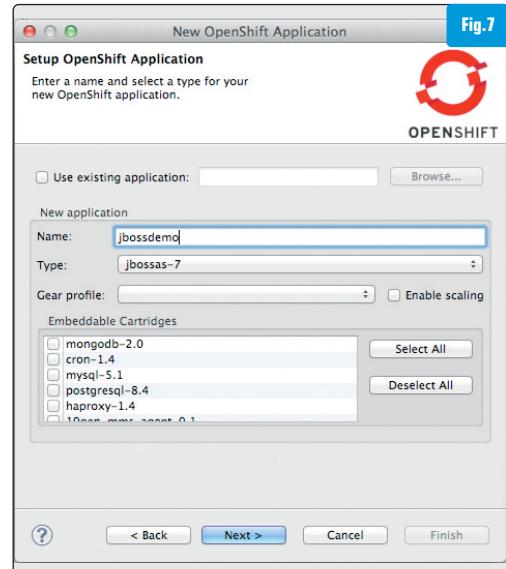
à jour du code source, une compilation et un redéploiement de l'application sont effectués. Il est donc possible de développer localement et de tester les modifications « dans le cloud » en utilisant les fonctionnalités de compilation d'OpenShift. Après le redéploiement, la nouvelle version de l'application est accessible en ligne.

En suivant les quelques étapes présentées ici, nous sommes en mesure d'installer simplement un environnement de développement complet et opérationnel pour travailler efficacement avec OpenShift, tout en tirant profit des différentes fonctionnalités offertes. Le recours à la ligne de commande n'est pas obligatoire et le développeur habitué à Eclipse n'a pas à quitter son IDE favori.

CONCLUSION

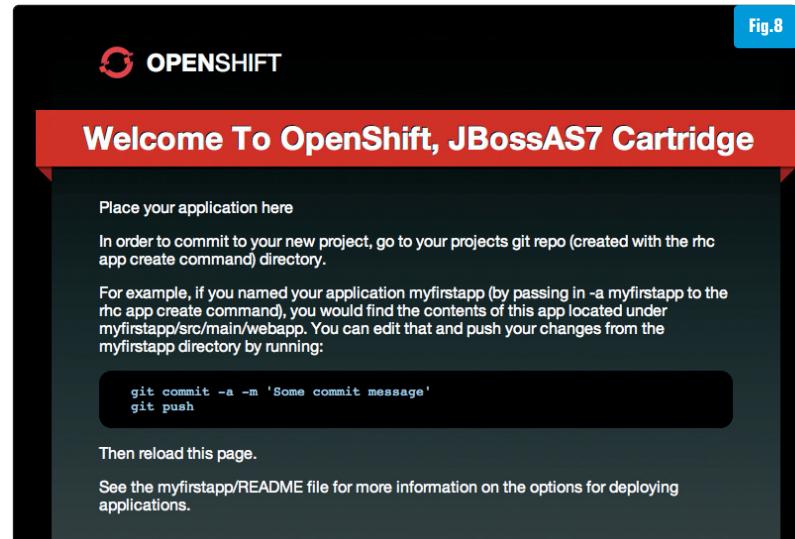
Nous nous sommes intéressés dans cet article aux possibilités offertes par OpenShift pour la mise en place d'applications Web Java. L'offre est complète, cohérente et simple à appréhender. L'existence d'un plugin Eclipse est un réel plus, il permet aux développeurs de s'affranchir de la ligne de commande pour la plupart des tâches habituelles tout en gardant leurs habitudes de développement. Le service est actuellement en phase de test et est donc disponible gratuitement avec quelques limitations, notamment en termes de mémoire et de nombre d'applications.

Gilles Bougenière et Jean-Sébastien Mercy, Consultants Osaxis
 (<http://www.osaxis.fr>)



(2) Téléchargements Eclipse : <http://www.eclipse.org/downloads/>

(3) Console d'administration d'OpenShift : <https://openshift.redhat.com/app/account>



L'INFO permanente

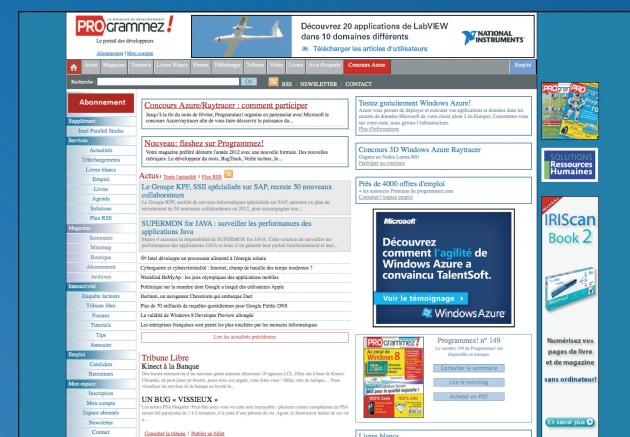
- **L'actu** : le fil d'info quotidien de la rédaction
- **La newsletter hebdo** : abonnez-vous, comme 38 000 professionnels. **C'est gratuit !**

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

LE MAGAZINE DU DÉVELOPPEMENT
Programmez!
www.programmez.com



Créer son service de stockage en ligne disponible 24/24h

Il est très facile de créer un service de stockage de type « DropBox » qui permet à chacun d'emporter partout ses documents, photos et vidéos et de les partager facilement de façon sécurisée. L'accès à vos données est sécurisé par un certificat SSL. Celles-ci transitent de manière cryptée et sont donc illisibles sur Internet ; de plus, les données entreposées sur le Cloud sont cryptées par vos soins. Il est à noter que nous mettrons en place une solution de synchronisation transparente de vos données entre votre PC et la solution Cloud de stockage.



Crédit Photo: D.R.

Pour mettre en place rapidement cette solution, nous nous appuierons sur la solution Cloud de Amazon Simple Storage Service (S3) de Amazon Web Service (AWS) qui offre un service de base sécurisé et hautement disponible. Nous aurons ainsi des données stockées jusqu'à 99,999999999 % de durabilité, et 99,99 % de disponibilité.

Cette solution nécessite l'ouverture d'un compte sur Amazon Web Service. De nombreuses solutions de stockage en ligne du marché s'appuient sur S3. Vous pouvez créer votre compte AWS via ce lien <https://aws.amazon.com/fr/> et par la même occasion, récupérer vos identifiants pour utiliser le service S3.

AWS est une véritable offre Cloud. Elle permet donc d'accéder à tout ces services via des requêtes REST ou SOAP et offre de nombreuses API de plus haut niveau pour y accéder. Dans notre exemple nous allons utiliser AWS SDK for .NET pour implémenter notre solution de stockage.

Pour rester le plus simple possible, nous allons commencer par créer une application de type Console dans Visual Studio 2010 et ajouter via NuGet (<http://nuget.org/>) les dépendances pour utiliser les Web Services Amazon.

Commençons par initialiser notre client Amazon S3 avec les identifiants récupérés précédemment :

```
amazonS3Client amazonS3Client = new AmazonS3Client("AWSAccess
Key", "AWSSecretKey");
```

Ensuite nous allons créer notre Bucket, qui correspond à un emplacement de stockage sur S3, vous pouvez l'imaginer comme votre répertoire racine. Il faut savoir que le nom de ce Bucket doit être unique pour tous les utilisateurs.

Nous allons donc créer notre Bucket, mais avant, nous allons vérifier s'il existe déjà parmi nos Buckets

```
ListBucketsResponse listBucketsResponse = amazonS3Client.List
Buckets();
bool bucketIsExist = listBucketsResponse.Buckets.Any(s3Bucket
=> s3Bucket.BucketName == BucketName);

if (!bucketIsExist)
{
    amazonS3Client.PutBucket(new PutBucketRequest().WithBucket
Name("ValtechProgrammez"));
}
```

Comme nous pouvons le voir, nous récupérons facilement la liste de nos Buckets via le client Amazon S3, et s'il n'existe pas, nous le créons via une requête de type PUT (S3 expose un Web Service de type REST). Si un Bucket existe déjà sous ce nom, à vous ou pas, une erreur sera signalée. Il faudra choisir un nouveau nom pour pouvoir passer à l'étape suivante.

Le contrôle de l'accès au partage de l'espace de stockage entre amis se fera via la console de configuration du Service S3. Il est possible de partager son compte AWS avec de nombreux utilisateurs choisis. Pour plus d'informations sur les diverses politiques de contrôle d'accès disponibles dans Amazon S3, il faut se référer au sujet Contrôle d'accès dans le Guide du développeur Amazon S3.

Afin de renforcer la sécurité, nous souhaitons crypter les fichiers transmis à S3. Pour cela, nous allons utiliser la technique dite du « Téléphone Rouge » basée sur une phrase (l'algorithme de crypto Rijndael). Nous initialiserons donc notre algorithme de cryptage via

une simple phrase que nous pourrons transmettre à toutes les personnes avec qui nous partagerons notre répertoire afin de décrypter automatiquement les données stockées sur le Cloud :

```
Rfc2898DeriveBytes rfcDb = new Rfc2898DeriveBytes(password, System.
Text.Encoding.UTF8.GetBytes(password));

Rijndael rijndael = Rijndael.Create();
byte[] key = rfcDb.GetBytes(32); //256 bits key
byte[] iv = rfcDb.GetBytes(16); // 128 bits key
_aesEncryptor = rijndael.CreateEncryptor(key, iv);
_aesDecryptor = rijndael.CreateDecryptor(key, iv);
```

Nous allons utiliser le cryptage dans les deux méthodes ci-dessous qui permettent respectivement de télécharger et « d'uploader » un fichier :

```
private static void UploadFile(FileInfo fileInfo)
{
    System.Console.WriteLine("Uploading " + fileInfo.Name);
    using (FileStream inputFileStream = new FileStream(fileInfo.
FullName, FileMode.Open))
        using (MemoryStream outputMemoryStream = new MemoryStream())
            using (CryptoStream cryptoStream = new CryptoStream(output
MemoryStream, _aesEncryptor, CryptoStreamMode.Write))
                {
                    int data;
                    while ((data = inputFileStream.ReadByte()) != -1)
                    {
                        cryptoStream.WriteByte((byte)data);
                    }
                    cryptoStream.FlushFinalBlock();
                }
                PutObjectRequest createRequest = new PutObjectRequest();
                createRequest.WithMetaData("x-amz-meta-LWT", fileInfo.LastWrite
Time.ToString("G"));
                createRequest.WithBucketName(BucketName);
                createRequest.WithKey(fileInfo.Name);
                createRequest.WithInputStream(outputMemoryStream);
                _amazonS3Client.PutObject(createRequest);
            }
}

private static void DownloadFile(S3Object s3Object)
{
    System.Console.WriteLine("Downloading " + s3Object.Key);
    GetObjectResponse getObjectResponse = _amazonS3Client.GetObject
(new GetObjectRequest { BucketName = BucketName, Key = s3Object
.Key });

    string filePath = Path.Combine(_folder, s3Object.Key);
    using (BufferedStream inputBufferedStream = new BufferedStream
(getObjectResponse.ResponseStream))
        using (CryptoStream cryptoStream = new CryptoStream(input
BufferedStream, _aesDecryptor, CryptoStreamMode.Read))
            using (FileStream outputFileStream = new FileStream(filePath,
FileMode.Create, FileAccess.ReadWrite))
                {
                    int data;

```

```
                    while ((data = cryptoStream.ReadByte()) != -1)
                    {
                        outputFileStream.WriteByte((byte)data);
                    }
                }
                new FileInfo(filePath).LastWriteTime = DateTime.Parse(getObject
Response.Metadata["x-amz-meta-LWT"]);
            }
}
```

Maintenant que nous disposons d'un Bucket pour entreposer nos fichiers et de deux méthodes pour transmettre des fichiers, nous allons créer une autre méthode que nous appellerons à intervalle régulier pour synchroniser notre répertoire avec S3. Nous utilisons un style d'architecture de type PULL, chaque PC, mobile, tablette demandent à intervalle régulier s'il y a de nouvelles données, des changements sur notre entrepôt de données Cloud.

Pour vérifier les modifications sur les fichiers, nous pouvons nous baser sur la date de dernière écriture. Amazon S3 nous permet de rajouter des métadonnées sur chaque fichier, nous allons utiliser cette fonctionnalité pour stocker la date de modification, malheureusement, il faut faire une requête spécifique pour les récupérer.

```
private static void GetFilesOnS3(object state)
{
    System.Console.WriteLine("Check Files on S3...");
    ListObjectsResponse listObjectsResponse = _amazonS3Client.List
Objects(new ListObjectsRequest { BucketName = BucketName });
    foreach (S3Object s3Object in listObjectsResponse.S3Objects)
    {
        FileInfo fileInfo = new FileInfo(Path.Combine(_folder, s3
Object.Key));
        if (fileInfo.Exists)
        {
            System.Console.WriteLine(s3Object.Key + " exist");
            GetObjectMetadataRequest getObjectMetadataRequest = new Get
ObjectMetadataRequest();
            getObjectMetadataRequest.WithBucketName(BucketName).WithKey
(s3Object.Key);
            GetObjectMetadataResponse getObjectMetadataResponse = _amazon
S3Client.GetObjectMetadata(getObjectMetadataRequest);
            //int dateCompare = DateTime.Compare(fileInfo.LastWriteTime.,
DateTime.Parse(getObjectMetadataResponse.Metadata["x-amz-meta-LWT"]));
            int dateCompare = (int)(fileInfo.LastWriteTime - DateTime.Parse
(getObjectMetadataResponse.Metadata["x-amz-meta-LWT"])).Total
Seconds;
            if (dateCompare == 0)
            {
                System.Console.WriteLine("No difference, do nothing");
            }
            else if (dateCompare < 0)
            {
                System.Console.WriteLine("S3 is newer");
                DownloadFile(s3Object);
            }
            else if (dateCompare > 0)
            {
                System.Console.WriteLine("FS is newer");
                UploadFile(fileInfo);
            }
        }
    }
}
```

```
        }
    }
}

else
{
    System.Console.WriteLine(s3Object.Key + " doesn't exist");
    DownloadFile(s3Object);
}

}

foreach (string file in Directory.GetFiles(_folder))
{
    FileInfo fileInfo = new FileInfo(file);
    if (!listObjectsResponse.S3Objects.Any(s => s.Key == fileInfo.Name))
    {
        UploadFile(fileInfo);
    }
}
}
```

Pour gérer efficacement les suppressions et les « renommages », nous allons faire appel au `FileSystemWatcher` fourni par le Framework .NET pour surveiller notre répertoire :

```
private static void InitializeFileSystemWatcher()
{
    FileSystemWatcher fileSystemWatcher = new FileSystemWatcher(_folder)
    {
        IncludeSubdirectories = false,
        NotifyFilter = NotifyFilters.FileName
    };

    fileSystemWatcher.Deleted += fileSystemWatcher_Changed;
    fileSystemWatcher.Renamed += fileSystemWatcher_Renamed;

    // Begin watch
    fileSystemWatcher.EnableRaisingEvents = true;
}

private static void fileSystemWatcher_Renamed(object sender,
RenamedEventArgs e)
{
    System.Console.WriteLine("Rename of " + e.OldName + " in " + e.Name);
    RenameS3Object(e.OldName, e.Name);
}

private static void fileSystemWatcher_Changed(object sender, FileSystemEventArgs e)
{
    switch (e.ChangeType)
    {
        case WatcherChangeTypes.Deleted:
            System.Console.WriteLine("Deletion of " + e.Name);
            DeleteS3Object(e.Name);
            break;

        default:
            System.Console.WriteLine("Bug " + e.Name);
    }
}
```



```

        break;
    }
}

private static void DeleteS3Object(string key)
{
    DeleteObjectRequest deleteRequest = new DeleteObjectRequest
    {
        BucketName = BucketName,
        Key = key
    };
    _amazonS3Client.DeleteObject(deleteRequest);
}

private static void RenameS3Object(string oldKey, string newKey)
{
    // Copy
    CopyObjectRequest request = new CopyObjectRequest
    {
        SourceBucket = BucketName,
        SourceKey = oldKey,
        DestinationBucket = BucketName,
        DestinationKey = newKey
    };
    _amazonS3Client.CopyObject(request);

    // Delete
    DeleteS3Object(oldKey);
}
}

```

Pour retrouver l'ensemble des sources exposées dans cet article, il suffit d'aller sur GitHub à l'adresse suivante : <https://github.com/kakaruto/S3Sync>

Nous voyons à travers cet exemple qu'il est simple et rapidement possible de développer un service de stockage de type « DropBox » disponible partout et hautement sécurisé. Il ne vous reste plus qu'à faire une belle IHM et le tour est joué.

Philippe Garry & Hervé Desaunois
Seniors Consultants
Valtech Toulouse

Renderscript, optimisation CPU/GPU pour Android

Google fournit un outil permettant de se rapprocher des performances du code natif, sans pour autant sacrifier les avantages du SDK : la simplicité d'utilisation et, surtout, la portabilité. Cet outil se nomme Renderscript. Il a été conçu pour développer des éléments graphiques gourmands en ressources, comme les LiveWallpaper ou les carrousels 3D de certaines applications, en s'affranchissant de l'architecture matérielle. Mais il est aussi l'outil idéal dès que l'on a besoin de performances, notamment pour les calculs.

Les téléphones Android actuels possèdent de plus en plus de coeurs et les architectures des processeurs se diversifient. Par exemple, le Galaxy S3 de Samsung possède 4 coeurs et Intel sort un téléphone équipé d'un processeur x86. Cette diversification, et non fragmentation, complique le travail du développeur quand on en vient à l'optimisation.

Le SDK permet de développer des applications en utilisant des API en Java pour la Machine Dalvik. Le code est alors exécutable sur l'ensemble des plateformes qui possèdent un portage de cette machine (ARM, x86...). Le principal problème de l'ensemble Java, Dalvik et Garbage Collector est de ne pas pouvoir assurer des performances proches du code machine. Une solution existe pour interfaçer du code C/C++ avec du code Java, avec le Native Development Kit (NDK) et Java Native Interface (JNI). Seulement cette approche demande d'écrire beaucoup de code et ce qui est gagné en performance, est perdu en portabilité. Effectivement, le code du NDK doit être compilé en binaire et ne ciblera donc qu'une seule architecture. Il faudra autant de binaires ou de packages que d'architectures.

Cependant, il existe une solution développée par Google qui change la donne, elle s'appelle Renderscript.

RENDERSCRIPT, COMMENT CA MARCHE ?

Renderscript permet d'écrire des scripts de rendu et/ou de calcul. Le langage choisi est le C, dans sa version C99. Ce choix n'est pas anodin. Le C est proche de la machine et bien connu des développeurs graphiques et de la communauté scientifique. Il peut être compilé pour un grand nombre d'architectures.

Un premier compilateur, basé sur LLVM, est exécuté en amont lors du build du projet, et générera un code intermédiaire, le bytecode Renderscript. Ce bytecode est indépendant de l'architecture sur laquelle il s'exécutera.

Un second compilateur cette fois-ci sur le client transformera le bytecode en code machine (CPU, GPU...) et optimisera le code selon les instructions disponibles. Le résultat de cette compilation sera sauvegardé pour être utilisé plus tard.

Le dernier élément est le runtime Renderscript. Il se chargera de gérer ces phases de compilation ainsi que l'environnement d'exécution du script. Il se chargera aussi d'utiliser au maximum les capacités matérielles disponibles, notamment le nombre de coeurs des processeurs.

C'est toute cette mécanique qui nous permet à nous, développeurs, d'écrire du code portable, performant mais aussi multi threadé sans trop d'efforts.

RENDERSCRIPT POUR LE CALCUL DANS LA PRATIQUE

Pour le calcul, Renderscript possède deux avantages. Le premier est l'écriture de code natif avec une intégration simplifiée avec la génération de code Java. Et le second est la possibilité de faire du multithreading sur l'ensemble des coeurs du CPU sans effort.

Nous allons voir ces points au travers d'un exemple d'application. L'idée de cet exemple est de prendre un ensemble d'objets, des particules, et de calculer les interactions qui influent sur leur vitesse, trajectoires et position. Ces interactions seront par exemple, la répulsivité qu'elles peuvent générer les unes sur les autres, les rebonds sur les surfaces et la gravité. A partir d'un état d'entrée, de positions, vitesses et masses des particules, on peut calculer un état de sortie. Le but n'est pas de réaliser une simulation physique précise mais de simplement appliquer des calculs sur un ensemble d'objets. L'intégralité des lignes de code ne seront pas représentées ici. Nous montrerons l'essentiel pour mettre en place un script avec les différents mécanismes propres à Renderscript. Nous ne nous attarderons pas sur la partie rendu qui mériterait un article à elle seule.

ÉCRITURE D'UN SCRIPT RENDERSCRIPT

Tout d'abord nous allons écrire un fichier d'en-têtes particules.rsh qui contiendra la définition de la structure que nous allons utiliser, la particule.

```
typedef struct Particule {
    float2 position;
    float2 vitesse;
    float masse;
} Particule_t;
```

Notre particule est composée d'une position et d'un vecteur vitesse ainsi que d'une masse. Cette structure générera par réflexion une classe Java que l'on pourra utiliser pour initialiser notre script.

Nous allons maintenant écrire le script en lui-même dans le fichier particules.rs. Nous allons commencer par les en-têtes, obligatoires pour tout fichier Renderscript, ainsi que les includes nécessaires.

```
#pragma version(1)
#pragma rs java_package_name(com.genymobile.renderscript.particule)

#include <particules.rsh>
#include <rs_graphics.rsh>
```

Il est nécessaire de préciser la version utilisée de Renderscript. Actuellement, il n'existe qu'une seule version, la 1. Il faut aussi spécifier un nom de package qui sera celui utilisé pour les classes générées. Le premier include est notre header, défini plus haut. Le second rs_graphics.rsh est important, il est nécessaire pour les scripts de rendu. Dans notre cas, nous en avons besoin pour dessiner les particules mais nous ne détaillerons pas cette partie. Si votre script ne fait que du calcul, ce header n'est pas nécessaire.

```
// Allocations de particules
Particule_t* particules1;
Particule_t* particules2;
// Pointeurs vers les allocations d'entrée et de sortie
Particule_t *aout;
Particule_t *ain;
// Paramètres du script
int count;
int width;
int height;
```

Nous déclarons des éléments particules1 et particules2 sur lesquels nous attacherons les allocations depuis notre code Java. Les aout et ain serviront à alterner les entrées et sorties.

```
// Méthode exécutée à l'initialisation du script
void init () {
```

Nous définissons maintenant la fonction init() qui est une des deux fonctions définies par Renderscript. Cette fonction est appelée à la création du script. Elle n'est pas obligatoire. Et nous n'avons rien à mettre dedans mais au moins vous êtes maintenant au courant de son existence !

```
void init_particules() {
    // Récupération du nombre de particules
    count = rsAllocationGetDimX(rsGetAllocation(particules1));
    // Initialisation des particules
    Particule_t* p = particules1;
    for(int i = 0; i < count; i++) {
        p->position.x = rsRand(width);
        p->position.y = rsRand(height);
        p->vitesse.x = rsRand(0.f, 20.f) - 10;
        p->vitesse.y = rsRand(0.f, 20.f) - 10;
        p->masse = rsRand(0.f, 1.f);
        p++;
    }
}
```

Cette fonction a pour but d'initialiser les paramètres nécessaires à l'exécution de notre script, notamment le nombre total d'éléments qui correspond à la dimension de l'allocation allouée à particules1.

Nous définissons une position, une vitesse et une masse de manière aléatoire pour chaque particule.

```
// fonctions appliquant les différents paramètres modifiant la
position des particules
void updatePosition() {
    [...]
```

Le but de cet article n'étant pas de faire une simulation physique réaliste, nous ne détaillerons pas cette partie du code. Cette fonction comporte des opérations sur des vecteurs de Float. La position est calculée depuis l'état entrant, et l'application de plusieurs forces, la gravité et une répulsion entre les particules. C'est cette fonction qui est coûteuse en ressources.

Nous allons maintenant définir root(). Cette fonction est obligatoire pour tout script Renderscript. Pour un script de rendu, cette fonction ne prend aucun paramètre et retourne un entier. Cet entier représente le nombre de millisecondes entre deux appels à cette fonction. Si la fonction prend trop de temps entre deux traitements, le runtime Renderscript la relancera dès qu'elle se terminera. Une valeur de zéro signifie qu'on ne veut l'exécuter qu'une fois. Dans notre cas, nous voulons être appelés le plus souvent possible et retournons donc la valeur 1.

```
int root() {
    // Changement des paramètres d'entrée et de sortie.
    if (totalFrame % 2) {
        aout = particules1;
        ain = particules2;
    } else {
        aout = particules2;
        ain = particules1;
    }
    totalFrame++;
    updatePosition();
    draw();
    return 1; // On désire être appelé le plus souvent possible
}
```

La première partie de la fonction échange les états d'entrée et de sortie pour que l'ancien état de sortie, à jour, devienne le nouvel état d'entrée. Nous mettons ensuite les positions à jour et nous dessinons les particules (des pixels) sur l'écran.

Comme pour la fonction updatePosition(), nous ne détaillerons pas la méthode draw(). Elle se contente de dessiner l'ensemble des particules comme un nuage de points. Notre script est donc maintenant complet il faut l'intégrer à notre code Java.

UTILISATION DU SCRIPT DEPUIS JAVA

Utilisation des classes générées

La première étape est de compiler le projet pour voir apparaître le bytecode Renderscript dans le dossier res/raw/ et de nouvelles classes java dans le répertoire gen/ du projet. Ces classes correspondent à des éléments définis dans notre script : des scripts, des structures, etc.

Le tableau page suivante présente le lien entre ce qui est défini côté Renderscript et ce qui sera reflété en Java :

Définition domaine Renderscript	Génération domaine Java
my_script.rs	ScriptC_my_script.java + /res/raw/my_script.bc
typedef struct My_Struct {...} My_Struct_t;	ScriptField_My_Struct.java
My_Struct_t* my_struct;	.bind_my_struct(ScriptField_My_Struct a);
int entier;	.set_entier(int i) / .get_entier();
void my_function();	.invoke_my_function();

Nous encapsulerons l'utilisation du script dans une classe nommée ParticuleRS. Cette classe s'occupera de toute l'initialisation du script et comportera deux méthodes importantes. Une méthode init() qui crée le script, alloue la mémoire nécessaire pour les structures et les lie au script. Un script Renderscript n'alloue pas lui-même sa mémoire. L'allocation se fait dans le code Java et les attributs sont liés. C'est aussi la meilleure manière d'échanger des données entre un script et le code java qui l'utilise. Ici nous avons une zone mémoire pour l'état d'entrée de nos particules et une pour l'état de sortie. La seconde méthode setWidthAndHeight() nous permettra de définir les dimensions de l'écran et d'initialiser la position et la vitesse des particules.

```
public void init(RenderScriptGL rs, Resources res, int particulesCount) {
    // Création du script (note: init() sera exécutée)
    mScript = new ScriptC_particules(rs, res, R.raw.particules);

    // Allocation de la mémoire pour les particules et initialisation
    mScript.bind_particules1(new ScriptField_Particule(rs, particulesCount, Allocation.USAGE_SCRIPT));
    mScript.bind_particules2(new ScriptField_Particule(rs, particulesCount, Allocation.USAGE_SCRIPT));
    // Mise en place des éléments de rendu
    setupRendering(rs, res, particulesCount);
}

public void setWidthAndHeight(int width, int height) {
    mScript.set_width(width);
    mScript.set_height(height);
    mScript.invoke_init_particules();
}
```

Lors de l'instanciation du script, nous utilisons l'id R.raw.particules. Ceci représente l'id du bytecode Renderscript qui a été généré dans le fichier res/raw/particules.bc.

UTILISATION DU SCRIPT DANS UNE RSTEXTUREVIEW

RSTextureView est un nouveau composant graphique introduit depuis Android 4.0 (API 14). C'est une View classique mais dont le rendu est géré par un script Renderscript. Nous allons créer une classe qui hérite de RSTextureView et qui utilise notre script.

```
public class ParticuleView extends RSTextureView {
    private static final int PARTICULE_COUNT = 750;
    private RenderScriptGL mRs;
    private ParticuleRS particules;
    @Override
    protected void onAttachedToWindow() {
        super.onAttachedToWindow();
        // Création du contexte RenderScript
        SurfaceConfig sc = new SurfaceConfig();
        mRs = createRenderScriptGL(sc);
```

```
// Création et initialisation de notre script
particules = new ParticuleRS();
particules.init(mRs, getResources(), PARTICULE_COUNT);
}

@Override
protected void onSizeChanged(int w, int h, int oldw, int oldh) {
    super.onSizeChanged(w, h, oldw, oldh);
    // Définition des dimensions et bind
    particules.setWidthAndHeight(w, h);
    mRs.bindRootScript(particules.getScript());
}
```

Sur le onAttachedToWindow(), nous créons un context RenderScriptGL. Ce contexte est spécifique au script de rendu car il initialise un context OpenGL pour nous. Dans le cas d'un script de calcul uniquement, il faut utiliser un simple context RenderScript. Il suffit maintenant d'ajouter cette vue dans un layout de manière à ce qu'elle prenne l'ensemble de l'écran et d'observer le résultat [Fig.1]. Si l'on regarde les logs de l'application, sur un Galaxy Nexus (Dual Core) :

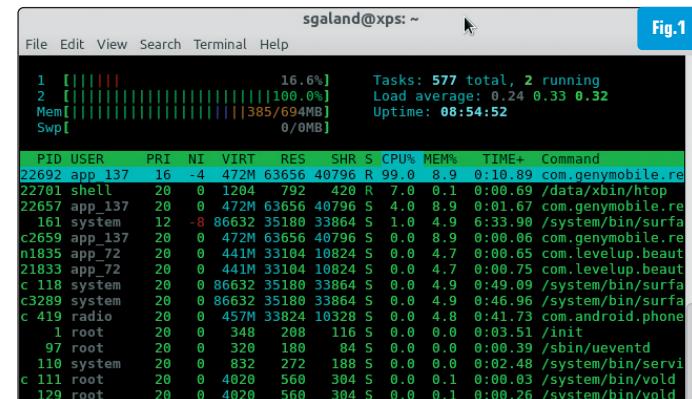
```
D/RenderScript(24958): Particules - fps: 17 0x11
D/RenderScript(24958): Particules - fps: 17 0x11
D/RenderScript(24958): Particules - fps: 17 0x11
```

On observe donc une animation un peu saccadée qui tourne à 17 images par seconde. Ceci est dû aux lourds calculs que l'on fait, notamment le mode répulsion qui calcule en *brute force* l'impact des particules les unes sur les autres. Nous avons installé le logiciel htop sur notre mobile de test pour observer en temps réel la charge du processeur. On observe qu'un seul cœur est à 100% pendant que l'autre n'affiche que 16% d'activité [Fig.1].

PARRALLÉLISATION DES CALCULS : RSFOREACH

Nous n'utilisons pas pleinement la puissance de Renderscript dans l'exemple précédent. Nous avons pu utiliser du code natif que nous avons assez simplement intégré à notre application Java. Le code est portable, il s'exécutera partout. La partie graphique tournera dans le GPU (l'affichage des points) et les calculs sur le CPU en profitant des optimisations et instructions disponibles. Mais tout se fait sur un unique cœur alors que Renderscript fournit tout ce qui est nécessaire pour distribuer les calculs sur l'ensemble des coeurs.

Pour cela nous allons sortir updatePosition() pour créer un script contenant uniquement cette fonction. Ce nouveau script sera dédié au calcul et non au rendu, il différera alors légèrement du précédent.



```
#pragma version(1)
#pragma rs java_package_name(com.genymobile.renderscript.particule)
#include <particules.rsh>

void root(const Particule_t *in, Particule_t *out, const rs_allocation *inputData) {
    [... équivalent à la fonction updatePosition() ...]
}
```

Cette fois-ci, nous n'incluons pas la partie graphique de Renderscript et effectuons le calcul dans la fonction `root()` du script. Cette fonction est différente pour les scripts de calcul, elle ne retourne rien et prend au minimum deux paramètres, un pointeur sur un élément d'entrée et un sur un élément de sortie (`in` et `out`). Le troisième paramètre est un pointeur sur ce que l'on veut, dans notre cas c'est sur l'allocation des particules de l'état d'entrée qui nous permettra de calculer la répulsion. Il ne nous reste qu'à légèrement modifier le code précédent pour intégrer la nouvelle implémentation du calcul :

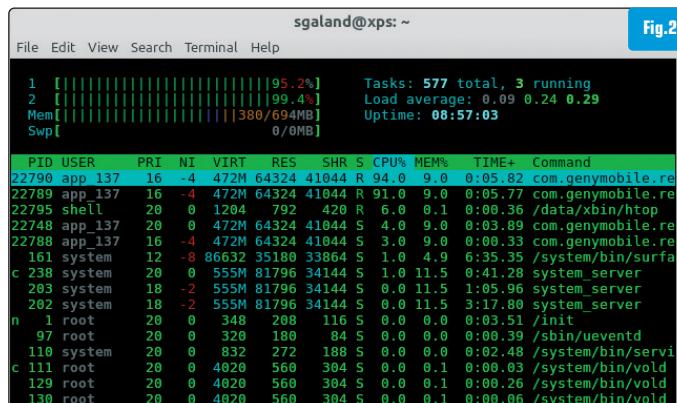
```
- particules.rs :
// Script avec gestion de la physique
rs_script physics;

int root() {
    [...]
    // Remplacement de l'appel à updatePosition()
    rs_allocation in = rsGetAllocation(ain);
    rs_allocation out = rsGetAllocation(aout);
    rsForEach(physics, in, out, &in, count);
    [...]
    return 1; // On désire être appelé le plus souvent possible
}
```

Cette dernière fonction est la plus importante, elle permet d'exécuter le traitement sur l'ensemble des éléments alloués. Elle va donc effectuer l'opération pour chaque particule et, surtout, distribuer les calculs sur l'ensemble des coeurs du processeur. La différence de performance est immédiate.

```
D/RenderScript(24958): Particules - fps: 32 0x20
D/RenderScript(24958): Particules - fps: 30 0x1e
D/RenderScript(24958): Particules - fps: 34 0x22
```

Nous avons en moyenne 32 fps, quasiment deux fois plus que précédemment. Un petit coup de `htop` nous montre que les calculs sont



bien répartis sur l'ensemble des coeurs du processeur, ici deux, chargés à 95,2% et 99,4% [Fig.2].

Cette fonction `rsForEach` peut être directement appelée depuis le code Java. Cela permet d'utiliser Renderscript uniquement pour effectuer des opérations coûteuses sur un ensemble d'objets.

POUR ALLER PLUS LOIN...

Cet article a un point de vue très théorique sur le calcul. Renderscript est utilisable dans un grand nombre de cas de figure, tout dépend de ce que vous voulez faire avec.

Nous avons ici réalisé des opérations sur des particules pour effectuer la gestion de la physique, mais on peut imaginer plein d'autres usages : calcul sur des pixels pour appliquer un filtre sur une image, intelligence artificielle, simulation de systèmes multi-agents, etc. Tous les calculs pouvant être distribués peuvent tirer parti de cette fonctionnalité.

Nous ne parlons pas ici du rendu graphique de Renderscript. Google a développé la partie graphique de Renderscript comme une surcouche d'OpenGL. Le runtime se chargera pour vous d'initialiser les éléments nécessaires au rendu OpenGL pour tourner dans le GPU. Cette solution sera toujours plus simple que d'apprendre OpenGL et simplement intégrable dans vos applications.

Le code se base sur les API 14 (Android 4.0). Bien que présent depuis Android 3.0, les changements apportés depuis les API 14 permettent d'avoir un code moins verbeux.

Mais le raisonnement est identique et le portage vers Android 3.0 demandera un effort minimal.

CONCLUSION

Renderscript est un outil qui nous permet d'écrire du code performant et portable. Si l'année prochaine un appareil sous Android sort avec un processeur x86_64 et 12 coeurs, notre code est sûr de pouvoir s'exécuter dessus de manière optimale en profitant de l'ensemble des ressources disponibles.

Pour autant, tout n'est pas rose. Renderscript n'est disponible que sur 7% des appareils répertoriés par Google (Android 3.0 et +, juin 2012). Les outils de débogage sont inexistant et rien n'est proposé pour aider le développement dans nos IDE préférés. Les calculs ne sont pas déportés dans le GPU, et, même si Google aimerait bien inclure cette fonctionnalité, aucune date n'a été annoncée.

La documentation n'est pas très abondante. Il ne faut pas hésiter à aller voir les exemples fournis par Google ainsi que le code source d'Android, notamment des LiveWallPaper. Le code complet de cette application est disponible sous licence libre à l'adresse suivante : <https://github.com/sgaland/renderscript-particules>

Documentation officielle Renderscript :

<http://developer.android.com/guide/topics/graphics/renderscript/index.html>

APIs : <http://developer.android.com/reference/renderscript/index.html>

Article dev blog officiel :

<http://android-developers.blogspot.fr/2011/02/introducing-renderscript.html>

<http://android-developers.blogspot.fr/2011/03/renderscript.html>

<http://android-developers.blogspot.fr/2012/01/levels-in-renderscript.html>

Une présentation sur Renderscript par des ingénieurs de Google :

<http://www.youtube.com/watch?v=5jz0kSuR2j4>

Sylvain Galand

Ingénieur conception et développement - Genymobile

SharePoint 2010 : PowerShell et la brique Search

Dans cet article nous allons aborder la modification de propriété d'indexation du moteur de recherche, ainsi que celle des Métadonnées gérées de SharePoint 2010 en PowerShell. L'objectif recherché est de proposer des exemples simples, compréhensibles et de voir, via l'utilisation de ILSPY, que certaines commandes ne sont pas aussi complètes qu'on pourrait le penser.

CRAWLING

Trois cmdlets sont principalement utilisées pour le Crawling : **Get-SPEnterpriseSearchMetadataCrawledProperty**, qui permet de récupérer les propriétés, **Set-SPEnterpriseSearchMetadataCrawledProperty**, celle-ci pour les mettre à jour et **New-SPEnterpriseSearchMetadataCrawledProperty**, pour la création.

Voici un exemple afin de récupérer des composants existants et de les mettre à jour avec la propriété **IsMappedToContents** qui permet de faire un mapping sur des métadonnées existantes, nous utilisons un fichier XML pour la récupération des éléments.

```
<? xml version="1.0" encoding="utf-8"?>
<SearchProperties>
    <ServiceName>Search Service Application</ServiceName>
    <CrawledProperties>
        <CrawledProperty Name="ows_ContentType" />
    </CrawledProperties>
    <ExistingProperties>
        <ManagedProperty Name="ContentType"/>
    </ExistingProperties>
    <NewManagedProperties>
        <!--Types = 1:Text, 2:Integer,3:Decimal,4:DateTime,5:YesNo,6 :Binary-->
        <ManagedProperty Type="4" Name="Demo">
            <Map Category="Basic" Variant="64">15</Map>
        </ManagedProperty>
    </NewManagedProperties>
</SearchProperties>
```

Voici maintenant le code de Recherche :

```
if ( (Get-PSSnapin -Name Microsoft.SharePoint.PowerShell -ErrorAction SilentlyContinue) -eq $null )
{
    Add-PSSnapin Microsoft.SharePoint.PowerShell
}
#Fichier XML
[System.Xml.XmlDocument] $XmlDoc = new-object System.Xml.XmlDocument
$file = resolve-path("./MonFichier.xml")
if (!$file)
{
    Write-Host "Fichier non trouvé- Opération Annulé." -ForegroundColor red
    Break
}
write-host -f Green "Traitement du fichier:" $file
$xmlDoc = [xml](Get-Content $file)
```

```
#Parse des données
$sa = $XmlDoc.SearchProperties.ServiceName
$searchapp = Get-SPEnterpriseSearchServiceApplication $sa

$CrawledPropNodeList = $XmlDoc.SearchProperties.CrawledProperties

write-host -f Yellow " Recherche des données Crawl existantes :"
foreach ($CrawledPropNode in $CrawledPropNodeList.CrawledProperty)
{
    $SPCrawlProp = $CrawledPropNode.Name
#Vérification

    if($crawledproperty =Get-SPEnterpriseSearchMetadataCrawledProperty -Name $SPCrawlProp -VariantType $SPCrawlVariant -SearchApplication $searchapp -ea "silentlycontinue")
    {
        write-host -f Green "Des Crawl existent: Nom :" $crawledproperty.Name "- Indexé :" $crawledproperty.IsMappedToContents

        write-host "Mise à jour :"
        $crawledproperty | Set-SPEnterpriseSearchMetadataCrawledProperty -IsMappedToContents $True

        write-host -f Green "Nom :" $crawledproperty.Name "- Indexé :" $crawledproperty.IsMappedToContents

    }
    else
    {
        write-host -f Red "Veuillez vérifier dans le fichier XML et Sharepoint qu'ils existent"
    }
}
```

METADATA

Concernant les Metadata, ici nous utilisons deux méthodes **Get-SPEnterpriseSearchMetadataManagedProperty**, pour récupérer des métadonnées et **Set-SPEnterpriseSearchMetadataManagedProperty**, pour la mise à jour. Concernant les attributs nous utilisons **EnabledForScoping**, pour définir l'indexation, **FullTextQueryable**, pour rendre la propriété requêteable, **RespectPriority** qui correspond

dans votre interface utilisatrice aux phrases suivantes : « *Inclure les valeurs à partir de toutes les propriétés analysées mappées* » ou « *Inclure les valeurs à partir d'une propriété analysée unique en fonction de l'ordre spécifié* »

```
function ExistingMeta
{
#Gestion de propriétés pour Settings

$OthersPropertyNodeList = $XmlDoc.SearchProperties.ExistingProperties
foreach ($PropertyNode in $OthersPropertyNodeList.ManagedProperty)
{
    $SharePointProp = $PropertyNode.Name
    if($mp=Get-SPEnterpriseSearchMetadataManagedProperty -SearchApplication $searchapp -Identity $SharePointProp -ea "silentlycontinue" )
    {
        if($mp.EnabledForScoping -eq $False)
        {
            $mp | Set-SPEnterpriseSearchMetadataManagedProperty -EnabledForScoping 1
        }
        if($mp.FullTextQueriable -eq $False)
        {
            $mp | Set-SPEnterpriseSearchMetadataManagedProperty -FullTextQueriable 1
        }
        if($mp.RespectPriority -eq $False)
        {
            $mp | Set-SPEnterpriseSearchMetadataManagedProperty -RespectPriority 1 -ea "silentlycontinue"
        }
    }
}
}
```

Concernant la création nous utilisons la cmdlet suivante **New-SPEnterpriseSearchMetadataManagedProperty**

```
try
{
    write-host "Création :"
    $mp = New-SPEnterpriseSearchMetadataManagedProperty -SearchApplication $searchapp -Name $SharePointProp -Type $SharePointPropType -EnabledForScoping 1 -FullTextQueriable 1
    $mp.HasMultipleValues = $True
    $mp.Update()

}
catch [Exception]
{
    write-host -f Red "Erreur de création" $_;
}
```

Pour associer un mapping au Metadata il vous faut exécuter le code suivant :

```
$PropertyNodeList = $XmlDoc.SearchProperties.NewManagedProperties
foreach ($PropertyNode in $PropertyNodeList.ManagedProperty)
{
```

```
$SharePointProp = $PropertyNode.Name
$SharePointPropType = $PropertyNode.Type
$SharePointPropMapList = $PropertyNode.Map
$SharePointPropMultiple = $PropertyNode.Multiple

foreach ($SharePointPropMap in $SharePointPropMapList)
{
    $SPMapCat = $SharePointPropMap.Category
    $SPMapName = $SharePointPropMap.InnerText
    $SPVariant = $SharePointPropMap.Variant
    $cat = Get-SPEnterpriseSearchMetadataCategory -SearchApplication $searchapp -Identity $SPMapCat

    $prop = Get-SPEnterpriseSearchMetadataCrawledProperty -SearchApplication $searchapp -Category $cat -Name $SPMapName -VariantType $SPVariant
    write-host -f Cyan $cat.Name " - " $prop.Name

    $mp=Get-SPEnterpriseSearchMetadataManagedProperty -SearchApplication $searchapp -Identity $SharePointProp -ea "silentlycontinue"
    try
    {
        write-host "Création de la donnée finale"
        $property = New-SPEnterpriseSearchMetadataMapping -SearchApplication $searchapp -CrawledProperty $prop -ManagedProperty $mp
    }
    catch [Exception]
    {
        write-host -f Red "Erreur de création de la donnée finale" $_;
    }
}
```

ILSPY

Deux propriétés ne sont pas accessibles au niveau des options des cmdlets, celles qui apparaissent dans la section suivante de l'interface utilisatrice des métadonnées : « *Optimiser le stockage des propriétés gérées* ». Pour mettre à jour les propriétés, nous utilisons ILSPY afin de voir le code appelé dans la librairie suivante : Microsoft.Office.Server.Search.dll, puis nous cherchons la propriété suivante : Microsoft.Office.Server.Search.Internal.UI.ManagedPropertyPage, la méthode correspondante à `OkButtonClicked(object sender, EventArgs e)`. Dans cette méthode nous observons que deux propriétés n'existent pas dans les attributs des cmdlets, `QueryPropertyBlob = true`; et `PutInPropertyBlob = true`; Il ne vous reste plus qu'à les appeler dans votre code PowerShell de la façon suivante :

```
$mp.QueryPropertyBlob = $True
$mp.PutInPropertyBlob = $True
$mp.Update()
```

Notons toutefois que ces propriétés sont référencées dans la MSDN et sont pleinement accessibles en utilisant les API de SharePoint. En conclusion, nous avons la possibilité grâce à PowerShell d'automatiser certains processus de création et de modification des composants de recherche.

Nabil Babaci - Consultant SharePoint - MCNEXT
<http://dotnet4ever.fr>



Réaliser vos fractales

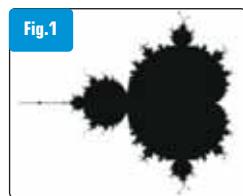
Les fractales ont toujours fasciné par leur complexité et leur rendu, même s'il s'agit de la mise en forme d'une fonction mathématique. Certains langages sont plus adaptés pour effectuer ce type de calculs car ils sont très gourmands au niveau de la puissance, mais le langage PHP peut aussi effectuer ce type de réalisation. La preuve !

Les fractales se définissent par un ensemble de points assez complexe et récursif. Elles ont été mises en avant par Benoit Mandelbrot, en 1975. Lorsque vous zoomez sur une zone, celle-ci doit vous permettre de retrouver l'image originale grâce à la définition de ses points.

> Rendu théorique

Il existe différentes techniques et approches pour réaliser des fractales suivant la formule appliquée, c'est ainsi que l'ensemble des calculs permet d'avoir une utilisation dans différents domaines :

- Les mathématiques
- L'économie
- Analyse statistique
- Etc.



Effet Mandelbrot

Lorsque vous utilisez la fractale Mandelbrot, le résultat affiche une forme très détaillée quels que soient le niveau d'agrandissement et la méthode de conception [Fig.1]. Cette représentation graphique montre une grosse forme appelée cardioïde entourée de petits cercles de tailles différentes. Chaque cercle autour du gros cercle représente une autre caractéristique des fractales, ce qui permet d'obtenir un effet de zoom de manière infinie. La formule mathématique se présente de la manière suivante : Partant de tout point $z = x + iy = (x, y)$, on construit la suite complexe (Z_n) par la formule de récurrence :

$$Z_0 = z$$

$$Z_{n+1} = Z_n^2 + C$$

Sa particularité est qu'il s'agit d'une seule Mandelbrot avec l'utilisation de la variable fixe 'C', ce qui peut donner une illusion plus importante de complexité.

Effet Julia

La fractale Julia peut être considérée comme une généralisation du domaine de Mandelbrot car la valeur 'C' n'est plus une valeur fixe mais une valeur variable, représentée de la manière suivante :

$C = x + yi$. Pour obtenir la figure 2, la valeur C devient $C = 0,72 + 0,11i$ [Fig.2].

Pour obtenir la figure 3, la valeur C devient $C = 0,28 - 0,02i$ [Fig.3].

Sa particularité de l'infini et de la complexité la rend plus représentative par rapport à une Mandelbrot classique, car chaque point sur lequel on zoomé peut représenter une forme différente et fortement similaire à l'ensemble de l'entier.



> Coder une fractale

Il existe différentes méthodes de calculs et de nuances pour réaliser une fractale. Nous allons présenter les 2 fractales les plus importantes : Mandelbrot et Julia.

Fractale Mandelbrot

Vue entière

Tout d'abord, lorsque vous lancez une fractale, il est important d'avoir la vue d'ensemble avec la définition de certains paramètres, qui sont les éléments clés de la Mandelbrot,

- $x1$ correspond à la limite gauche de l'image
- $x2$ correspond à la limite droite de l'image
- $y1$ correspond à la limite du haut de l'image
- $y2$ correspond à la limite du bas de l'image
- $precision$: niveau de précision

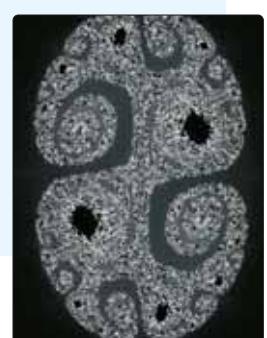
```
<?php
$x1 = -2.1;
$x2 = 0.6;
$y1 = -1.4;
$y2 = 1.2;
$zoom = 100;
$precision = 50;
?>
```

L'étape suivante crée une image true color, en utilisant une couleur noire si le croisement entre la position de la fractale et le point d'affichage est différent.

```
<?php
$largeur = 300;
$hauteur = 300;
$image = imagecreatetruecolor($largeur,$hauteur);
?>
```

L'étape la plus compliquée consiste à mettre en place la fonction mathématique Mandelbrot. Le principe de calcul détermine le test de chaque point (pixel) d'une image, à savoir, si le point correspond bien à la précision définie, ce point bénéficiera de la couleur définie. Si ce n'est pas le cas, une autre couleur sera affichée.

```
<?php
$x=0;
do {
  $y=0;
  do {
    $cr = $x / $zoom + $x1;
    $ci = $y / $zoom + $y1;
    $zr = 0;
    $zi = 0;
```





```

$im = 0;
do{
    $old_zr = $zr;
    $zr = $zr * $zr - $zi * $zi + $cr;
    $zi = 2 * $old_zr * $zi + $ci;
    $i++;
} while($zr*$zr + $zi*$zi < 4 AND $i < $precision);
if ($i == $precision) {
    $color = imagecolorallocate($image, 0, 0, 0);
    imagesetpixel($image, $x, $y, $color);
} else {
    $color = imagecolorallocate($image, $i*255/$precision,
    $i*255/$precision, $i*255/$precision);
    imagesetpixel($image, $x, $y, $color);
}
$y++;
} while ($y<$hauteur);
$x++;
} while ($x<$largeur);
?>

```

La partie de ce script montre que l'on boucle en x pour ensuite déterminer le statut des points en Y. Par ailleurs, chaque position effectue un calcul pour définir exactement si ce point fait partie de la Mandelbrot. Lorsque l'ensemble de l'image est calculé, il ne reste plus qu'à afficher le résultat dans une image au format PNG.

```

<?php
header('Content-type: image/png');
imagepng($image);
?>

```

Vue zoomer

Pour appliquer un ZOOM sur une fractale Mandelbrot, il est nécessaire d'effectuer quelques modifications au niveau des paramètres par défaut :

```

<?php
$x1 = -1.0;
$y1 = 0.25;
$zoom = 270;
$precision = 150;
?>

```

Les 2 premières valeurs à modifier concernent la définition de la position dans l'image au niveau du point de départ.

En plus de la position dans l'image, il faut prévoir le niveau du zoom et aussi le niveau de précision supérieure, car il est nécessaire d'être de plus en plus précis lors de la définition de cette nouvelle position, et montre la partie en bas à gauche de la fractale d'origine (vue entière).

Elle montre des petits détails supplémentaires et lorsque vous zoomez, vous retrouvez la forme de la fractale d'origine. Vous pouvez modifier ces valeurs pour effectuer des zooms différents.

Fractale Julia

La fractale Julia s'inspire de l'approche des fractales Mandelbrot. Sauf le coefficient multiplicateur, qui devient variable. Si nous reprenons l'exemple vu précédemment, en appliquant les critères par défaut :

```

<?php
$x1 = -0.9;
$x2 = 2;
$y1 = -1.0;
$y2 = 0.5;
$zoom = 150;
$precision = 100;
?>

```

et la formule mathématique Julia, le script PHP devient alors :

```

<?php
$cr = 0.285;
$ci = 0.01;
$zr = $x / $zoom + $x1;
$zi = $y / $zoom + $y1;
$i = 0;
do{
    $old_zr = $zr;
    $zr = $zr * $zr - $zi * $zi + $cr;
    $zi = 3 * $old_zr * $zi - $ci;
    $i++;
} while($zr*$zr + $zi*$zi < 4 AND $i < $precision);
?>

```

Suivant la complexité du calcul, le résultat sera différent, mais la technique de zoom reste identique. Pour effectuer cela, vous devez modifier les critères par défaut en appliquant bien le zoom et la précision, ou encore utiliser les valeurs de l'exemple précédent.

> Plus loin en 3D

Aujourd'hui, la 3D est entrée dans les esprits et il est donc logique que de nombreux passionnés de fractales se soient penchés sur ce sujet pour proposer la vue en version 3D. Cette forme de fractale s'appelle 'Mandelblub' et c'est le volume de Mandelbrot en 3D que propose Daniel White.

Pour réaliser cette fractale, il n'est pas possible d'appliquer le calcul par tranche qui est utilisé pour la fractale 2D, il faut partir d'un point de vue trigonométrie. Autrement dit, l'algorithme nécessaire pour effectuer les calculs est :

```

double pi=3.14159265;
double r      = sqrt(x*x + y*y + z*z );
double yang = atan2(sqrt(x*x + y*y) , z ) ;
double zang = atan2(y , x);
newx = (r*r) * sin( yang*2 + 0.5*pi ) * cos(zang*2 +pi);
newy = (r*r) * sin( yang*2 + 0.5*pi ) * sin(zang*2 +pi);
newz = (r*r) * cos( yang*2 + 0.5*pi );

```

Son auteur propose de nombreuses réalisations et rentre dans le détail pour réaliser les fractales 'Mandelblub' sur le lien suivant <http://www.skytopia.com/project/fractal/mandelbrot.html#pandora>

> Pour en savoir plus

L'ensemble des scripts est disponible directement sur le site <http://www.hello-design.fr>.

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.

Personnaliser la gestion des produits sous Ubercart/Drupal 6

Le CMS Drupal 6 et le module de e-commerce Ubercart constituent un environnement d'une richesse incomparable. Cependant, un jour ou l'autre, vous devrez utiliser leur flexibilité pour les adapter à vos besoins particuliers. Nous voyons aujourd'hui comment personnaliser la gestion des produits d'un site de e-commerce.

Drupal est un CMS open source écrit en PHP, hautement polyvalent nativement. Environnement totalement ouvert, il est entièrement personnalisable au moyen de modules d'extension. Certains de ces modules sont eux-mêmes des environnements complets. Tel est le cas d'Ubercart, une suite de e-commerce dotée de très riches fonctionnalités, qu'il est possible d'enrichir encore par des modules d'extensions à Ubercart. Ces modules d'extensions à Ubercart sont forts nombreux et disponibles à drupal.org. Cependant il arrivera toujours un moment où vous aurez un tout petit besoin si spécifique que personne n'aura développé de module pour cela. Il devient donc nécessaire d'écrire un peu de code.

1 CAHIER DES CHARGES

Récemment, j'ai été confronté à une limitation un peu stupide d'Ubercart, relative à l'ordre d'affichage des produits dans le catalogue. Cet ordre est défini à partir d'un champ nommé 'List position' dans le formulaire de saisie d'un produit. Le problème est que ce champ est une liste déroulante dont les valeurs s'étendent de -25 à 25, ce qui fait un maximum de 51 valeurs. Or, il était question de faire un site qui présente des magazines au format papier ou PDF, dont les numéros dépassaient très largement la centaine. Il était donc nécessaire de lever cette limitation. En outre, il était nécessaire de signaler les numéros épuisés du magazine papier. Nous allons voir ensemble comment traiter cela, ce qui nous amènera à découvrir d'autres aspects de Drupal que nous n'avons pas eu l'occasion d'aborder dans de précédents articles. Cet article traite de Drupal 6, mais les principes sont directement transposables à Drupal 7. Le lecteur trouvera tous les sources qui accompagnent cet article sur notre site www.programmez.com.

2 CE QU'IL NE FAUT PAS FAIRE

L'ordre d'affichage des produits apparaît dans le catalogue. Si l'on consulte les fichiers du sous-module Catalogue (uc_catalogue) d'Ubercart, on trouve ce code dans le fichier uc_catalogue.pages.inc :

```
// Build an ORDER BY clause for the SELECT query based on table sort info.
if (empty($_REQUEST['order'])) {
  $order = 'ORDER BY p.ordering, n.title, n.nid';
}
else {
  $order = tapirsort_sql(uc_product_table_header());
}

$sql = «SELECT DISTINCT(n.nid), n.sticky, n.title, n.created,
```

```
p.model, p.sell_price, p.ordering
FROM {node} n
  INNER JOIN {term_node} tn ON n.vid = tn.vid
  INNER JOIN {uc_products} AS p ON n.vid = p.vid
WHERE tn.tid = %d AND n.status = 1
  AND n.type IN («. db_placeholders($types, 'varchar') .») .. $order;
```

Nous comprenons que le système extrait de la table des nœuds (sous Drupal 6 tout (ou presque :) est un nœud) tous les noeuds qui sont publiés, et qui sont aussi des produits, sous une catégorie du catalogue, qui n'est rien d'autre qu'un vocabulaire. Ensuite ces résultats sont triés par une savante clause ORDER BY qui trie d'abord en fonction d'un enregistrement 'ordering' de la table uc_products. Cet enregistrement correspond au champ 'List Position' du formulaire de saisie d'un produit. Nous savons que nous sommes limités ici. Puis à valeurs ordering égales, le tri s'effectue par ordre alphabétique sur le titre des nœuds puis éventuellement sur les identifiants des noeuds, ce qui ne peut nous convenir. Ici, grande est la tentation de modifier ORDER BY p.ordering pour trier à partir d'un champ qui sera ajouté au formulaire de création d'un produit. Une demi ligne de code et c'est terminé, le comportement de Drupal est ajusté à nos besoins. C'est toutefois ce qu'il ne faut pas faire. Ce serait en opposition totale avec la philosophie de Drupal. La raison la plus évidente est qu'une mise à jour d'Ubercart écraserait notre code, et notre site ne se comporterait plus correctement.

3 CE QU'IL FAUT FAIRE : TRAVAILLER AVEC LES HOOKS

Toute l'architecture de Drupal repose sur le principe des hooks. Les hooks sont des fonctions de rappels que le système invoque automatiquement et à l'intérieur desquels il est possible d'agir pour modifier à la volée le comportement natif. Ubercart, véritable système dans le système, est un groupe de modules écrits dans la philosophie Drupal. En tant que tel, il expose un jeu de hooks lui aussi. Ceux-ci sont documentés dans le fichier docs/hooks.php que vous trouverez dans l'arborescence des sources de Ubercart. Les hooks de Drupal sont bien évidemment documentés sur drupal.org. Se brancher sur un hook est on ne peut plus simple. Il suffit d'écrire une fonction dont le nom obéit à une convention de nommage, et qui reçoit les arguments prévus par sa documentation.

4 LES FORMULAIRES DRUPAL

Drupal propose une API très puissante pour la création de formulaires de pages Web. Plus question de travailler fastidieusement avec les balises HTML. Au lieu de cela, le développeur qui souhaite construire un formulaire donne une structure de données hiérar-

chique construite à base de tableaux. Les propriétés et attributs des balises HTML sont décrites par des paires clé/valeur des tableaux. Le code ci-dessous montre comment construire un formulaire ne comportant qu'un seul bouton submit :

```
function monformulaire(&$form_state) {
  $form['submit'] = array(
    '#type' => 'submit',
    '#value' => t('Enregistrer'),
  );
}

function contruire_formulaire () {
  return drupal_get_form('monformulaire');
}
```

Le formulaire est construit avec l'appel à drupal_get_form, qui va invoquer automatiquement la fonction monformulaire. A l'issue de l'opération, un formulaire d'identifiant monformulaire est connu par le système. Le système construit des formulaires de façon similaire et permet ensuite de les modifier à la volée au moyen de hooks qui peuvent obéir à plusieurs conventions de nommage. La plus simple étant un nom de fonction se terminant par form_alter. Nous en savons assez pour aborder l'écriture de notre module. Nous supposons que le lecteur dispose d'un Ubercart installé et que le module CCK est installé également. Nous supposons qu'il a créé des produits de type 'magazine' et 'ebook', et que pour eux il a créé un champ CCK 'field_issue' [Fig.1] qui servira à numérotter les magazines. Pour mettre tout cela en place, le lecteur pourra utiliser le dump de MySQL que nous fournissons sur notre site en accompagnement de cet article.

5 NOTRE MODULE

Celui-ci doit agir sur les produits, le catalogue et le panier, et réagir à l'état du stock. Nous consignons tout cela dans le fichier descriptif de notre module proguc.info pour éviter de rencontrer un problème de dépendances :

```
; $Ids
name = Programmez Ubercart
description = Ajoute des fonctionnalités à Drupal et Ubercart
spécifiques au site Programmez
package = Programmez
core = 6.x
```

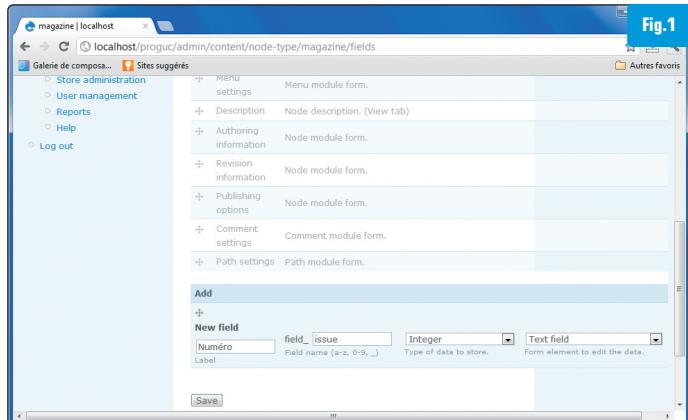


Fig.1

Un champ sera ajouté à nos types de contenu, pour stocker le numéro d'un magazine.

```
version = «6.x-1.0»
php = 5.2

dependencies[] = uc_cart
dependencies[] = uc_stock
dependencies[] = uc_catalog
dependencies[] = uc_product
```

Ensuite, nous implémentons hook_form_alter dans le fichier de code du module (proguc.module). Ce hook va nous permettre d'agir sur les formulaires d'édition des produits. Ces formulaires sont générés par Drupal, non par Ubercart. Notre hook sera donc tout à fait classique :

```
/*
 *  implémentation de hook_form_alter(&$form, $form_state, $form_id)
 *
 *  hooker de façon classique certains formulaires
 */
function proguc_form_alter(&$form, $form_state, $form_id) {
  //echo $form_id . '<br />'; // pour essais
  switch ($form_id) {
    case 'magazine_node_form':
      proguc_modifie_magazine_node_form($form, $form_state, $form_id);
      break;
    case 'ebook_node_form':
      proguc_modifie_ebook_node_form($form, $form_state, $form_id);
      break;
    default:
      break;
  }
}
```

La fonction du hook reçoit trois arguments, le formulaire sur lequel on peut agir, son état et son identifiant. Notre première difficulté est que l'on ne connaît pas les identifiants des formulaires. Une petite instruction echo \$form_id est là très instructive. Nous apprenons que les formulaires Drupal d'édition des produits ont des identifiants tout à fait logiques. Notre hook peut donc les traiter facilement. En revanche l'affichage d'une page de catalogue fait apparaître des identifiants de formulaires peu sympathiques tels que : [Fig.2].

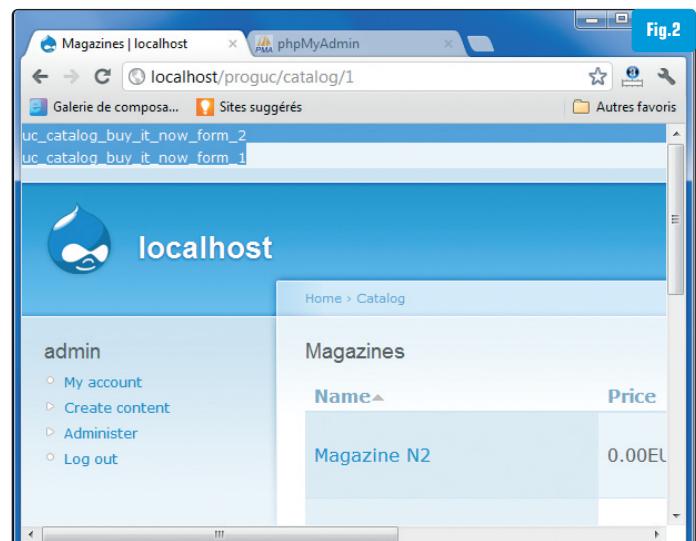


Fig.2

Ubercart génère des identifiants de formulaires tels qu'un hook Drupal classique ne conviendra pas pour les traiter.

```
uc_catalog_buy_it_now_form_2
uc_catalog_buy_it_now_form_1
```

La difficulté est que les identifiants sont décorés de la valeur de leur rang dans le catalogue. Bien sûr, il serait malgré tout possible de traiter cela dans le 'hook Drupal', mais nous verrons plus loin qu'il est beaucoup plus aisés de le faire dans un 'hook Ubercart'. Regardons maintenant le code de la fonction `proguc_modifie_magazine_node_form` qui modifie le formulaire d'édition d'un magazine au format électronique :

```
function proguc_modifie_ebook_node_form(&$form, $form_state, $form_id) {
  unset($form['base']['ordering']['#delta']);
  $form['base']['ordering']['#type'] = 'textfield';
  $form['base']['ordering']['#size'] = 6;
  $form['base']['ordering']['#disabled'] = true;
  $form['base']['ordering']['#description'] = «La position d'un magazine dans le catalogue est redéfinie par le négatif du numéro du magazine, donné ci-dessous»;
  $form['#validate'][] = 'proguc_ebook_node_form_validate';

  $form['base']['shippable']['#disabled'] = true;
  $form['base']['shippable']['#default_value'] = false;
}
```

Le code commence par travailler sur le widget `ordering` qui apparaît par défaut dans le formulaire. Tout d'abord la plage de valeurs est détruite. Ensuite nous modifions carrément le type du widget, pour le transformer en une ligne de texte capable d'afficher 6 caractères (voyons grand :) Ensuite le widget est désactivé afin qu'il ne soit plus possible à l'utilisateur qui fait la saisie d'un produit d'agir sur l'ordre de classement depuis cet endroit. Il pourrait être pertinent de carrément cacher ce widget en situation réelle. Juste après, nous expliquons à l'utilisateur de quoi il retourne au moyen d'un commentaire placé sous notre widget transformé. Ensuite, nous déclarons une fonction qui sera invoquée lorsque l'utilisateur validera le formulaire, c'est-à-dire, lorsqu'il cliquera sur le bouton enregistrer. Nous examinerons le code de cette fonction plus loin. Comme on le voit dans le code ci-dessus, un formulaire peut se voir attribuer une ribambelle de fonctions de validations, celles-ci étant rangées dans un tableau. Il n'y a aucune garantie quant à l'ordre d'invocation des fonctions de ce tableau. Enfin, comme nous travaillons sur le formulaire d'un magazine électronique à télécharger, il n'est pas question que le service client du site expédie celui-ci par voie postale. Donc nous profitons de l'occasion pour décocher la case 'shippable' et la désactiver afin que l'utilisateur ne puisse plus la cocher par un clic de souris malencontreux. Nous arrivons maintenant au code de validation du formulaire :

```
function proguc_ebook_node_form_validate($form, &$form_state) {
  $form_state['values']['ordering'] = -$form_state['values']['field_issue'][0]['value'];
  $form_state['values']['shippable'] = false; // pour assurer :-
}
```

Ici nous lisons la valeur dans le champ `field_issue` du formulaire, valeur qui doit contenir le numéro du magazine en cours de saisie. Puis nous plaçons le négatif de cette valeur dans le champ `ordering` que nous avons travaillé plus haut. Ceci parce qu'Ubercart classe les produits par valeurs 'ordering' croissantes. Attention: Nous avons

ici, dans notre code d'exemple, une faille de sécurité potentielle. En effet, nous ne vérifions pas la validité de la saisie de l'utilisateur. Nous laissons la sécurisation de la saisie au lecteur.

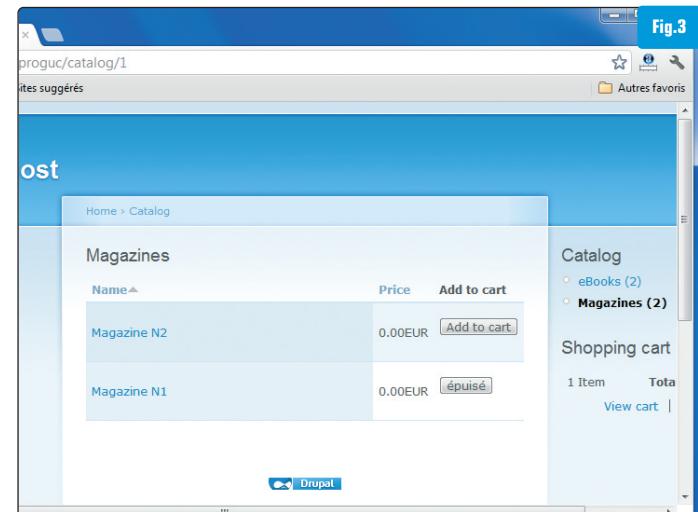
6 MODIFIER UN FORMULAIRE UBERCART

Comment faire pour présenter un produit au catalogue, qui est momentanément épuisé, sans que le client puisse l'ajouter à son panier ? Une possibilité est d'afficher le mot 'épuisé' au lieu du message 'Ajouter au panier' dans le bouton. [Fig.3] Là encore cela revient à modifier un formulaire. Cette fois, il est plus aisés de placer le hook dans l'API Ubercart, comme nous l'avons mentionné plus haut.

```
function proguc_uc_form_alter(&$form, $form_state, $form_id) {
  //echo $form_id . '<br />';
  switch ($form_id) {
    case 'uc_product_add_to_cart_form':
    case 'uc_catalog_buy_it_now_form':
      proguc_modifie_uc_product_add_to_cart_form($form, $form_state, $form_id);
      break;
    default:
      break;
  }
}
```

Là encore, une petite instruction `echo $form_id;` saura nous informer des différents identifiants de formulaires que nous pouvons rencontrer. Examinons maintenant le code qui agit sur le formulaire.

```
function proguc_modifie_uc_product_add_to_cart_form(&$form, $form_state, $form_id) {
  $node = node_load($form['nid']['#value']);
  $in_catalog = count(taxonomy_node_get_terms_by_vocabulary($node, variable_get('uc_catalog_vid', 0)));
  // si le produit est au catalogue, mais n'est pas en stock
  // on affiche un bouton 'épuisé'
  if (uc_stock_level($node->model) <= 0
```



Quand le stock d'un produit est épuisé, nous modifions le comportement du bouton 'Ajouter au panier'.

```

    && uc_stock_is_active($node->model) && $in_catalog != 0) {

    $form['#validate'][] = 'proguc_add_to_cart_never_form_
validate';

    $form['submit'] = array(
      '#type' => 'submit',
      '#value' => 'épuisé',
      '#id' => 'edit-submit-'. $node->nid,
      '#attributes' => array(
        'class' => 'node-add-to-cart'),
    );
  }

  return;
}
}

```

Cet exemple nous montre de nouvelles possibilités, fort utiles. Tout d'abord comment charger le nœud/produit concerné par le formulaire. Ensuite nous voyons comment détecter si le produit est au catalogue, ce qui revient à demander si un terme du vocabulaire du catalogue lui est attribué. Ensuite nous voyons si le stock est activé pour ce produit, et si oui, nous réagissons à un niveau de stock inférieur ou égal à zéro en modifiant le bouton. Contrairement à l'exemple précédent où nous avons modifié le widget attribut par attribut, là nous redéfinissons d'un coup le bouton submit en fournissant un tableau d'attributs sous forme de paires clé/valeur. Voici maintenant la fonction de validation associée, fonction qui provoque systématiquement l'échec de l'ajout au panier d'un produit dont le stock est épuisé :

```

function proguc_add_to_cart_never_form_validate($form, &$form_
state) {
  form_set_error('', t('Vous ne pouvez pas ajouter un produit
épuisé au panier!'));
}

```

7 MODIFIER LE SCHÉMA DE BASE DE DONNÉES

Si nous avons bien avancé, il reste un problème à traiter. Problème qui ne saute pas aux yeux, mais dont on doit toujours se méfier lorsque l'on modifie la plage d'une valeur stockée en base de données. Le module product d'Ubercart stocke la valeur ordering dans la table uc_products avec le type de données tinyint, [Fig.4] ce qui nous limite à une plage de valeurs de -127 à 127. Ceci ne peut nous convenir car nous allons très rapidement nous heurter à un dépas-

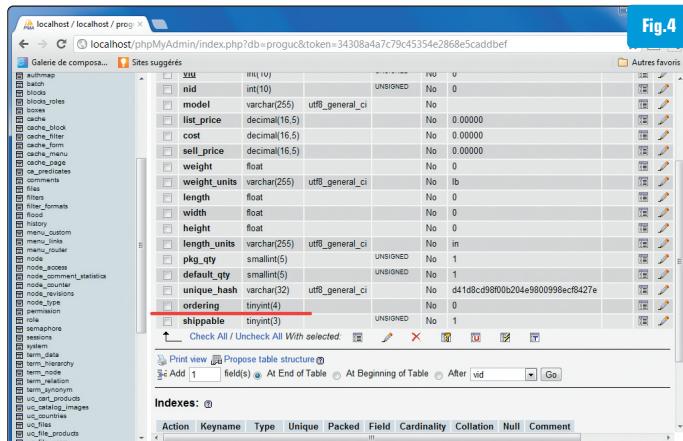


Fig.4

Par défaut la valeur de classement des produits est de type tinyint.

sement de capacité. Nous allons donc modifier la table afin de disposer d'un smallint [Fig.5]. Pour cela, nous allons placer quelques fonctions de hook dans le fichier d'installation du module proguc.install. Tout d'abord hook_install/uninstall :

```

function proguc_install() {
  db_change_field($ret,
    'uc_products',
    'ordering',
    'ordering',
    array(
      'type' => 'int',
      'size' => 'small',
      'not null' => TRUE,
      'default' => 0,
    )
}

function proguc_uninstall() {
}

```

Ainsi à l'installation du module, la table uc_products sera modifiée selon nos besoins. Ceci ne peut suffire. Il faut que tout module qui en aurait besoin soit informé du changement du schéma de la base de données. Pour cela, nous devons implémenter hook_schema_alter :

```

function proguc_schema_alter(&$schema) {
  $schema['uc_products']['fields']['ordering'] = array(
    'type' => 'int',
    'size' => 'small',
    'not null' => TRUE,
    'default' => 0,
  );
}

```

Frédéric Mazué - fmazue@programmez.com

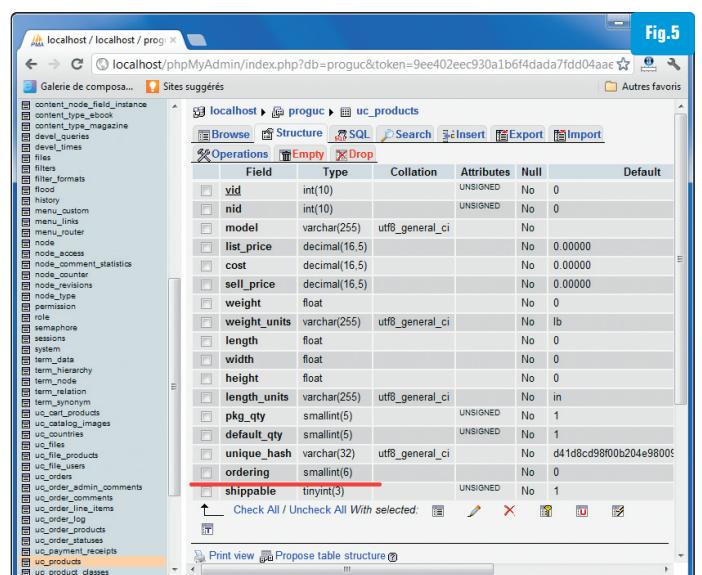


Fig.5

Nous modifions uc_products afin de ranger la valeur dans un smallint.

Administre le RAID d'un serveur Linux

Pour un webmaster, la sécurité d'un serveur et la préservation des données qui lui sont confiées sont des préoccupations constantes. Avec cet article, nous abordons quelques bonnes pratiques concernant le deuxième point.

Le gros problème existentiel d'un webmaster est qu'il est toujours sur la défensive. Ce sont les hackers qui ont la vie belle ! Ils viennent essayer des techniques qui fonctionnent ... ou pas. Le webmaster ne sait pas à l'avance, lui, avec quelles techniques son serveur ou son site sera attaqué, ni à quels problèmes techniques il devra faire face. C'est pour cette raison que l'on dit souvent qu'un bon webmaster est un authentique paranoïaque :-) Un bon webmaster ne prend jamais assez de précautions, et si le pire arrive, il doit être en mesure de retrouver ses données quelque part pour remettre le ou les sites dont il a la charge en production. Qu'est-ce que le pire ? Un serveur hacké avec un pirate malveillant qui efface tout le disque ... ou un disque qui s'efface tout seul, c'est-à-dire qui tombe en panne. Nous allons voir quelques bonnes pratiques pour faire face sans paniquer à une telle situation qui se produira inévitablement un jour ou l'autre. Cet article s'adresse plus particulièrement au webmaster qui débute dans l'administration d'un serveur Linux.

1 UN RAID SUR VOS DONNÉES

Si vous avez la possibilité de choisir le serveur dont vous aurez la responsabilité, n'hésitez pas une seconde et optez pour un serveur qui dispose d'un RAID, opportunité qu'offre la grande majorité des hébergeurs aujourd'hui. Sur leurs offres de serveurs, vous verrez, au chapitre hardware : RAID soft 0/1. Cela signifie que vous pourrez faire tourner votre serveur soit avec un RAID 0, soit avec un RAID 1, ce dernier étant en général le RAID installé par défaut lorsque la machine est livrée. RAID est une technologie qui date de 1987 et qui signifiait à l'origine Redundant Arrays of Inexpensive Disks, pour devenir Redundant Array of Independent Disk. Cette technologie consiste à répartir des données selon différentes configurations, sur plusieurs disques durs qui forment une grappe. La gestion de la grappe peut être assurée par le matériel lui-même, ou par un logiciel. C'est dans ce dernier cas que l'on parle de RAID soft. Il existe de nombreux types de RAID, les plus courants étant le 0 et le 1 sur les serveurs. Un RAID 0 consiste à saucissonner la capacité des disques pour faire travailler ceux-ci en parallèle. Cet arrangement vise à gagner en performances lors des opérations de lecture/écriture, mais surtout d'écriture. Sur un serveur destiné à héberger des sites Internet, ceci est de relativement peu d'intérêt, car les opérations d'écriture sont très peu nombreuses en comparaison des opérations de lecture. Le RAID 1 est une configuration en miroir. Chaque disque de la grappe, en général constituée de deux disques, contient les mêmes données. Cette configuration est excellente pour un serveur, car elle apporte la **tolérance aux pannes**. Tout simplement, quand un disque tombe en panne, le système continue de fonctionner avec le ou les disques restants. Lorsque le disque défectueux est remplacé, une opération de reconstruction du RAID est faite, qui duplique les données à l'identique sur le nouveau disque.

2 SURVEILLER L'ÉTAT DU RAID

Normalement votre hébergeur se charge de ce travail, et réagit en cas de panne d'un disque. Mais un administrateur paranoïaque préférera effectuer une petite surveillance lui-même, ce qui est très simple. Linux conserve l'état du RAID dans mdstat, un fichier du pseudo système de fichier /proc. L'état du raid se lit ainsi :

```
cat /proc/mdstat
```

Et l'on obtient un affichage semblable à ceci :

```
Personalities : [linear] [raid0] [raid1] [raid10] [raid6] [raid5]
[raid4] [multipath] [faulty]
md1 : active raid1 sdb1[1] sda1[0]
      10238912 blocks [2/2] [UU]

md2 : active raid1 sdb2[1] sda2[0]
      721806272 blocks [2/2] [UU]

unused devices: <none>
```

Nous avons ici un RAID 1 dont la grappe est constituée de deux disques. Les deux disques sont fonctionnels comme en témoigne [UU], avec U pour Up. Si jamais vous obtenez un affichage comme ceci :

```
Personalities : [linear] [raid0] [raid1] [raid10] [raid6] [raid5]
[raid4] [multipath] [faulty]
md1 : active raid1 sdb1[1] sda1[0]
      10238912 blocks [2/2] [U_]

md2 : active raid1 sdb2[1] sda2[0]
      721806272 blocks [2/2] [U_]
```

Alors l'un des disques est cassé (sda dans cet exemple), que votre hébergeur vous en ait averti ou pas. Vous devez donc le contacter pour qu'il remplace le disque défaillant. Bien entendu, il ne fait pas sens de surveiller ainsi ponctuellement et surtout manuellement. On automatisera en faisant exécuter quotidiennement par cron un script shell dans cet esprit :

```
#!/bin/bash
RECIPIENT=$(cat /root/.email)
cat /proc/mdstat | mail -s "[IMPORTANT] etat RAID de mon serveur" $RECIPIENT
```

Ce script vous enverra un mail que vous aurez tout loisir de lire en prenant votre café le matin :-)

3 RECONSTRUIRE UN RAID

Le disque ayant été changé, vous avez un disque qui contient vos précieuses données, et un disque vierge, même pas formaté. Il est nécessaire de reconstruire le RAID, afin que les données du disque survivant

soient répliquées sur le disque qui vient d'être remplacé. Votre hébergeur considérera très probablement que c'est de votre ressort d'effectuer cette opération. Ce n'est pas une opération difficile en soi, mais si on se trompe, on peut détruire les données sur le disque valide, ce qui est gênant, car cela va vous obliger à réinstaller le système et à réinstaller tous les sites que vous avez régulièrement sauvegardés sur une autre machine (paranoïa toujours). L'interruption de service sera toutefois longue, donc mieux vaut éviter. Voici quelques conseils.

4 REBOOTER LA MACHINE

Très probablement, votre hébergeur vous aura annoncé le changement du disque et vous aura rendu la machine dans un état où elle ne fonctionne pas pleinement, un mode 'rescue', ou une appellation du même genre selon votre hébergeur. Vous devez donc commencer par redémarrer la machine, depuis son interface d'administration que vous fournit votre hébergeur. L'opération de redémarrage peut s'avérer très longue car le système va faire une vérification du système de fichier (qui est probablement ext3). Pas de panique, et se faire un café en attendant. Si le disque changé est le second (sdb) l'opération de redémarrage se passera probablement bien car le secteur amorce est présent sur sda. Si le disque changé est sda, très probablement la machine ne pourra pas démarrer. A moins que vous n'utilisiez Grub comme chargeur de démarrage et que vous n'ayez configuré celui-ci pour qu'il soit présent sur tous les disques du RAID. Si ce n'est pas le cas et que la machine ne démarre pas, ne touchez pas paniquer, mais rechercher dans l'interface d'administration la configuration pour démarrer en boot réseau. Dans ce mode, le noyau est chargé via le réseau de l'hébergeur par le BIOS de la machine. Vous pouvez choisir la version du noyau qui vous convient, et choisir quel disque va être monté. Votre serveur va donc démarrer, mais la patience est toujours de rigueur, à cause de la vérification du système de fichier.

5 ATTENTION À LA LOI DES « EMBÈTEMENTS » MAXIMUM

Cette loi implacable a donné des cheveux blancs à plus d'un webmaster ou administrateur système... La situation est pour l'instant la suivante : nous avons un serveur qui tourne parfaitement, sur un seul disque. Les visiteurs des sites ne se rendent compte de rien, votre patron non plus. Grande est la tentation de différer la reconstruction du RAID. L'opération paraît compliquée, hasardeuse, et vous avez envie de la faire à tête reposée. C'est une erreur, et voici pourquoi. Quand vous avez loué votre serveur, votre hébergeur l'a assemblé et monté deux disques. Votre hébergeur commande les disques par camions... Très probablement les deux disques qu'il a montés sur votre serveur sont de la même marque, et de la même série. Et très probablement leur durée de vie va être identique. Comme vous êtes un webmaster/administrateur paranoïaque, vous sentez que vous êtes sous le coup de la menace d'une double panne. Que personne ne rie dans la salle, les double crash disques sont une mésaventure beaucoup plus fréquente qu'on ne le pense. Il ne faut donc pas tarder à reconstruire le RAID.

6 COMPRENDRE LE RAID DE SA MACHINE

Nous allons maintenant nous attacher à bien comprendre l'anatomie d'un RAID, avant de le reconstituer. Le lecteur trouvera sans doute

avantage à adapter ce que nous allons dire à son ou ses serveurs et à constituer ainsi un petit manuel pratique qui lui servira à faire face à un cas d'urgence. Le stress n'est pas l'ami du webmaster lors de la reconstitution d'un RAID. Nous partons du principe que nous travaillons avec un serveur dont la grappe est constituée de deux disques identifiés par le système respectivement par /dev/sda et /dev/sdb. Les disques sont, par définition du RAID, formatés à l'identique comme suit. Une "petite" partition (1) sur laquelle est montée, et qui contient donc le système d'exploitation et le répertoire de root, une "grosse" partition 2 sur laquelle est montée home et qui contient toutes les données, et une partition 3 de swap. Nous pouvons visualiser ceci avec la commande fdisk -l saisie dans un terminal, comme illustré. [Fig.1] **ATTENTION** : à partir de maintenant, si vous expérimentez en lisant cet article, sachez que vous n'avez pas le droit à l'erreur de frappe sous peine de risquer de perdre vos précieuses données :

```
# fdisk -l

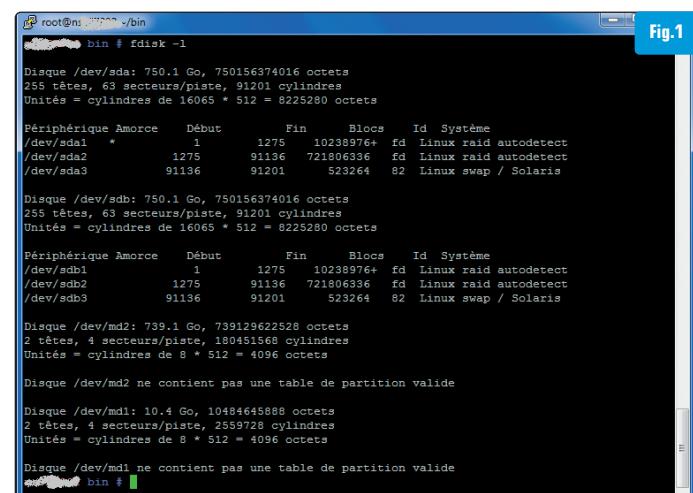
Disque /dev/sda: 750.1 Go, 750156374016 octets
255 têtes, 63 secteurs/piste, 91201 cylindres
Unités = cylindres de 16065 * 512 = 8225280 octets

Périphérique Amorce Début Fin Blocs Id Système
/dev/sda1 * 1 1275 10238976+ fd Linux raid autodetect
/dev/sda2 1275 91136 721806336 fd Linux raid autodetect
/dev/sda3 91136 91201 523264 82 Linux swap / Solaris

Disque /dev/sdb: 750.1 Go, 750156374016 octets
255 têtes, 63 secteurs/piste, 91201 cylindres
Unités = cylindres de 16065 * 512 = 8225280 octets

Périphérique Amorce Début Fin Blocs Id Système
/dev/sdb1 1 1275 10238976+ fd Linux raid autodetect
/dev/sdb2 1275 91136 721806336 fd Linux raid autodetect
/dev/sdb3 91136 91201 523264 82 Linux swap / Solaris

Disque /dev/md2: 739.1 Go, 739129622528 octets
2 têtes, 4 secteurs/piste, 180451568 cylindres
Unités = cylindres de 8 * 512 = 4096 octets
```



La commande fdisk permet de visualiser le partitionnement des disques de votre RAID.

```
Disque /dev/md2 ne contient pas une table de partition valide

Disque /dev/md1: 10.4 Go, 10484645888 octets
2 têtes, 4 secteurs/piste, 2559728 cylindres
Unités = cylindres de 8 * 512 = 4096 octets

Disque /dev/md1 ne contient pas une table de partition valide
```

Vous pouvez clarifier les sorties de cette commande en donnant successivement fdisk -l /dev/sda puis fdisk -l /dev/sdb. Nous retrouvons bien ce qui était annoncé au niveau des partitions. Nous voyons aussi que seul le premier disque contient le chargeur du système. (Sur la machine qui sert de base à cet article, c'est lilo qui est le bootloader). La sortie émise par notre commande nous présente également deux disques, ou plutôt deux périphériques, /dev/md1 et /dev/md2 dont il est dit qu'ils ne contiennent pas de tables de partition valides. Ne nous affolons pas, c'est normal. Ces disques n'existent pas physiquement. Ce sont les disques vus par le RAID logiciel (ou RAID soft). Il est ici important de comprendre que ce qui est appelé ici disque, n'en est en fait pas tout, au sens disque complet et physique du terme. Au contraire /dev/md1 et /dev/md2 correspondent à des partitions. Cela se comprend plus clairement en examinant la fstab du système : [Fig.2].

```
# <fs>           <mountpoint>   <type>      <opts>
                  <dump/pass>

# NOTE: If your BOOT partition is ReiserFS, add the notail option
# to opts.
# <file system> <mount point><type><options><dump>  <pass>
/dev/md1  /  ext3errors=remount-ro  0  1
/dev/md2  /home  ext3defaults,usrquota  1  2
/dev/sda3  swap  swapdefaults0  0
/dev/sdb3  swap  swapdefaults0  0

# NOTE: The next line is critical for boot!
Proc    /proc  proc  defaults  0  0

Sysfs  /sys  sysfs  defaults  0  0
```

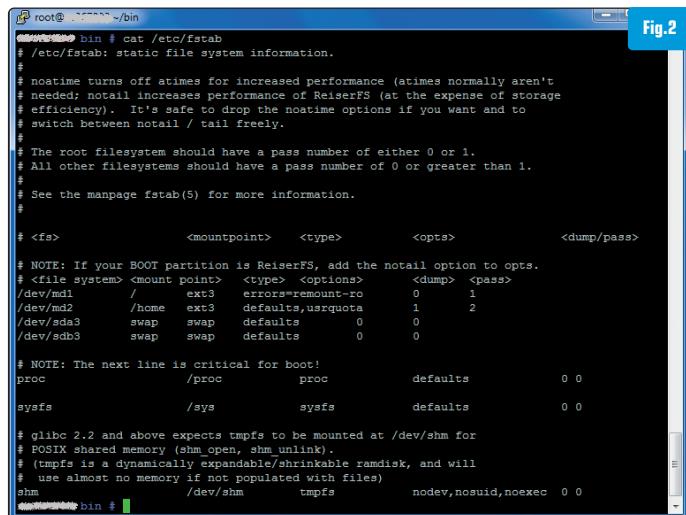


Fig.2

```
root@...:~# cat /etc/fstab
# /etc/fstab: static file system information.

#  
# noatime turns off atimes for increased performance (atimes normally aren't  
# needed; notail increases performance of ReiserFS (at the expense of storage  
# efficiency). It's safe to drop the noatime options if you want and to  
# switch between notail / tail freely.  
#  
# The root filesystem should have a pass number of either 0 or 1.  
# All other filesystems should have a pass number of 0 or greater than 1.  
#  
# See the manpage fstab(5) for more information.  
#  
# <fs>           <mountpoint>   <type>      <opts>      <dump/pass>  
  
# NOTE: If your BOOT partition is ReiserFS, add the notail option to opts.
# <file system> <mount point> <type> <options> <dump> <pass>
/dev/md1  /  ext3  errors=remount-ro  0  1
/dev/md2  /home  ext3  defaults,usrquota  1  2
/dev/sda3  swap  swap  defaults  0  0
/dev/sdb3  swap  swap  defaults  0  0  
  
# NOTE: The next line is critical for boot!
proc    /proc  proc  defaults  0  0  
  
sysfs  /sys  sysfs  defaults  0  0  
  
# glibc 2.7 and above expects tmpfs to be mounted at /dev/shm for
# POSIX shared memory (shm_open, shm_unlink).
# (tmpfs is a dynamically expandable/shrinkable ramdisk, and will
# use almost no memory if not populated with files)
shm    /dev/shm  tmpfs  nodev,nosuid,noexec  0  0
```

Les périphériques RAID correspondent à des partitions, non à des disques complets.

```
# glibc 2.2 and above expects tmpfs to be mounted at /dev/shm for
# POSIX shared memory (shm_open, shm_unlink).
# (tmpfs is a dynamically expandable/shrinkable ramdisk, and will
# use almost no memory if not populated with files)
Shm  /dev/shm  tmpfs  nodev,nosuid,noexec  0  0
```

Il devient ici très clair que /dev/md1 correspond au point de montage / (ou "petite" partition, ou partition 1) et que cette correspondance, par la logique même d'un RAID 1 ou RAID en miroir, s'applique à la partition 1 de chacun des deux disques. Autrement dit, quand une opération d'écriture sera effectuée sur /dev/md1, ce sont les deux disques recouverts par /dev/md1 qui verront cette opération d'écriture répercutee vers leur partition 1. Si notre RAID avait une grappe constituée de trois disques, /dev/md1 recouvrirait la petite partition de ces trois disques, et ainsi de suite. Ce que nous venons de dire s'applique évidemment de même pour /dev/md2 et le point de montage /home pour la partition 2

7 FAIRE CONNAISSANCE AVEC MDADM

mdadm est la commande système qui permet de tout faire en ce qui concerne le RAID. Elle permet de le créer, d'obtenir des informations très détaillées à son sujet et bien sûr de le reconstruire, c'est-à-dire le réparer. Très bizarrement, il est fréquent que cet outil ne fasse pas partie de l'installation par défaut des Linux chez les hébergeurs. Il est pertinent de l'installer maintenant, alors que vous êtes serein. Vérifiez sa présence :

mdadm -version

et en cas d'absence, installez-le avec le gestionnaire de paquets de la distribution Linux que vous utilisez. Cette installation ne présente pas de danger par elle-même. Ensuite, prenez le temps de consulter la page de man de mdadm, car son jeu d'options est riche et complexe. Nous pouvons utiliser mdadm pour obtenir des informations sur /dev/md1 et /dev/md2 bien plus riches que ce que nous avons obtenu jusqu'ici : [Fig.3].

```
# mdadm /dev/md1
```

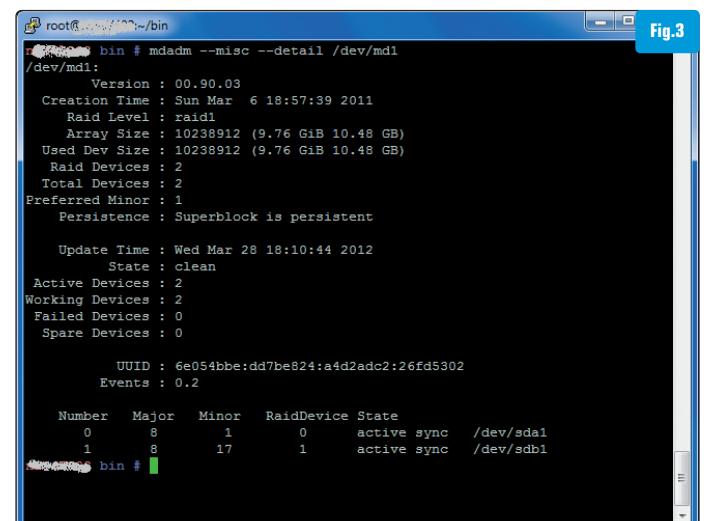


Fig.3

```
root@...:~# mdadm --misc --detail /dev/md1
/dev/md1:
      Version : 00.90.03
      Creation Time : Sun Mar  6 18:57:39 2011
      Raid Level : raid1
      Array Size : 10238912 (9.76 GiB 10.48 GB)
      Used Dev Size : 10238912 (9.76 GiB 10.48 GB)
      Raid Devices : 2
      Total Devices : 2
      Preferred Minor : 1
      Persistence : Superblock is persistent

      Update Time : Wed Mar 28 18:10:44 2012
      State : clean
      Active Devices : 2
      Working Devices : 2
      Failed Devices : 0
      Spare Devices : 0

      UUID : 6e054bbe:dd7be824:a4d2adc2:26fd5302
      Events : 0.2

      Number  Major  Minor  RaidDevice State
            0      8        1          0  active sync  /dev/sda1
            1      8        17         1  active sync  /dev/sdb1
```

La commande mdadm vous permettra de tout connaître sur votre RAID et aussi de le réparer.

```
/dev/md1:
  Version : 00.90.03
  Creation Time : Sun Mar  6 18:57:39 2011
  Raid Level : raid1
  Array Size : 10238912 (9.76 GiB 10.48 GB)
  Used Dev Size : 10238912 (9.76 GiB 10.48 GB)
  Raid Devices : 2
  Total Devices : 2
  Preferred Minor : 1
  Persistence : Superblock is persistent

  Update Time : Wed Mar 28 18:29:33 2012
  State : active
  Active Devices : 2
  Working Devices : 2
  Failed Devices : 0
  Spare Devices : 0

  UUID : 6e054bbe:dd7be824:a4d2adc2:26fd5302
  Events : 0.3
```

Number	Major	Minor	RaidDevice	State
0	8	1	0	active sync /dev/sda1
1	8	17	1	active sync /dev/sdb1

Nous avons ici confirmation que /dev/md1 recouvre bien la partition 1 de nos deux disques sda et sdb. Nous voyons bien que nous avons deux disques et qu'ils travaillent correctement. Supposons que le disque sdb soit en panne. Nous verrions :

```
# mdadm /dev/md1

/dev/md1:
  Version : 00.90.03
  Creation Time : Sun Mar  6 18:57:39 2011
  Raid Level : raid1
  Array Size : 10238912 (9.76 GiB 10.48 GB)
  Used Dev Size : 10238912 (9.76 GiB 10.48 GB)
  Raid Devices : 2
  Total Devices : 1
  Preferred Minor : 1
  Persistence : Superblock is persistent

  Update Time : Wed Mar 28 18:29:33 2012
  State : active, degraded
  Active Devices : 1
  Working Devices : 1
  Failed Devices : 0
  Spare Devices : 0

  UUID : 6e054bbe:dd7be824:a4d2adc2:26fd5302
  Events : 0.3
```

Number	Major	Minor	RaidDevice	State
0	8	1	0	active sync /dev/sda1
1	0	0	1	removed

Les lignes indiquant la panne ont été mises en gras ci-dessus.

8 L'OPÉRATION DE RECONSTRUCTION DU RAID

L'opération se déroule en deux temps. Formatage du disque qui vient d'être changé et reconstruction du RAID proprement dite. Nous supposons pour l'exemple que c'est le disque sdb qui a été remplacé et qui est donc vierge de toutes données et de tout formatage. En situation réelle, et quoi que vous ait dit votre hébergeur, vérifiez en premier lieu qu'il n'y a pas confusion dans les noms des disques avec un simple fdisk -l qui vous mettra en évidence quel disque n'est pas formaté. Maintenant que nous sommes sûrs que c'est bien de sdb dont nous devons nous occuper, nous devons le formater très exactement comme l'est sda. Pour cela on utilise l'outil sfdisk, comme ceci (attention, avec cet outil aucune faute de frappe n'est permise !)

```
sfdisk -d /dev/sda | sfdisk /dev/sdb
```

Par cette ligne de commande sfdisk dumpe toutes les informations de formatage du disque sda dans un format utilisable par lui-même, et nous lui envoyons ces informations dans un tube pour qu'il formate sdb. N'oubliez en aucun cas le commutateur -d ! Bien évidemment si c'est le disque sda qui a été remplacé, la commande devient sfdisk -d /dev/sdb | sfdisk /dev/sda. Il ne reste plus qu'à ajouter le nouveau disque au RAID comme ceci :

```
mdadm /dev/md1 --manage --add /dev/sdb1
```

A partir de ce moment la reconstruction commence et les données sont dupliquées, alors même que votre serveur tourne normalement. Vous pouvez admirer par exemple en donnant cette commande :

```
watch -n 3 cat /proc/mdstat
```

qui produira des sorties comme ceci :

```
Personalities : linear raid0 raid1 raid5
read_ahead 1024 sectors
md1 : active raid1 sdb12 sda10
3068288 blocks 2/1 U_
=>..... recovery = 5.4% (167028/3068288) finish
=0.8min speed=55676K/sec
md2 : active raid1 sda20
240597376 blocks 2/1 U_
```

Puis on n'oubliera pas de s'occuper du swap :

```
mkswap /dev/sdb
swapon -a
```

9 EST-CE TOUT ?

Non... vous savez l'essentiel sur le RAID et mdadm, mais il y aurait mieux à faire. Ainsi, si vous détectez vous même que votre RAID se dégrade, il serait pertinent de marquer le disque comme défaillant puis de retirer proprement le disque du RAID avant que votre hébergeur n'intervienne manu militari, avec cette succession de commandes :

```
mdadm --manage /dev/md0 --fail /dev/sdb1
mdadm --manage /dev/md0 --remove /dev/sdb1
mdadm --manage /dev/md1 --fail /dev/sdb2
mdadm --manage /dev/md1 --remove /dev/sdb2
```

Enfin un webmaster paranoïaque ne fera pas confiance au RAID seulement. Il fera une sauvegarde supplémentaire de ses données sur une autre machine, ce qui sort du cadre de cet article.

Frédéric Mazué - fmazue@programmez.com

Windows 8 et async/await

Grâce à `async/await`, l'écriture de code asynchrone devient un jeu d'enfant...à tel point que l'on oublie parfois d'y protéger l'accès aux ressources critiques. Ce genre d'oubli provoque des erreurs d'exécution totalement aléatoires dans les applications mais la bonne nouvelle c'est qu'il est très facile d'y remédier, surtout une fois que l'on a compris pourquoi ☺.

Un exemple typique dans les applications Windows 8 : l'accès aux fichiers. On les utilise à plus forte raison dans les applications Metro, par exemple pour :

- le fichier servant de cache off-line
- le fichier de sauvegarde du contexte courant de l'application
- le fichier contenant les données applicatives
- ...

Plusieurs clients m'ont remonté le fait que des exceptions sont déclenchées lors des écritures fichier, or le plus souvent, le problème provient de l'écriture du code asynchrone correspondant. Une démo vaut mieux que de longs discours, je vais donc illustrer ces propos pas à pas. D'abord, voyons le bout de code que je vais utiliser : un simple bouton dans l'AppBar qui déclenche un accès fichier en écriture.

```
private void Save_Click(object sender, RoutedEventArgs e)
{
    SaveAsync();
}

// Write data to a file
async void SaveAsync()
{
    StorageFolder localFolder = Windows.Storage.ApplicationData.
    Current.LocalFolder;

    var sampleFile = await localFolder.CreateFileAsync("dataFile.
    txt", CreationCollisionOption.OpenIfExists);
    var ws = await sampleFile.OpenStreamForWriteAsync();

    // Ecriture dans mon fichier, blablabla...

    ws.Dispose();
}
```

Au lancement de l'application, si l'on appuie plusieurs fois d'affilée sur le bouton "Save", en principe il n'y a pas de souci particulier car la méthode `SaveAsync` s'exécute dans un délai très court et a donc le temps de se terminer avant qu'on l'a ré-exécuté. A présent, simulons un accès fichier plus long (5 secondes) entre l'ouverture du fichier en écriture et sa fermeture :

```
// Write data to a file
async void SaveAsync()
{
    StorageFolder localFolder = Windows.Storage.ApplicationData.
    Current.LocalFolder;

    var sampleFile = await localFolder.CreateFileAsync("dataFile.
```

```
txt", CreationCollisionOption.OpenIfExists);

    var ws = await sampleFile.OpenStreamForWriteAsync();

    await Task.Delay(5000);
    ws.Dispose();
}
```

Cette fois, si je clique 2 fois sur le bouton en moins de 5 secondes, une exception sera levée. Pourquoi ? Parce que je dois garantir un accès exclusif au fichier ! Et pour l'instant ce n'est pas le cas.

1er réflexe : je désactive mon bouton le temps de réaliser la sauvegarde, sur l'evt click.

PROTECTION PAR L'UI

A votre avis, quelle est la différence entre ces 2 snippets :

Snippet 1

```
private void Save_Click(object sender, RoutedEventArgs e)
{
    btSave.IsEnabled = false;
    SaveAsync();
    btSave.IsEnabled = true;
}

// Write data to a file
async void SaveAsync()
{
    StorageFolder localFolder = Windows.Storage.ApplicationData.
    Current.LocalFolder;

    var sampleFile = await localFolder.CreateFileAsync("dataFile.
    txt", CreationCollisionOption.OpenIfExists);
    var ws = await sampleFile.OpenStreamForWriteAsync();

    await Task.Delay(5000);
    ws.Dispose();
}
```

Snippet 2

```
private void Save_Click(object sender, RoutedEventArgs e)
{
    SaveAsync();
}

// Write data to a file
```

```

async void SaveAsync()
{
    btSave.IsEnabled = false;

    StorageFolder localFolder = Windows.Storage.ApplicationData.Current.LocalFolder;

    var sampleFile = await localFolder.CreateFileAsync("dataFile.txt", CreationCollisionOption.OpenIfExists);
    var ws = await sampleFile.OpenStreamForWriteAsync();

    await Task.Delay(5000);
    ws.Dispose();

    btSave.IsEnabled = true;
}

```

On pourrait penser que l'exécution de ces snippets sera équivalente, mais ce n'est pas le cas. La commande `await` permet de synchroniser un appel asynchrone avec les lignes de code qui la suivent, mais il ne faut pas oublier que la méthode appelante – elle – ne sera pas bloquée, reprendra la main et continuera à s'exécuter. C'est d'ailleurs ce qui permet au thread de l'UI de ne pas être bloqué pendant ce temps. Cela signifie que l'on pourra déclencher plusieurs sauvegardes simultanées en appuyant sur le bouton "Save" dans le cas du 1er snippet, alors que dans le second, le bouton de sauvegarde sera désactivé tant que l'appel asynchrone ne se sera pas terminé. Ainsi le premier snippet est totalement inefficace pour protéger l'accès au fichier ! Donc, rappelez-vous que lors d'un appel asynchrone avec `async/await`, on retourne **immédiatement** dans la fonction appelante sans attendre la fin de l'appel asynchrone ! Le snippet 2 peut être suffisant dans de nombreux cas, mais il ne faut pas oublier que vous empêchez ici les accès concurrents au bouton `btSave`, et non pas les accès concurrents au fichier ! Très souvent, une sauvegarde intervient automatiquement sans que celle-ci soit commandée par l'UI : c'est d'autant plus le cas sur Windows 8 où le contexte courant est supposé être maintenu en « suspend », à la fermeture de l'application ou tout simplement régulièrement au fil de l'utilisation de l'application.

PROTECTION PAR LE CODE MÉTIER

Dans notre exemple nous avons couplé le code métier à l'UI en accédant au bouton `btSave` dans la méthode `SaveAsync`. Il est plus propre de protéger l'accès au fichier au niveau du code métier, tout en restant indépendant de l'UI c'est-à-dire directement dans la méthode `SaveAsync` mais sans interagir avec la Vue. Justement, dans mon exemple j'ai besoin de déclencher une sauvegarde automatique lors du passage dans l'état « suspend » de mon application. Ce nouvel état - nouvellement introduit pour les applications Win8 Metro - intervient 10 secondes après qu'une application passe en arrière-plan. Le système peut à tout moment arrêter une application qui se trouve dans l'état « suspend », s'il est en manque de ressources. Pour le confort de l'utilisateur, il est conseillé de sauvegarder le contexte applicatif lors du passage en « suspend » d'une application Metro, ce qui permettra de rétablir ce contexte en cas de besoin quand l'application sera relancée.

Dans notre cas, l'appel du `SaveAsync` sur le passage dans l'état « suspend » se fait à l'aide du code suivant :

```

public MainPage()
{
    this.InitializeComponent();
    App.Current.Suspending += Current_Suspending;
}

void Current_Suspending(object sender, Windows.ApplicationModel.SuspendingEventArgs e)
{
    SaveAsync();
}

```

Si vous lancez votre application, que vous cliquez sur le bouton "Save" puis passez immédiatement en état « suspend » grâce au bouton associé en mode debug dans Visual Studio, vous obtiendrez une erreur d'exécution. Pour y remédier, je vais protéger l'accès à ma ressource critique à l'aide d'un sémaphore : l'objet `SemaphoreSlim` qui permet de mettre en place un verrou asynchrone (comme un objet `Semaphore` mais que l'on peut attendre avec un `await`).

Snippet 3

```

SemaphoreSlim _semSave = new SemaphoreSlim(1);

private void Save_Click(object sender, RoutedEventArgs e)
{
    btSave.IsEnabled = false;
    SaveAsync();
    btSave.IsEnabled = true;
}

// Write data to a file
async void SaveAsync()
{
    await _semSave.WaitAsync();

    StorageFolder localFolder = Windows.Storage.ApplicationData.Current.LocalFolder;

    var sampleFile = await localFolder.CreateFileAsync("dataFile.txt", CreationCollisionOption.OpenIfExists);
    var ws = await sampleFile.OpenStreamForWriteAsync();

    await Task.Delay(5000);
    ws.Dispose();

    _semSave.Release();
}

```

Avec ce snippet, l'accès au fichier est protégé, quelle que soit l'origine de l'appel de la méthode `SaveAsync`. Le code de désactivation du bouton "Save" reste pertinent : il faut un feedback visuel permettant à l'utilisateur de savoir quand la sauvegarde est terminée. Mais elle n'est pas suffisante en tant que telle et demeure complémentaire à la protection de la ressource critique.

Stéphanie Hertrich - Microsoft

Twitter : @stepheMicrosoft - Blog : blogs.msdn.com/stephe

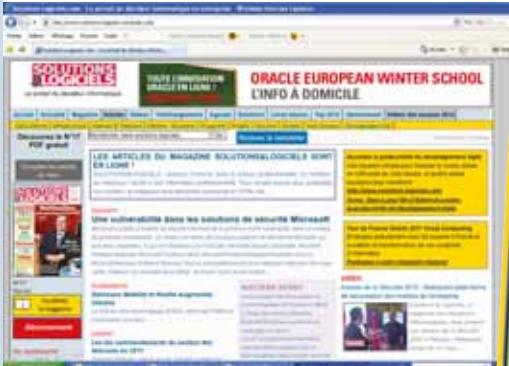
Les outils des Décideurs Informatiques

*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



LE MAGAZINE DES DÉCIDEURS INFORMATIQUES

SOLUTIONS & LOGICIELS **IT**

ETÉ 2012 N°31

www.solutions-logiciels.com

STOCKAGE UNIFIÉ Coûts réduits de 25% | **Data Centers** L'éco-système reste vital

OPEN SOURCE Au cœur des entreprises

Sécurité UTM Protéger les postes de travail itinérants

DEMAT Capture de documents

MANAGEMENT Formation des équipes Management au Féminin

STRATEGIE MAILEVA

Le Courier as a Service

Laurent Richardieu, Directeur Marketing et Communication

Maileva, leader et pionnier du courrier à la demande célèbre son 10^e anniversaire et lance de nouveaux services.

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 Boulanger 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

Optimiser et Mesurer la performance de vos applications C++ et Fortran avec



Intel® Parallel Studio XE 2013

Séminaire, Gratuit à Paris le 27 septembre 2012

microsigma.fr



Optimiser et Mesurer la performance de vos applications C/C++ et Fortran
avec la nouvelle suite Intel® Parallel Studio XE 2013

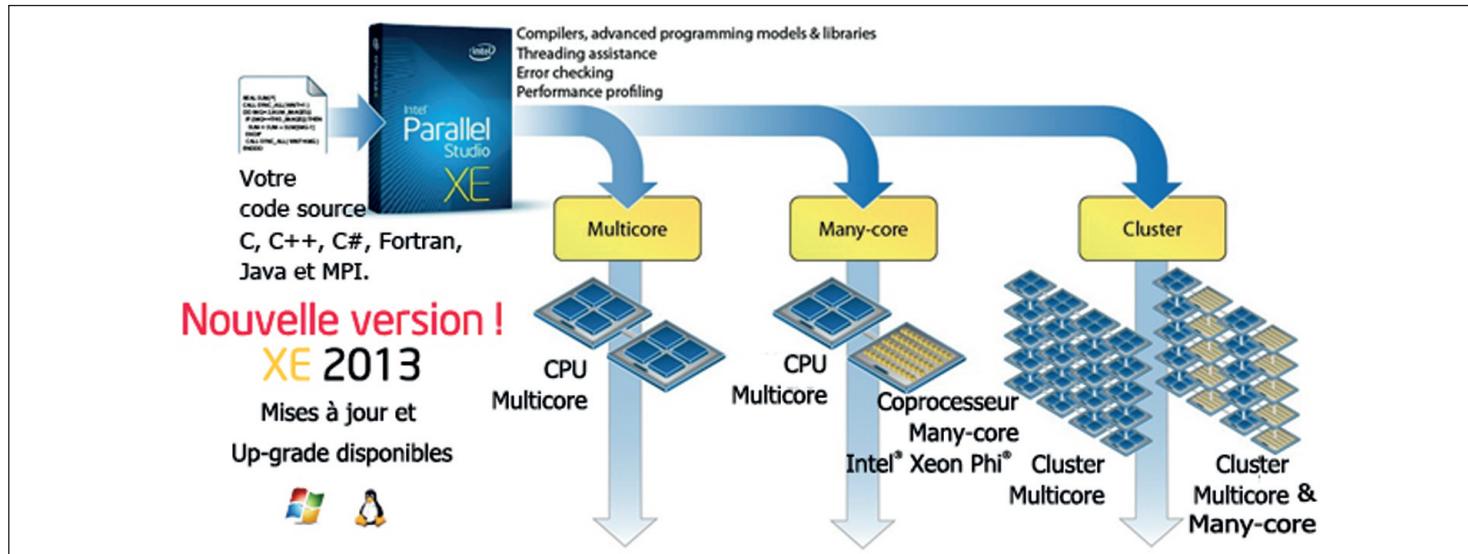
INVITATION

microsigma et Intel® vous invitent à un Séminaire Technique Gratuit

à Paris le 27 septembre 2012

**Invitation réservée aux Développeurs, Chefs de Projet, Architectes
Présentations techniques et démonstrations en Français.**

Renseignements et inscriptions : <http://www.microsigma.fr/intel/> ou par email à : info@microsigma.fr



Intel® Parallel Studio XE 2013 combine les meilleurs compilateurs et les bibliothèques de performance parallèle, les outils de vérification d'erreur et de profilage de performance pour le **C/C++** et le **Fortran**. Disponible pour **Windows** et **Linux**.

Destiné aux développeurs, cet ensemble d'outils indispensables comprend :

- Les meilleurs *compilateurs C, C++ et Fortran* du marché
- De nouvelles innovations de programmation,
- Des bibliothèques de haute performance, avec **Intel® Advisor** nouvel assistant au threading,
- La nouvelle génération d'outils de profilage et d'exactitude de code **Intel® Inspector XE 2013** et **Intel® VTune™ Amplifier XE 2013** profileur de performance et de thread pour développeurs **C, C++, C#, Fortran, Java** et **MPI**.

Renseignements et inscriptions : <http://www.microsigma.fr/intel/> ou par email à : info@microsigma.fr

microsigma

► N°Azur 0 810 120 240
PRIX APPEL LOCAL

Outils de développement &
logiciels techniques

www.microsigma.fr/intel/

