

PROgrammez !

mensuel n°152 - mai 2012

www.programmez.com

Etes-vous un développeur écologique ?



© i-stockphoto/ Ricardo Infante Alvarez

Des logiciels plus écologiques,
moins gourmands en ressources ? Oui c'est possible !

**Ayez les réflexes d'une
programmation éco-responsable**



Enquête

Les salair€s informatiques en 2012

Je débute avec...

Zend Framework
Drupal Commerce

Choisir son outil
pour bases
de données

UML

M 04319 - 152 - F: 5,95 €



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

**ActiveReports 6** à partir de € 521GrapeCity
PowerTools

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

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

**GdPicture.NET** à partir de € 2 384GdPicture
Imaging Technologies

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 € 1 863ASPOSE
The .NET & Java Component Publisher™

Tous les composants Aspose .NET réunit 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

**LEADTOOLS Document Imaging SDK V17.5** à partir de € 1 860LEAD
TECHNOLOGIES

Incluez des fonctionnalités puissantes d'imagerie aux applications Windows, Web et mobile.

- Modules complémentaires d'OCR, OCR asiatique et arabe, ICR, OMR et MICR
- Modules codes barre disponibles : codes barre 1D et 2D
- Modules PDF disponibles : Lecture/écriture raster PDF et PDF avancé
- Lecteurs d'images fonctionnels à encombrement zéro et annotations pour tout périphérique supportant HTML5

Blackberry a-t-il (encore) un avenir ?

Il y a des chiffres qui font peur. RIM évoque plus de 70 millions d'utilisateurs, 1 million de Playbook depuis son lancement il y a environ un an, une nouvelle gamme Blackberry 7.1 qui arrive, une pénétration du marché grand public par la messagerie instantanée, mais cela ne suffit pas.

La comparaison avec les chiffres d'Apple est violente : 37 millions d'iPhone sur les trois derniers mois de 2011, 3 millions du nouvel iPad en 3 jours ! C'est cruel mais c'est la réalité. RIM peut se dire que d'autres subissent le même sort, par exemple, Nokia qui ne profite pas encore de l'effet Windows Phone 7.x.

Le constructeur souffre de plusieurs problèmes : le virage tactile tardif, une tablette mal positionnée avec d'importantes lacunes, une stratégie en perte de vitesse.

RIM cherche donc à recentrer son activité (sans réellement le dire) sur son cœur métier, ce qui a fait son succès : l'entreprise. Cela passe par le renouvellement de la gamme avec l'arrivée de la famille 7.1 et la relance de sa tablette avec la Playbook 2 et la nouvelle version du système. Quid du grand public ? On doit s'attendre à un ralentissement des investissements et une sortie matérielle limitée. Dans le même temps, RIM investit fortement sur la partie administration des parcs mobiles avec la gamme Mobile Fusion qui supporte maintenant iOS



et Android. L'ouverture est l'un des maîtres mots du constructeur depuis quelques semaines.

L'ENJEU DU SYSTÈME BLACKBERRY OS 10

Initialement prévue pour 2011, la version 10 du système Blackberry ne devrait pas être disponible avant septembre / octobre. Cette v10 doit aussi marquer l'unification des systèmes mobiles smartphones et tablettes car RIM prévoit de l'installer sur sa Playbook. Elle s'appuie sur l'excellent système temps réel QNX. HTML 5, introduit du cloud computing, de la haute sécurité,

une interface 3D, et une refonte de l'interface. L'enjeu sera important pour RIM, surtout sur tablette, car si la v2 a comblé certaines lacunes, le résultat n'est pas encore à la hauteur des espoirs... RIM doit attirer les développeurs et les garder. La petite polémique autour de la fonction side-loading permettant de porter des applications Android sur Blackberry (voir encadré) laisse un petit air de flottement. Cependant, la plateforme garde un bon capital auprès des développeurs. A la conférence européenne DevCon, plus de 2000 personnes étaient présentes. Reste à savoir si RIM pourra encore jouer un rôle majeur dans l'industrie mobile...

Interview de Sanyu Kiruluta Team Lead, Developer Relations chez RIM

Un des arguments de la Playbook 2 est la possibilité d'utiliser des applications Android avec la fonction side-loading. Un jour, RIM dit que l'on veut supprimer la fonction, un autre, que l'on continue à la supporter. Qu'en est-il vraiment ?

Nous ne nous sommes pas débarrassés de la fonction side-loading sur Playbook OS ou sur Blackberry 10, mais elle change de nature. Il s'agit d'une fonction destinée aux développeurs pour leur permettre de charger les applications et de les tester sur leurs terminaux. Les développeurs peuvent ainsi envoyer des versions bêta pour les utilisateurs. Mais certains l'utilisent pour pira-

ter des applications. A partir de la prochaine version de Blackberry OS, nous allons crypter les applications et ainsi seules les applications achetées pourront s'exécuter. Nous travaillons activement avec la communauté pour aider les développeurs à tester les applications sur du matériel réel. C'est une des raisons pour lesquelles nous lançons Blackberry Jam en donnant à chaque participant un prototype matériel pour tester rapidement.

Blackberry OS 10 est une grande étape pour RIM et les développeurs. Pouvez-vous nous en dire plus ?

La plateforme Blackberry 10 est innovante

car elle permet l'intégration de nos récentes acquisitions. Blackberry 10 combine le meilleur de Blackberry et de QNX pour créer une nouvelle branche permettant de connecter, intégrer les utilisateurs, les contenus, les services et les terminaux. La plateforme intégrera de nombreuses technologies telles que TAT, Torch Mobile, Scoreloop. HTML 5 avec WebWorks et C++ avec le framework Cascades UI continueront à être au cœur du modèle de développement. D'autre part, les applications créées aujourd'hui pour la Playbook fonctionneront demain sur Blackberry 10, pareillement pour les applications utilisant notre solution HTML 5. Blackberry 10 supportera aussi Adobe AIR et le Blackberry Runtime pour les applications Android.



DEVELOPPEUR DU MOIS

Nabil Babaci : « Geek un jour,
Geek pour toujours. ».....28



BUZZWORD

Blackberry a-t-il (encore) un avenir ?.....3

ACTUS

En bref6

MATERIEL

Innorobo 2012.....10

Test : le WD Sentinel DX4000.....12

OUTILS

PHP 5.4 est-il le successeur de PHP 6 ?13

Outils UML pour les bases de données18

RESSOURCES

Livres et événements du mois30

Agenda31

CARRIERE

Emplois et rémunérations informatiques en 2012 ..44

Ces entreprises qui recrutent47



Wattmètre



VEILLE TECHNO

Etes-vous un développeur écologique ?

Eco-conceptions
des Logiciels et Green Patterns34

Devoteam :
une étude 2012 sur le GreenIT.....42



Juin 2012 - **n°153**, parution le 31 mai 2012

Un numéro à ne pas manquer !

⇒ Spécial Open Source

Créer son projet open source de A à Z

Open cloud : l'open source pose les fondations du cloud

Faire carrière dans l'open source : les filières, les jobs, les formations

Microsoft fait sa « révolution » open source

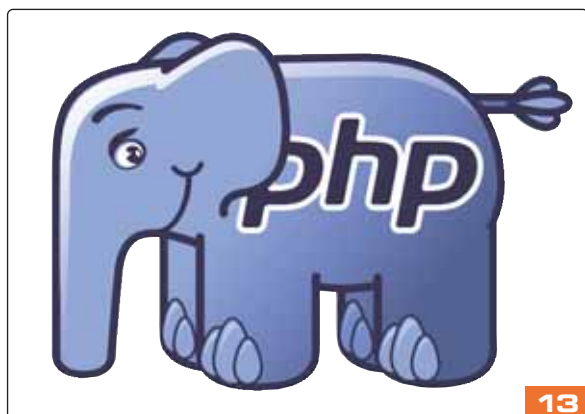
⇒ Linux et hébergement

Quoi de neuf ?



JE DÉBUTE AVEC

Introduction à la programmation GPU (4e partie)....	48
Windows 8, partie 2 : agilité avec TFS11.....	51
Premiers pas avec Zend Framework	54
Prise en main de Drupal Commerce	61



CAS D'USAGE

Programmation SOLID : une autre approche	65
Créer une liste téléphonique avec SQL Server Reporting Services (1 ^{ère} partie).....	70

PRATIQUE

Les dépendances en Javascript	76
-------------------------------------	----

WEBMASTER

Prise en main de WordPress	78
----------------------------------	----

BUGTRACK

Bug : « élément feed fermant manquant dans un flux OData »	82
---	----



Cet édito tu liras et appliqueras... ... sinon ton code source tu brûleras !

Il y a 40 ans, Gerald Weinberg a écrit une des références de la littérature informatique et particulièrement de la « culture développeur » : *The Psychology of Computing Programming*, (La psychologie de la programmation informatique). Il y évoque un concept primordial, la programmation « Egoless », sans ego. L'ego du développeur ne doit pas primer sur le reste, notamment sur le code en y laissant trop sa marque, rendant le code difficilement compréhensible par d'autres. Weinberg définit ainsi dix commandements de la programmation « sans ego », liste maintes fois reprise et adaptée :

- 1 Comprendre et accepter que vous ferez des erreurs : eh oui personne n'est infallible...
- 2 Vous n'êtes pas votre code : il faut comprendre que lorsqu'une revue de code est réalisée, des problèmes seront découverts. Il ne faut pas vous identifier au code et donc ne pas prendre pour vous (notion « égoïste » du code) les erreurs.
- 3 Peu importe le karaté que vous connaissez, il y aura toujours quelqu'un qui en saura plus que vous : apprenez de nouveaux gestes, affinez un geste grâce à un autre. C'est comme cela que vous serez meilleur, que vous apprendrez à vous perfectionner. Ne pas se dire que l'on connaît tout et que l'on n'a rien à apprendre des autres.
- 4 Ne pas réécrire un code sans consulter : la correction d'un bug, d'un problème de code ne doit pas entraîner plus de modifications que nécessaire.
- 5 Considérer, traiter les personnes connaissant moins que vous avec respect, déférence et patience : bref ne soyez pas arrogant, ne dites pas qu'ils ne connaissent rien, qu'ils ne comprennent rien... Ah le stéréotype du développeur et du geek asocial !
- 6 La seule constance dans le monde est le changement : garder un esprit ouvert, il faut accepter les nouveaux outils, les nouvelles méthodes. Le changement est dans la nature de l'informatique et de la programmation.
- 7 La seule vraie autorité vient du savoir, de la connaissance et non de sa position, de son poste
- 8 Battez-vous pour ce à quoi vous croyez mais acceptez la défaite... avec élégance. Bon là OK, précepte que je dois intégrer au skip-bo (mauvais joueur, moi ?)
- 9 Ne soyez pas le « gars de la pièce » : décidément le côté asocial nous poursuit ! Sortez de chez vous ! Discutez, partagez, rencontrez les autres.
- 10 Critiquez le code et non les personnes : être bon avec le codeur, et dur avec le code : ne dites plus « ce développeur est nul ». Proposez des solutions pour améliorer, corriger le code...

Bon, si après avoir suivi ces 10 commandements, vous êtes toujours un développeur égoïste, nombriliste et sectaire, j'abandonne !

François Tonic

Rédacteur en chef Programmez ! - ftonic@programmez.com

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

Rédaction : redaction@programmez.com

Directeur de la Rédaction : Jean Kaminsky. Rédacteur en Chef : François Tonic - ftonic@programmez.com. Ont collaboré à ce numéro : F. Mazué, Y. Grandmontagne, S. Saurel. Experts : S. Amar, F. Hardy, C. Soutou, O. Philippot, B. Boucard, P. Didiergeorges, V. Fabing, G. Bougenière, J.P. Gouigoux, M. Nison, F. Romelard.

Photos couverture : © i-stockphoto/Natalia Lukiyanova - © i-stockphoto/ Ricardo Infante Alvarez

Publicité : Régie publicitaire, K-Now sarl.

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

Directeur de la publication : J-C Vaudecrane

Ce numéro comporte un encart jeté Component Source sur une partie du tirage

Abonnement : Programmez, 17, Chemin des Boulangers, 78926 Yvelines Cedex 9 - Tél. : 01 55 56 70 55 - abonnements.programmez@groupe-gli.com - Fax : 01 55 56 70 91 - du lundi au jeudi de 9h30 à 12h30 et de 13h30 à 17h00, le vendredi de 9h00 à 12h00 et de 14h00 à 16h30. Tarifs abonnement (magazine seul) : 1 an - 11 numéros France métropolitaine : 49 € - Etudiant : 39 € - CEE et Suisse : 55,82 € - Algérie, Maroc, Tunisie : 59,89 € Canada : 68,36 € - Tom : 83,65 € - Dom : 66,82 € - Autres pays : nous consulter. PDF : 30 € (Monde Entier) souscription exclusivement sur www.programmez.com



» Microsoft passe les **API Bing Search** en mode payant !

Les développeurs devront donc passer à la caisse ! La transition est en préparation et les nouvelles API seront disponibles sur Windows Azure marketplace, qui accueille les API de Microsoft Translator. Le coût mensuel sera tout de même de 40 \$ pour 20 000 requêtes / mois.

Jusqu'à la transition complète, les développeurs sont invités à tester les nouvelles API et elles restent gratuites. D'autre part, les Bing Search API 2.0 ne seront plus utilisables gratuitement. L'éditeur annonce par ailleurs une amélioration des résultats. Enfin le modèle de développement va lui aussi subir quelques changements.

» **Microfocus** dévoile une nouvelle version

de son outil de modernisation des applications, de tests et de gestion : Caliber. Il arrive en version 10.1. Cet outil Borland aide les équipes à mieux travailler ensemble et à gérer les changements, la qualité logicielle, tout en ayant comme objectif la réduction des coûts. Caliber 10.1 propose une intégration fine entre CaliberRM et CaliberRDM.

» **Netbeans 7.1.1** est disponible.

Il s'agit d'une mise à jour mineure de l'IDE. Elle fournit le support à Java ME SDK 3.0.5, Glassfish 3.1.2, JavaFX 2.0.3, une nouvelle interface de debug pour JavaFX et Swing, des améliorations diverses sur Maven, PHP, Git, Subversion, etc. Tous les détails : <http://netbeans.org/community/releases/71/relnotes.html>

» **C** détrône **Java** au baromètre langage de **TIOBE**

qui mesure la popularité des langages. C arrive en tête avec 17,555 % contre 17,026 pour Java. Ce n'est pas la première fois que Java perd la tête du classement. C++ reste en 3e place. Belle montée en revanche pour Objective-C qui passe de la 8e à la 4e place.

Cette progression prouve la vigueur des plateformes OS X et iOS. Il devance assez largement C# et PHP qui perdent une place. Plusieurs langages connaissent une explosion de popularité : PL/SQL, VB.Net, NXT-G.

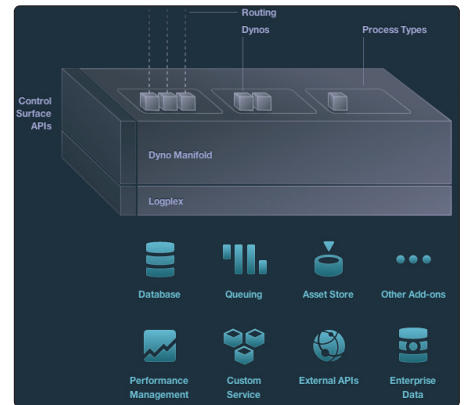
Mais aucun ne dépasse 1,5 % de popularité. Ce classement vaut ce qu'il vaut.

Tous les détails :

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

» **Salesforce** a voulu mettre en avant sa plateforme cloud : **Heroku**.

Lors de la journée Cloudforce du 10 avril dernier, l'éditeur organisait un dev garage pour force.com et surtout pour Heroku. Une occasion pour rencontrer quelques responsables Heroku et discuter avec d'autres utilisateurs de ce PaaS. Cela a permis de faire un point sur la plate-forme et comment on développe une application Heroku. C'est finalement assez simple : on s'inscrit, on installe les outils, on développe, on déploie... Cependant, Heroku souffre de plusieurs



défauts : pas de montée en charge automatisée comme sur Amazon, des datacenters uniquement aux Etats-Unis. Mais la plate-forme bénéficie d'un écosystème actif qui propose de nombreux add-ons pour compléter les fonctions ! Une grande conférence Heroku sera proposée à Londres le 22 mai prochain !

» **Oracle** a lancé le 10 avril dernier une pré-version de la version 5.6.x de son **SGBD** ouvert : **MySQL**.

Plusieurs nouveautés sont annoncées par l'éditeur :

- Des identifiants globaux de transactions (GTID) permettent de suivre l'avancement de la réplication sur une topologie de réplica-

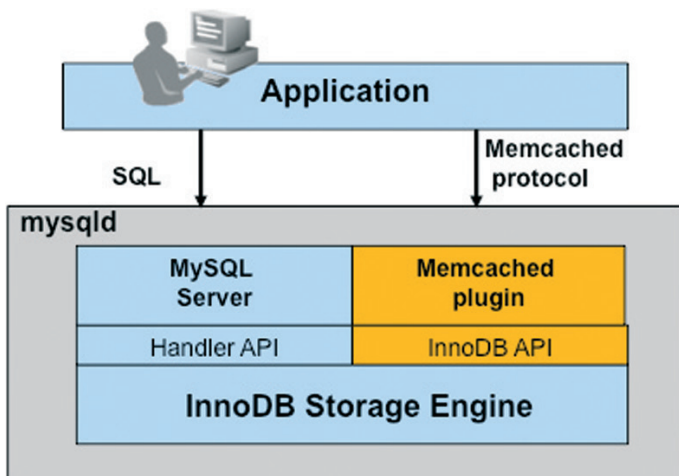
tion. Ils constituent la base pour fournir les mécanismes d'auto-réparation, et permettront de déployer des topologies de réplication plus complexes sans surcharge administrative.

- nouveaux utilitaires de réplication MySQL pour fournir la supervision avec des mécanismes de bascule et de reprise automatique, les GTID permettent d'éviter l'utilisation de solutions tierces de haute disponibilité, en protégeant les services Web et Cloud contre tout arrêt prévu ou imprévu.

- CURRENT_TIMESTAMP : désormais intégré dans l'optimiseur et utilisé comme valeur par défaut pour les colonnes DATETIME, il évite aux applications d'avoir à affecter cette valeur lorsqu'une valeur vide est utilisée par défaut.
- Sortie explain au format JSON

D'autre part, Oracle veut faire tester plusieurs fonctions sensibles : Add Index, NoSQL pour accéder aux données d'InnoDB, amélioration des performances pour les requêtes InnoDB et les contextes multi-cœurs.

Site officiel : <http://dev.mysql.com/>



Elever le niveau...

Modelez, dispersez et appliquez des fonctions spline... sur Business Intelligence (BI) mobile



Compatible avec
Microsoft® Visual
Studio® 11 Beta

+ de 50 façons différentes de Visualiser votre business intelligence et sur autant de périphériques possibles.

Du diagramme complexe au simple camembert, seul Infragistics offre une solution dynamique d'une telle ampleur. NetAdvantage est la solution graphique totalement mobile et de grande fiabilité.

Essayez gratuitement une évaluation entièrement prise en charge de NetAdvantage pour .NET dès aujourd'hui!

www.infragistics.com/.NET



Infragistics Sales US 800 231 8588 • Europe +44 (0) 800 298 9055 • India +91 80 4151 8042 • APAC (+61) 3 9982 4545

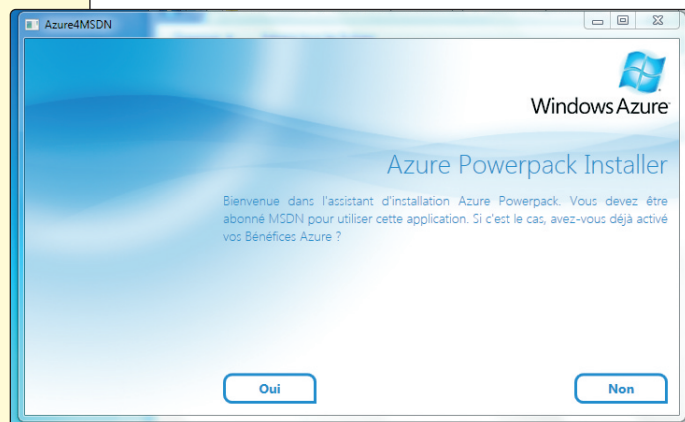
Copyright 1996-2012 Infragistics, Inc. All rights reserved. Infragistics and NetAdvantage are registered trademarks of Infragistics, Inc. The Infragistics logo is a trademark of Infragistics, Inc. All other trademarks or registered trademarks are the respective property of their owners.



» Microsoft propose des packages de services spécifiques à Windows Azure :

Powerpack Azure. Ce pack est pour le moment disponible uniquement pour les développeurs abonnés MSDN. Lors de sa sortie, trois services étaient proposés : créer rapidement jusqu'à 20 sites web personnalisés avec les noms de domaines, accès à un disque de stockage sur Azure et un accès à SQL Azure (sans que le firewall puisse bloquer les accès). Rapide à installer, ces services permettent surtout d'apprécier toute la puissance de Windows Azure et des fonctions que le développeur ne pense pas toujours à utiliser. D'autres services seront disponibles dans les prochains mois ! A découvrir :

<http://www.microsoft.com/france/communautes/offres/powerpack-azure.aspx>



» **Java 8 se précise.** Oracle a en effet dévoilé les dernières nouvelles sur l'agenda. Rappelons qu'Oracle avait découpé les fonctions entre Java 7 et 8 pour tenir les dates. Dans ce planning, Oracle annonce 6 versions préliminaires (milestones) entre le 24 avril 2012 et le 30 janvier 2013. La M6 devra figer l'ensemble des fonctionnalités.

A partir de là, une longue période de tests, de stabilisation, s'ouvrira. L'objectif est une sortie officielle de la JDK 8 pour septembre 2013. L'été sera l'occasion de terminer les détails et de fixer les bugs les plus importants.

» Un nouveau langage vient d'être normalisé par l'ISO : Ruby !

Oui vous avez bien lu. Le langage Ruby, populaire notamment grâce à Ruby on Rails, est officiellement une norme : ISO/IEC 30170 ! Elle définit strictement la syntaxe du langage. Ruby demeure un langage jeune (1995).

» **Yahoo a annoncé que son framework JavaScript (Node.js), Mojito,** passe en modèle open source. Pour mémoire, Mojito utilise le modèle MVC et permet de créer des widgets pouvant s'exécuter sur le serveur ou le poste client. Voilà une bonne nouvelle !

» Citrix fait évoluer son offre cloud computing.

Tout repose sur CloudStack qui devient le premier vrai projet cloud de la fondation Apache.

"Elever CloudStack au rang de projet open source Apache contribuera à accélérer l'adoption d'une plateforme puissante, éprouvée et totalement indépendante des hyperviseurs. Cela aidera ainsi les clients de toutes tailles à bâtir de véritables

nuages de type Amazon", indique l'éditeur. CloudStack sera proposé à la fois en open source et en version commerciale, cette dernière sera fournie par Citrix. CloudStack 3 est en licence Apache 2.

C'est une alternative à d'autres solutions cloud (openstack, red hat cloud, Hyper-V for cloud, etc.). Il s'agit d'un IaaS capable de faire tourner plusieurs hyperviseurs, l'administration se fait par une console web et supporte les API les plus répandues du marché (Amazon, vCloud, etc.).

» Mozilla travaille toujours à des solutions de vidéo sur le web

pour sortir de la situation de WebM (voir Programmez ! 151). La fondation met en avant popcorn.js., qui est la structure média HTML5 de Mozilla pour les cinéastes, les développeurs Web et

quiconque veut créer une vidéo éphémère sur le Web. D'usage plutôt simple, Popcorn.js est particulièrement destiné aux utilisateurs familiarisés avec JavaScript.

» Qt prépare activement la prochaine disponibilité de Qt 5.0.

Cette v5 inaugure une nouvelle génération du framework multiplateforme d'interface graphique en C++, sans rompre avec les codes de Qt 4.

Cette version apportera de nombreuses améliorations et nouveautés comme un support total d'OpenGL / OpenGL ES. Les premières versions bêta sont rendues publiques en ce moment (si tout va bien). La version finale est attendue pour fin juin. Le framework est disponible pour Windows, Linux, OS X. Pour en savoir plus :

<http://qt-project.org/wiki/Qt-5>



» Microsoft a dévoilé mi-avril dernier les différentes éditions de Windows 8

qui seront disponibles au lancement officiel du système cet automne. Trois éditions de Windows 8 seront proposées : Windows 8, Windows 8 Pro, Windows RT. A cela se rajoute une « version » spécifique entreprise, Windows 8 Enterprise. Elle sera disponible via la Software Assurance. Elle inclura toutes les fonctions de Windows 8 Pro et rajoutera des fonctions propres aux entreprises, comme

la gestion du PC, le déploiement, la virtualisation, la mobilité, etc.

Cette simplification de la gamme est la bienvenue et elle supprime toutes les éditions : familiale, starter, intégrale. Cela va faciliter la compréhension des utilisateurs. Windows 8 sera disponible en 32 et 64 bit en mode tactile et souris. L'éditeur annonce que toutes les versions auront le même niveau d'expérience utilisateur. La différence se situe surtout sur les fonctions disponibles.

Il est à noter que la version ARM est simplement nommée Windows RT (RT pour runtime, le nouveau moteur d'interface et d'API) pour éviter toute confusion. Il ne sera pas possible de migrer vers Windows RT depuis un Windows 7, ni d'installer des applications x86/64 ou des logiciels de bureau. Par contre, Windows RT inclura par défaut une version spécifique d'Office.



APPS FOUNDRY CONTEST

CODE, GRAPHIC AND UX DESIGN



17, 18 et 19 juin

CRÉEZ UNE APPLICATION MOBILE ORIGINALE ET INNOVANTE

RELEVEZ LE CHALLENGE : *soyez créatif et audacieux.*

Créez une application utile, ergonomique et au design unique

PRÉSENTEZ VOTRE APPLICATION AU FORMAT PECHAKUCHA DEVANT UN JURY

VIVEZ UNE EXPÉRIENCE UNIQUE ET BÉNÉFICIEZ D'UNE LARGE VISIBILITÉ !

GAINS :

1^{er} prix : un ordinateur portable pour chaque membre de l'équipe.

2^{ème} prix : une tablette pour chaque membre de l'équipe.

3^{ème} prix : un lecteur MP3 pour chaque membre de l'équipe.



OFFRES SPÉCIALES

Pour les 8 premiers inscrits, prise en charge des nuits d'hôtel du 17 et 18 juin.

TARIFS :

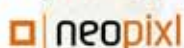
En équipe : 200€ HT

En individuel : 50€ HT

Gratuit pour les étudiants !

Découvrez toutes les opportunités de sponsoring, le programme complet, et inscrivez-vous sur www.apps-foundry-contest.org

Organisé par



Contacts :

contact@apps-foundry-contest.org

Sponsorisé par



LUXGSM

Avec le soutien de

jamendo



Innorobo 2012 : l'événement robotique français !

Les 14, 15 & 16 mars dernier, s'est tenue la seconde édition du salon de la robotique « Innorobo », c'est pour moi l'occasion de vous faire partager toutes les nouveautés présentées.

Comme l'année dernière nous avons profité d'une succession de conférences qui permettaient de faire un état des lieux de la robotique en France et dans le monde, d'un centre d'exposition avec 104 exposants et de nombreuses nouveautés, et cerise sur le gâteau, de la compétition Vex Robotic, le dernier jour du show.

> Les conférences

De nombreuses sessions étaient proposées cette année : Marché synergies et convergences : Robotique industrielle et de service, Cobotique : du robot assistant au robot coéquipier, Interaction Homme-robot, Cloud Robotics, robotique médicale et de santé, robotique urbaine : les solutions de mobilité. Vous trouverez sur mon blog les principales slides... De nombreux exposants étaient présents, preuve du dynamisme de l'événement. Nous en avons retenu quelques-uns :

Microsoft, toujours actif

Une organisation sans faille de Pierre Cauchois. Résultat : le stand n'a pas désempli durant les trois jours. Et pour cause, il y avait de nombreux produits et matériels : Eddie (la plateforme de développement de Microsoft Robotics), Nao sous .Net, du Cloud, du Windows Phone, du Kinect.

Génération Robot

A proprement parler, on ne fabrique pas de robot chez Génération Robot, en revanche on en vend et on développe autour de la robotique. Sur leur stand on retrouvera un nouveau module Wifi pour la brique MindStorm, Nao et son talent pour le « puissance 4 »,



Kinova



Hercule



Kibo

Eddie, etc. Je vous invite vraiment à vous rendre sur leur site pour l'achat de vos produits de robotique, l'équipe est ultra réactive (<http://www.generationrobots.com>) !

Aldebaran

Cette année, on n'est pas venu en force chez Aldebaran, en revanche le stand est plus accueillant, des canapés partout. On ne parle pas seulement de Nao mais aussi du Developer Program. A ce propos, Aldebaran a organisé un coda-thon pour mettre en avant le développement d'applications pour le robot, le week-end du 1er Avril, 3 jours mélangeant conférences & programmation sur Nao et surtout un concours d'applications dont le premier Prix était un Nao.



Induct

Déjà là l'année dernière, Induct présente de nouveau son véhicule de transport de personnes automatisé. L'année dernière c'était sur son stand, cette année c'était dehors, avec transport de visiteurs.

Kist

(Korea Institute of Science and Technology) Kibo. Le robot Humanoïde, la star du salon cette année. Première apparition pour ce robot hors de ses frontières natales. Il mesure 1m20 et pèse 43 kilos, il se déplace, danse, parle, chante, et est doté d'un visage ultra expressif. Bref, on adore !

Kinova

On a tous eu l'occasion de voir des bras robotisés. Mais de là à atteindre la précision des bras de Kinova ! Sur le stand, pour prouver l'efficacité de ces appareils, on vous propose de les piloter pour vous servir de l'eau avec une bouteille en plastique ! Pour l'avoir essayé je me suis dit la bouteille va être broyée, mais non. C'est précis et souple, génial !

RB3D

Sur ce stand nous avons un aperçu du futur de la robotique d'assistance. Le projet Hercule est un exosquelette qui permet à l'opérateur de porter une charge allant jusqu'à 100 kg, et de se déplacer à une vitesse de 4 km/h pour 20 km d'autonomie. Cet été il devrait recevoir les bras mécatroniques qui lui permettront également de manipuler des objets.

> Alors au final, on en a pensé quoi ?

C'était une cuvée exceptionnelle, nous avons eu droit à « la totale », une organisation top, des stands avec des robots d'un niveau de finition très élevé, des conférences particulièrement intéressantes. Si la robotique vous passionne, c'est l'endroit où il faut être. Alors rendez-vous l'année prochaine pour l'édition 2013, gardez-vous 3 jours de congés pour le mois de mars 😊

Salah Amer

Architecte logiciel expert sur les technologies .Net au sein du groupe One Point
<http://www.proteus-groupe.com/>

solutions
LiNux
Open Source



Le salon européen dédié à Linux
et aux logiciels libres

Toutes les solutions et nouveautés informatiques en Open Source...
Pour encore plus de libre au service de l'entreprise !

19-20-21
JUIN
2012

CNIT - Paris La Défense



Business Intelligence - Cloud Computing - Clustering & Grid - CMS - Collaboratif - CRM - Data Center - Développement - E-Commerce - ERP - Infrastructures - Innovation
Intéropérabilité - Mobilité - Network Management - Poste de Travail - Sécurité - SOA & Web Services - SGDB - Temps Réel & Embarqué - Virtualisation - VoIP

TENUE CONJOINTE
↑
LES ASSISES
← DU LIBRE ET DE
↓ L'OPEN SOURCE
NOUVEAUTÉ 2012

Un événement

Partenaire officiel

Tarsus
GROUPE MEDIA R10 &

monANNUAIRE
pro.com

www.solutionslinux.fr

Notre test :

WD Sentinel DX4000

Le nouveau NAS du constructeur WD est disponible. Le Sentinel DX4000 cible les petites entreprises et les utilisateurs exigeants. Le DX4000 expose un boîtier assez massif et très classique. Il propose 4 baies pour des disques 3.5. Sa capacité maximale est de 12 To ! Côté connectique, il propose deux ports Ethernet 1 Gb et deux USB 3 (dommage pour le Firewire 800 et le Thunderbolt).

Le NAS fonctionne avec un système interne Windows Storage Server 2008 R2 Essentials mais il est compatible, en usage, avec Windows et OS X. Il oblige cependant à le connecter en

premier à un PC Windows avant de pouvoir l'utiliser sur 25 machines maximum (Linux, Mac, Windows). Sur la partie sauvegarde, le DX4000 propose, par défaut, du RAID 1 mais monte sans difficulté au RAID 5. Pour éviter toute coupure d'alimentation, il possède une double alimentation pour deux alimentations séparées (1 seule livrée dans le carton). L'administration se fait par un service client et via la console web. Bien entendu il est multi-protocole (FTS, webdav, SMB, etc.). Surtout, il permet le changement à chaud des disques, ce qui est très pratique quand on ne peut arrêter le NAS. Cependant,



attention, le DX4000 assurera alors la récupération du système et des fichiers, ce qui prend beaucoup de temps. Pour la sauvegarde, des fonctions de sauvegarde automatique sont proposées (et plutôt efficaces) et une restauration ciblée ou complète. L'accès distant est proposé par défaut. A noter, la présence du verrou de sécurité

Kensington pour un câble antivol. WD propose aussi son service WD Guardian avec des options Express et Pro pour une assistance sur site, sans oublier un service de sauvegarde en ligne. Prix : environ 900 euros HT pour la version de base.

Pour en savoir plus : <http://wdc.com/fr/products/products.aspx?id=610>

en Bref

en Bref

en Bref

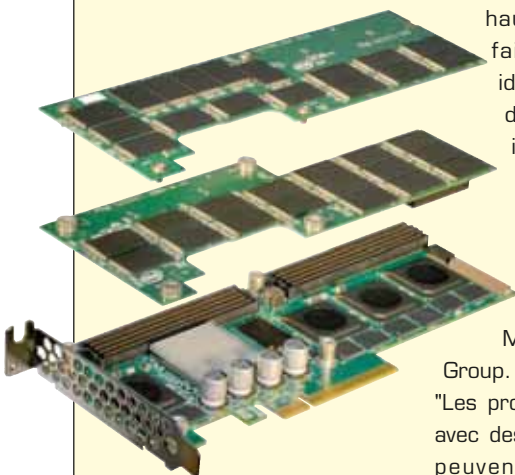
en Bref

» **Intel dévoile sa nouvelle gamme SSD** : SSD série 910. Le constructeur propose des disques de 400 et 800 Go ! Surtout, Intel cherche à

prix) du SSD. "L'Intel SSD 910 Series est une solution complète qui répond à la demande sur tous les fronts: une haute performance, une haute endurance, et une haute fiabilité; cela en fait le remplacement idéal pour les disques durs à usage I/O intensif," selon Roger Peene, directeur du marketing des SSD pour centres de données au sein de l'Intel Non-Volatile Memory Solutions Group.

"Les professionnels travaillant avec des centres de données peuvent faire confiance à cette solution SSD haut de gamme et facile à installer, qui est soutenue par Intel, leader en matière de qualité, fiabilité et support technique pour les technologies SSD". SSD série 910 utilise la technologie PCIe.

accroître la durée de vie des SSD et les performances en annonçant : 10 écritures complètes du disque par jour pendant 5 ans, soit une endurance multipliée par 30. Actuellement, c'est le principal point faible (avec le



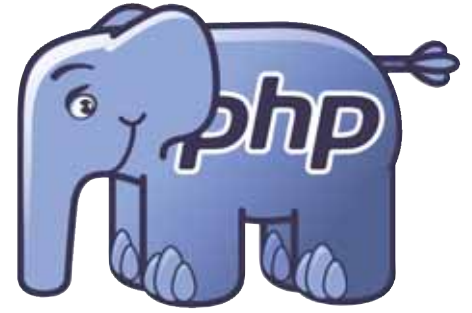
» **LINDY lance un convertisseur USB audio** numérique/analogique. Simple et pratique, cet accessoire permet aux mélomanes et aux cinéphiles de profiter, en qualité audio élevée jusqu'à 24 bits/192kHz, de toutes leurs musiques et de tous leurs films stockés en format compressé (mp3, wav, AAC...). Prix : 79 €

» **Synology lance les serveurs RackStation** RS2212+/RS2212RP+, compatibles RAID5. Conçus pour assurer une capacité d'extension idéale aux utilisateurs des PME, les deux modèles offrent une solution de stockage avec un total de 22 baies qui s'intègre aux unités d'extension Synology RX1211(RP), la mémoire RAM du système pouvant passer de 1 à 3 Go pour répondre aux besoins exigeants d'une entreprise en croissance. A partir de 960 € H.T.

» **ChromeOS va-t-il pouvoir redynamiser les Chromebook ?** Une nouvelle version du système proposera une expérience utilisateur plus proche du système d'exploitation classique. Google avait voulu la rupture mais il a constaté lui-même l'échec de l'approche. Dans le même temps, les Chromebook devraient eux aussi avoir des nouveautés notamment sur la suppression du BIOS plus moderne et performant, on parle de coreboot. A suivre...

» **Kingston Digital continue à miser sur les clés USB** et particulièrement sur USB 3 avec DT Elite 3.0. Cette clé est rétro compatible avec l'USB 2.0. Disponible en 16Go, 32Go, et 64Go, la DataTraveler Elite 3.0 est élégante, sans bouchon et rétractable pour protéger le connecteur USB. Elle peut être facilement accrochée à un porte-clés ou lanière, permettant aux utilisateurs de garder la clé sur eux à tout moment.

PHP 5.4 est-il le successeur de PHP 6 ?



Il y a deux ans, les développeurs de PHP ont abandonné le développement de PHP 6 à cause des difficultés qu'ils ont rencontré pour y intégrer de manière simple et efficace le support d'Unicode.

Cet événement aurait pu signer l'arrêt de mort du langage, mais il a au contraire insufflé un nouvel élan à son développement qui a abouti le mois dernier à la sortie de PHP 5.4.

En effet, suite à l'arrêt du développement de PHP 6, la communauté des développeurs du langage s'est remise en question, ce qui a abouti à des modifications profondes et des évolutions significatives dans son fonctionnement. Ainsi, des fonctionnalités réclamées depuis longtemps par les utilisateurs du langage ont enfin pu être développées et intégrées dans PHP 5.4.

À contrario, pour la première fois dans l'histoire de PHP, des fonctionnalités reconnues comme étant des « boulets historiques » du langage ont été supprimées. De plus, cette version contient, outre des développements originaux, des optimisations et des améliorations développées pour PHP 5 mais dont l'intégration avait été reportée dans l'attente de la sortie de PHP 6.

Au final, PHP 5.4 est l'œuvre de plus de 70 contributeurs qui ont effectué, en approximativement 700 jours, plus de 3400 modifications sur le code de PHP 5.3 ayant servi de base à son développement. Il est donc impossible d'en faire une présentation détaillée et exhaustive dans le cadre de cet article et je me contenterai donc d'en décrire les aspects les plus significatifs, à commencer par les fonctionnalités dont il ne dispose plus.

MIEUX PARTAGER

Depuis sa version 4, PHP dispose de trois fonctionnalités, connues respectivement sous le nom de « magic quotes », de « register globals » et de « safe mode », qui sont toutes trois censées en théorie aider le développeur ou le protéger de ses erreurs, mais qui en pratique présentent de nombreux inconvénients.

Destinées à empêcher l'injection ou l'exécution de code malicieux, les « magic quotes » dégradent les performances, s'appliquent automatiquement et sans discrimination et rendent la portabilité du code délicate car leur disponibilité dépend de la configuration du langage. Les « register globals » permettent quant à elles l'importation automatique et sans contrôle de données externes, donc contenant potentiellement du code malicieux, dans des variables globales et leur utilisation a été par le passé à l'origine de nombreuses failles de sécurité parmi plusieurs projets PHP populaires.

Enfin, le « safe mode » désactive quant à lui les fonctionnalités du langage pouvant être utilisées ou détournées à des fins malveillantes, mais il ne protège absolument pas de l'ensemble des formes d'attaques possibles et n'est donc pas une bonne solution pour obtenir un code sécurisé. Malgré leurs inconvénients, les développeurs du langage s'étaient toujours refusé à supprimer ces trois fonctionnalités, afin de garantir la sacro-sainte compatibilité de chaque nouvelle version de PHP avec le code existant, même si elle avait été marquée comme étant dépréciée depuis PHP 5.3.

Cette politique est cependant aujourd'hui révolue puisqu'ils se sont

enfin résolus à couper le cordon avec le passé en supprimant ces trois fonctionnalités de PHP 5.4. De plus, ils ont également fait le ménage au niveau des directives de configuration, puisqu'en plus des directives relatives aux « magic quotes », à « register globals » et au « safe mode », ils ont également supprimé des directives obsolètes telles que `y2k_compliance` relative au fameux « bug de l'an 2000 », `zend.zel_compatibility_mode` qui active le mode de compatibilité avec PHP 4 ou encore `allow_call_time_pass_reference` qui autorise le passage explicite d'argument par référence.

La suppression de ces deux dernières directives est d'ailleurs également très symbolique de la volonté des développeurs du langage de faire table rase du passé puisqu'elle a pour conséquence de ne plus pouvoir faire exécuter par PHP 5.4 du code écrit pour PHP 4.

Pour autant, PHP 5.4 n'est pas le successeur de PHP 6 car il ne supporte pas mieux Unicode que PHP 5.3.

En effet, suite à l'abandon du développement de PHP 6, les développeurs de PHP ont en effet décidé de se donner le temps de la réflexion pour trouver un moyen d'intégrer efficacement Unicode au sein du langage. Malheureusement, ce chantier n'a depuis que peu, voire pas du tout progressé, et PHP 5.4 n'apporte donc aucune nouveauté à ce niveau, hormis des ajustements mineurs notamment parmi les directives de configuration.

Ainsi, la valeur par défaut de la directive `default_charset` est maintenant `UTF-8`, et PHP 5.4 considère donc maintenant que les données à envoyer à un client HTTP sont par défaut encodées dans ce format et non plus en `iso-8859-1`.

Les nouvelles directives `zend.script_encoding` et `zend.multibyte` permettent, quant à elles, respectivement de définir le format par défaut du code source et d'en activer la détection automatique.

Cependant, l'encodage et la manipulation des chaînes de caractères au format Unicode restent de la responsabilité du programmeur et doivent toujours se faire à l'aide des extensions `mbstring` ou `intl`.

Pour compenser cela, les développeurs de PHP 5.4 ont ajouté plusieurs nouvelles fonctionnalités, à commencer par du « sucre syntaxique » qui permet au programmeur d'avoir à écrire moins de code pour utiliser certaines fonctionnalités du langage. Ainsi, PHP 5.4 permet d'accéder directement à un élément d'un tableau retourné par une fonction sans devoir passer par une variable intermédiaire.

```
<?php
// Avant PHP 5.4
$words = explode(' ', 'Frédéric Hardy');
echo $words[1];

// Depuis PHP 5.4
echo explode(' ', 'Frédéric Hardy')[1];
```

Dans le même style, il autorise également l'accès à un membre d'une

classe directement à partir de l'appel à son constructeur, donc également sans avoir besoin de passer par une variable intermédiaire.

```
<?php
class author
{
    protected $name = null;
    public function __construct($name) {
        $this->name = $name;
    }
    public function getName() {
        return $this->name;
    }
}
echo (new author('Frédéric Hardy'))->getName();
```

Il est également possible de définir un tableau à l'aide d'une syntaxe similaire à la notation [JSON](#).

```
<?php
// Avant PHP 5.4
$nouveautés = array('traits', 'sucre syntaxique', 'serveur HTTP');
$representations = array('decimal' => '', 'octal' => '0', 'hexa
decimal' => '0x', 'binaire' => '0b');

// Depuis PHP 5.4
$nouveautés = [ 'traits', 'sucre syntaxique', 'serveur HTTP' ];
$representations = [ 'decimal' => '', 'octal' => '0', 'hexa
decimal' => '0x', 'binaire' => '0b' ];
```

PHP 5.4 autorise également maintenant le développeur à appeler une méthode de classe statique à l'aide d'une expression.

```
<?php
class author
{
    public static function getBook() {
        return 'Programmez!';
    }
}
$method = 'Book';
echo author::{'get' . $method}();
```

Ce « sucre syntaxique » est donc agréable à l'usage, mais il ne faut cependant pas en abuser sous peine d'indigestion. En effet, il permet d'obtenir très facilement un code difficilement lisible et donc délicat à maintenir.

```
<?php
class author
{
    protected $name = null;
    public function writeFor($book) {
        echo $book;
    }
    public static function getBooks() {
        return [ 'PHP 5 avancé', 'Programmez!' ];
    }
}
```

```
$method = 'Books';
echo (new author('Frédéric Hardy'))->writeFor(author::{'get' .
$method}())[1];
```

D'autres modifications ont été effectuées au sein de PHP 5.4 afin d'améliorer le confort du développeur, à commencer par la possibilité d'utiliser la syntaxe `<?= $var ?>`, équivalente à `<?php echo $var ?>`, indépendamment de la valeur de la directive de configuration [short_open_tags](#), ce qui permet au développeur d'éviter d'avoir à se soucier de la portabilité de son code lorsqu'il utilise cette fonctionnalité. Toujours dans la lignée des évolutions effectuées pour rendre la vie du développeur plus confortable, la console de PHP, accessible à l'aide de l'argument `-a` en ligne de commande, est enfin devenue réellement utilisable avec PHP 5.4.

En effet, une erreur fatale ne provoque plus l'arrêt de la console et il est possible de définir l'aspect de son invite de commande, l'utilitaire devant être utilisé pour afficher le flux de sortie par écran successif et d'activer le support de la couleur, respectivement via les directives de configuration [cli.prompt](#), [cli.pager](#) et [cli_server.color](#).

De plus, il est possible de modifier la configuration de PHP directement dans la console à l'aide de la syntaxe `#directive=valeur`, mais uniquement pour la console et tant qu'elle est utilisée.

ALTERNATIVE À L'HÉRITAGE

PHP 5.4 apporte également un certain nombre de nouveautés fonctionnelles, à commencer par la possibilité d'indiquer qu'un argument de fonction ou de méthode doit obligatoirement être une fonction de rappel, grâce au mot-clé [callable](#).

Il n'est donc plus possible de l'utiliser comme nom de variable, de fonction, de classe ou de méthode, et il en est de même pour [trait](#) et [insteadof](#) qui font partie de la syntaxe servant à définir et utiliser un trait. Les traits sont considérés par beaucoup comme la nouveauté majeure de PHP 5.4 et sont une alternative à l'héritage multiple que l'on trouve dans d'autres langages orientés objet tels que le C++. Concrètement, un trait se présente comme une interface, mais contrairement aux méthodes d'une interface, les méthodes d'un trait disposent d'une implémentation.

```
trait cURL
{
    public function fetchUrl($url) {
        $curl = curl_init();
        curl_setopt($curl, CURLOPT_URL, $url);
        curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
        $output = curl_exec($curl);
        curl_close($curl);
        return $output;
    }
}
```

Une fois défini, un trait peut alors être utilisé dans le corps de chaque classe ayant besoin d'utiliser une ou plusieurs de ses méthodes à l'aide du mot-clé [use](#).

```
class twitter
{
    use cURL;
    public function get($url)
```

```
{
    return json_decode(
$this->fetchUrl(
'http://api.twitter.com' . $url
)
);
}
```

Lors de la compilation de la classe par le moteur de PHP, les méthodes du trait sont « copiées/collées » du trait vers la classe et peuvent donc être utilisées par les instances de la classe comme si elles avaient été codées au sein de la classe.

```
<?php
echo (new twitter())
->fetchUrl('http://www.programmez.com');
```

Cependant, il peut arriver qu'une classe ait besoin d'utiliser un trait mais que l'une de ses méthodes porte un nom identique à celui d'une méthode de la classe. Il est alors de la responsabilité du développeur de résoudre le conflit à l'aide des mots-clés `as` et `insteadof`.

```
class twitter
{
    use CURL { CURL::fetchUrl as fetchUrlWithCurl; }
    public function fetchUrl($url)
    {
        return json_decode(
$this->fetchUrlWithCurl(
'http://api.twitter.com' . $url
)
);
}
}
```

Les traits ne sont cependant pas l'unique nouveauté significative de PHP 5.4, puisqu'à partir de cette version, le langage dispose d'un serveur HTTP intégré accessible via la ligne de commande et l'argument `-S`.

```
# php -S 127.0.0.1:8080
```

Grâce à son serveur de base de données intégré SQLite et ce serveur HTTP, PHP dispense donc le développeur d'installer sur son poste de travail un serveur de base de données et un serveur HTTP en plus du langage, pour qu'il puisse commencer à travailler.

Attention cependant, ce serveur HTTP est destiné à être utilisé exclusivement en phase de développement et n'est donc pas suffisamment sécurisé pour être mis en œuvre dans un environnement de production. Le support des fonctions anonymes évolue également avec PHP 5.4 puisqu'il permet d'utiliser `$this` dans une fonction anonyme déclarée au sein d'une méthode de classe.

```
<?php
class author
```

Les rendez-vous de l'open source

**15
MAI**

Gestion documentaire : panorama des meilleurs outils open source

Séminaire Smile
Nantes - de 09:00 à 12:00

**22
MAI**

e-Commerce open source : le Choc des Titans

Séminaire Smile
Paris / Saint-Lazare - de 09:00 à 12:00

**29
MAI**

Bargento 2012

Le salon dédié à Magento
Paris - de 09:00 à 18:00

**05
JUIN**

Dictionnel open source : quel outil choisir pour mon projet ?

Séminaire Smile
Lyon - de 09:00 à 12:00

**05
JUIN**

Communication Mobile : quelle stratégie adopter ?

Séminaire Smile
Paris / Saint-Lazare - de 09:00 à 12:00

**14
JUIN**

CMSday, la rencontre du meilleur des CMS open source

Organisé par Smile, le CMSday est le 1^{er} événement majeur en France et en Europe dédié à la gestion de contenu open source
Paris - de 09:00 à 18:00



Pour vous inscrire
gratuitement,

www.smile.fr ou utilisez ce QR code


```
{
    protected $name = null;
    public function __construct($name) {
        $this->name = $name;
    }
    public function getClosure() {
        return function($book) { return sprintf('%s écrit pour %s',
$this->name, $book); };
    }
}
$closure = (new author('Frédéric Hardy'))->getClosure();
echo $closure('Programmez!');
```

De plus, il est possible de « transférer » une fonction anonyme rattachée à un objet, à un autre objet, grâce aux méthodes `\Closure::bind()` et `\Closure::bindTo()`.

```
<?php
$otherClosure = $closure->bindTo(
    new author('Christophe Villeneuve')
);
echo $otherClosure('Programmez!');
```

Dans un autre registre, PHP permettait déjà de définir un nombre via sa représentation décimale, octale ou hexadécimale. PHP 5.4 complète cette tripléte avec la possibilité de le définir via sa représentation binaire à l'aide du préfixe `0b`.

```
<?php
echo 31; // decimal
echo 037; // octal
echo 0x1f; // hexadecimal
echo 0b11111; // binaire, uniquement avec PHP 5.4
```

PHP 5.4 dispose également d'une fonctionnalité attendue depuis très longtemps, à savoir la possibilité de gérer une barre de progression d'envoi d'un fichier à partir d'un client vers le serveur.

À cette fin, six directives de configuration ont été ajoutées au langage, à savoir :

`session.upload_progress.enabled`, `session.upload_progress.cleanup`, `session.upload_progress.prefix`, `session.upload_progress.name`, `session.upload_progress.freq` et `session.upload_progress.min_freq`, qui permettent de contrôler la façon dont PHP peuple la variable `$_SESSION` avec les informations relatives à la progression de l'envoi du fichier.

Une fois les valeurs de ces directives définies, il est alors possible de récupérer ces informations à partir du client HTTP à l'aide d'une requête AJAX et de mettre ainsi à jour la barre de progression correspondante.

DU NOUVEAU SUR LA SÉCURITÉ

La sécurité a été également l'une des préoccupations des développeurs de PHP 5.4 puisque cette version introduit une nouvelle directive de configuration nommée `max_input_vars`, destinée à rendre inopérantes les attaques de type HashDOS.

Ces attaques exploitent en effet certaines caractéristiques de l'algorithme utilisé par PHP pour gérer ses tableaux associatifs afin de rendre le serveur visé indisponible en surchargeant son processeur.

En permettant de définir le nombre maximum de variables pouvant être contenues dans les tableaux `$_GET`, `$_POST` et `$_COOKIE`, la directive `max_input_vars` permet de limiter les effets de ces attaques. Plusieurs nouvelles fonctionnalités ont donc été ajoutées à PHP 5.4 mais cela n'a pas pour autant été fait au détriment de ses performances.

En effet, grâce à un gros travail d'optimisation, il peut se révéler être de 15 à 50% plus rapide et avoir une empreinte mémoire de 15 à 40% plus petite que PHP 5.3, en fonction de la nature du code exécuté. À configuration matérielle équivalente, une migration de PHP 5.3 vers PHP 5.4 permet donc un gain très significatif en termes de performance à peu de frais.

PHP 5.4 est également très fiable, car un grand nombre de bugs ont été corrigés et les développeurs du langage ont porté une attention très particulière aux tests unitaires, suite au bug sur la fonction `crypt()` présent dans PHP 5.3.7 et qui aurait pu être évité s'ils avaient vraiment tenu compte de leurs résultats.

Pour autant, la mise en œuvre de PHP 5.4 peut poser actuellement certains problèmes et la décision de l'utiliser pour remplacer PHP 5.3, ou dans le cadre d'un nouveau projet, doit donc être réfléchie.

Tout d'abord, si la disparition des « magic quotes » et de « register globals » n'est pas forcément un problème en termes de compatibilité, du fait qu'il était de notoriété publique qu'il ne fallait pas utiliser ces fonctionnalités, l'abandon du « safe mode » peut engendrer quelques problèmes de sécurité dans le cas où du code a été écrit en tenant compte de son utilisation.

Dans un tel cas, il est recommandé de réaliser un audit de sécurité afin de détecter et de combler les éventuelles failles de sécurité masquées par le « safe mode ».

Ensuite, dans le cas d'un nouveau projet, l'utilisation de fonctionnalités spécifiques à PHP 5.4, comme par exemple les traits ou l'utilisation de `$this` dans une fonction anonyme, rendra le code dans lequel elles sont utilisées exécutable uniquement par PHP 5.4.

En conséquence, sa portabilité sera donc limitée, du moins jusqu'à ce que cette version se démocratise suffisamment pour que ce ne soit plus un problème. Enfin, au moment où ces lignes sont écrites, les versions officielles d'outils tels que APC, Xdebug ou Suhosin ne sont pas compatibles avec PHP 5.4, ce qui peut se révéler problématique dans le cas de sites à forte charge, d'un développement complexe ou d'un site « sensible » en termes de sécurité.

Les versions de développement des outils précités supportent déjà relativement bien PHP 5.4, et ce n'est donc encore une fois qu'une question de temps pour qu'il soit supporté officiellement.

PHP 5.4 est donc une évolution de PHP 5.3 et non le remplaçant ou le successeur de PHP 6, car hormis le fait qu'il soit débarrassé de ses fonctionnalités les plus décriées, il n'apporte finalement aucune innovation majeure ou fondamentale, à l'exception peut-être des traits, et de plus il lui manque toujours le support natif d'Unicode.

Ses performances et ses nouveautés syntaxiques et fonctionnelles qui apportent à ses utilisateurs plus de confort et de puissance en font tout de même une version incontournable à court terme, notamment lorsque les outils utilisés traditionnellement en conjonction avec PHP seront compatibles.

Frédéric Hardy

Directeur technique et fonctionnel agile

Blog : <http://blog.mageekbox.net>

Twitter : <http://twitter.com/mageekguy>

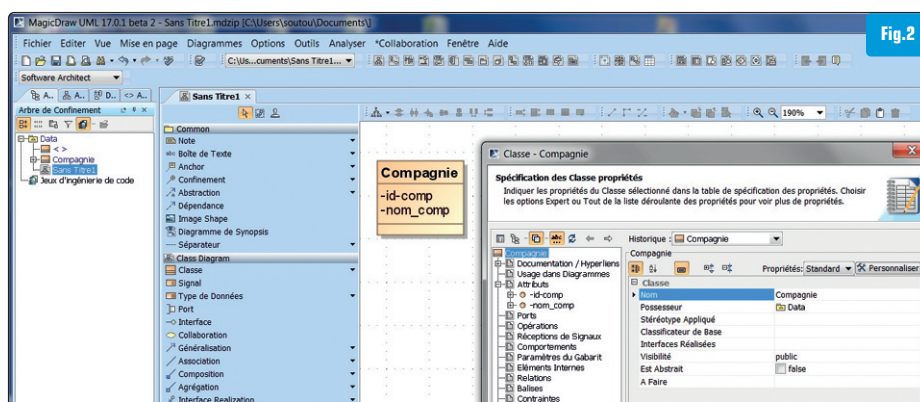
Outils UML pour les bases de données

Il y a déjà cinq ans, Programmez ! avait publié un premier comparatif des outils UML pour les bases de données. Il était temps de réactualiser notre analyse. Christian Soutou s'est remis au travail ! PowerAMC reste incontestablement un outil référence qui n'a rien perdu de ses qualités. Tous les détails ci-dessous.

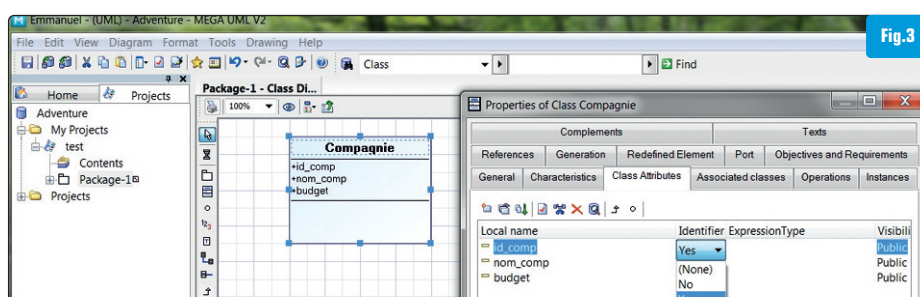
Avertissement : cette étude remonte à novembre 2011, certaines versions récentes n'ont pas été intégrées. Comparer et analyser différents outils est toujours un exercice difficile et subjectif. Tous les éditeurs n'ont pas été intégrés à cette étude et certains ont été retirés. La rédaction.

Fig.1	Produit	Note	Prix indicatif	Version évaluée
	PowerAMC	★★★★★	2800 €	12
	Win'Design	★★★★★	1800 €	7
	Rational Rose Data Modeler	★★★★★	3500 €	7.0
	Objectteering	★★★★★	2000 €	6
	MagicDraw	★★★★★	4250 €	12.0
	Enterprise Architect	★★★★★	118 €	6.5
	Visual Paradigm	★★★★★	540 €	3
	MEGA Designer	★★★★★	NC	2005 SP3
	Together	★★★★★	3500 €	2006 R2
	Visual UML	★★★★★	230 €	5.0
	MyEclipse	★★★★★	45 € /an	5.1
	Visio	★★★★★	630 €	2007
	Poseidon	★★★★★	700 €	5.x
	ModelSphere	★★★★★	1160 €/an	2.5

Le classement de 2007



Environnement de travail



Identification des classes

C'était en 2007...

Cet article compare les outils du marché qui mettent en œuvre la notation UML et la génération de script SQL ou l'interconnexion à une base de données le plus souvent par un pilote ODBC ou JDBC. Chaque outil est évalué sur la qualité de traduction d'un diagramme de classes incluant toutes sortes d'associations. La capacité de rétroconception d'une base de données est également étudiée. Le classement que nous avons établi en avril 2007 (Programmez n°96,) était le suivant : [Fig.1]. Nous avons noté les produits du moment en fonction de leur qualité à traduire un modèle conceptuel, de leur robustesse, ergonomie, documentation et support éventuel que j'avais dû solliciter. La coquille de l'époque concernait le prix de PowerAMC qui en vérité était bien plus élevé que celui affiché (le module objet n'était pas pris en compte), il devait déjà avoisiner les 5000 euros.

Cinq ans après...

Analysons l'évolution de ces offres 5 ans après. Enterprise Architect n'offre plus la possibilité de traduire un diagramme de classes, Visual UML a été inclus dans Visual Paradigm. Peu de nouveaux outils sont apparus et ceux qui sont gratuits ne sont ni fiables, ni performants. Ma sélection s'est portée sur les outils suivants :

- MagicDraw UML, <http://www.magicdraw.com/>
- MEGA Designer, http://www.mega.com/en-product/mega_designer/index.asp?l=en
- Modelio, <http://www.modeliosoft.com/>
- Objectteering, <http://www.objectteering.com/>
- PowerAMC, <http://www.sybase.fr/products/modelingdevelopment/poweramc>
- Rational Rose, <http://www.ibm.com/>
- Visual Paradigm, <http://www.visual-paradigm.com/product/vpuml/>

- Win'Design, <http://www.win-design.com/fr/>

Ces outils proposent tous un processus de conception basé sur les différents niveaux du conceptuel au physique (*forward engineering*), un processus de rétro conception (*reverse engineering*) et un processus d'inter échange (*round-trip engineering*) entre modèles. La partie consacrée aux bases de données n'est pas forcément prépondérante. D'autres fonctionnalités concernent la modélisation de processus métier BPM (*Business Process Models*) qui peuvent être importés ou exportés conformément au langage BPEL4WS (*Business Process Execution Language for Web Services*). Nombre d'entre eux fournissent un référentiel utile aux architectes des systèmes d'information qui en seront les principaux utilisateurs. Certains outils se combinent avec Hibernate, ce qui n'arrange pas nos affaires en vérité.

MAGIC DRAW

La version Architect ou Enterprise présente sur le site Web des fonctionnalités relatives à la génération de code SQL.

La version d'évaluation est plus que restrictive, enregistrement et vérifications multiples, nombre d'objets limités, fichiers en lecture seule, le tout pour 27 jours ! Au secours [Fig.2]. Le clou de l'essai : l'impossibilité de définir un identifiant de classe. En 5 ans : aucun progrès.

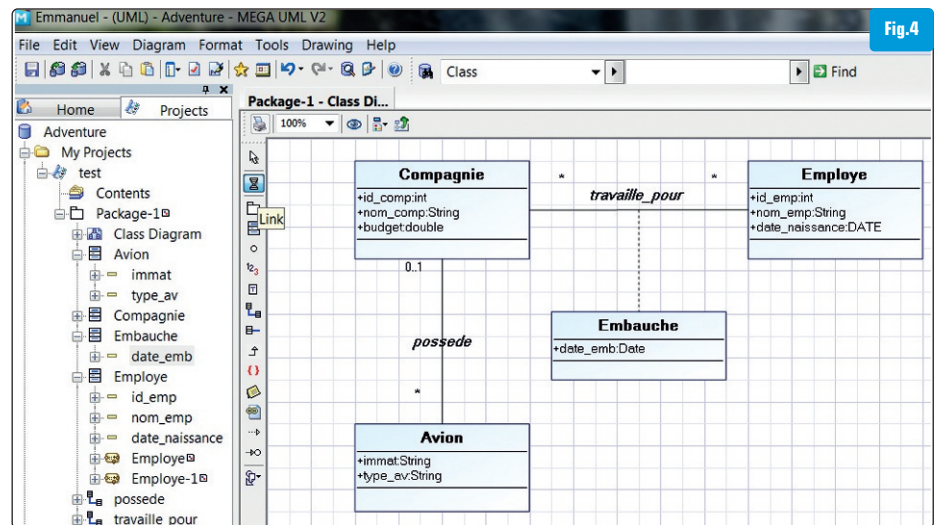
MEGA

L'offre logicielle de la société MEGA regroupe quatre catégories d'outils pour la modélisation, le contrôle, les transformations et les outils de communication. L'outil de modélisation est celui qui est ici brièvement présenté. Pour débiter, créez un projet, un paquetage et reliez le paquetage à la base de données. Cette commande n'est pas active par défaut et nécessite une action manuelle (menu Tools/Options, dossier Compatibility > Others, cochez l'option Data Package dans la partie droite de la fenêtre).

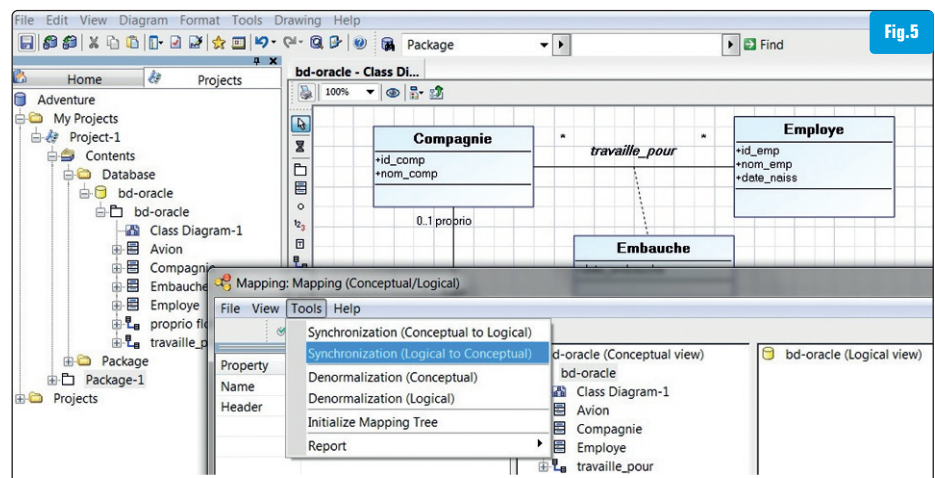
Identifiants

Un clic droit sur la classe choisit Attributs vous permettra de créer des attributs. Sélectionnez ensuite l'identifiant.

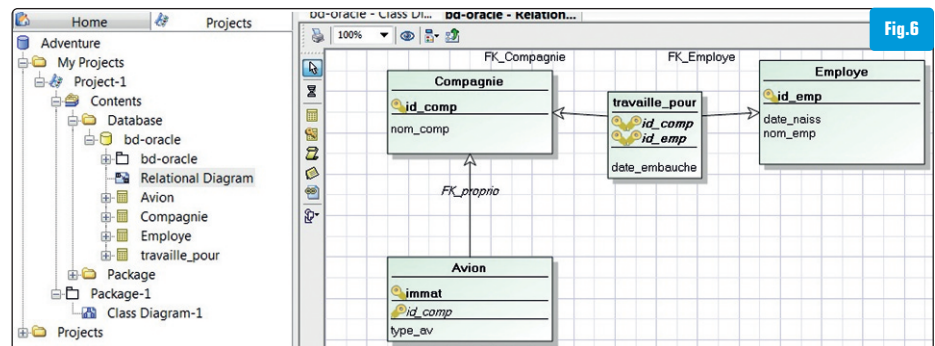
Pour le typage, sélectionnez chaque attribut par un clic droit dans l'explorateur, choisissez Propriétés, onglet Characteristics et trouvez le type adéquat dans la liste Expression Type [Fig.3].



Associations



Génération du modèle relationnel



Modèle relationnel

Associations

Pour créer une classe-association, créez la classe du milieu et, à l'aide du bouton Link, reliez cette classe à l'association [Fig.4].

Génération du modèle relationnel

Afin de générer un modèle logique, positionnez-vous sur votre base, clic droit choisissez Edit puis Outils (synchroniser du conceptuel au logique) [Fig.5]. Un assistant en quatre étapes est alors lancé. Le modèle relationnel est, par la suite, créé dans le paquetage [Fig.6].

Génération des tables

Pour obtenir le script SQL, positionnez-vous sur votre base, clic droit choisissez Generate puis SQL [Fig.7].

Rétro-conception

Créez une base, vide de tout diagramme, renseignez le type du SGBD (Propriétés), clic droit Reverse Target Database. Vous avez le choix entre sélectionner une source de données ODBC préalablement définie ou un script SQL. Pour ce dernier cas, Mega fournit un outil qui se connecte à une base par

ODBC et génère automatiquement un fichier XML décrivant les tables et clés. Afin de générer un modèle conceptuel, vérifiez que vous avez accès au Meta Model en mode Advanced, dans le menu Tools / Options. Un assistant en quatre étapes se lance et produit un nouveau modèle de données (Data Model). Dans ce modèle créez un diagramme de données (clic droit New / Data Diagram) puis insérez dans ce diagramme par un glisser-déposer les entités que vous avez obtenues après rétro-ingénierie [Fig.8]. Les schémas obtenus par rétro-conception sont de très bonne qualité. Dans l'exemple, seules les agrégations devraient être notées en tant que composition. Les points forts de cet outil sont la robustesse l'ergonomie de l'outil de synchronisation par étapes et les passerelles possibles avec d'autres outils.

MODELIO

Modelio est distribué par Modeliosoft, filiale de SOFTEAM, société Française. Ce produit est basé sur l'architecture d'Objectteering et présente une interface façon Eclipse. Avant tout, installez les modules Persistent Profile et SQL Designer, (MDA/Installer un module...) avant de créer un projet puis, à l'intérieur de celui-ci, un diagramme de classes. Chaque classe, attribut et association doit être déclarés persistants (cochez Element Persistent dans la fenêtre en bas à droite dans l'onglet Persistent).

Identifiants

Pour créer des attributs et un identifiant dans une classe, rendez-vous dans la fenêtre Modèle, clic droit sur la classe et Profil de persistance choisissez Property ou Identifier puis tapez la colonne dans la fenêtre en annotant l'élément dans l'onglet Element [Fig.9].

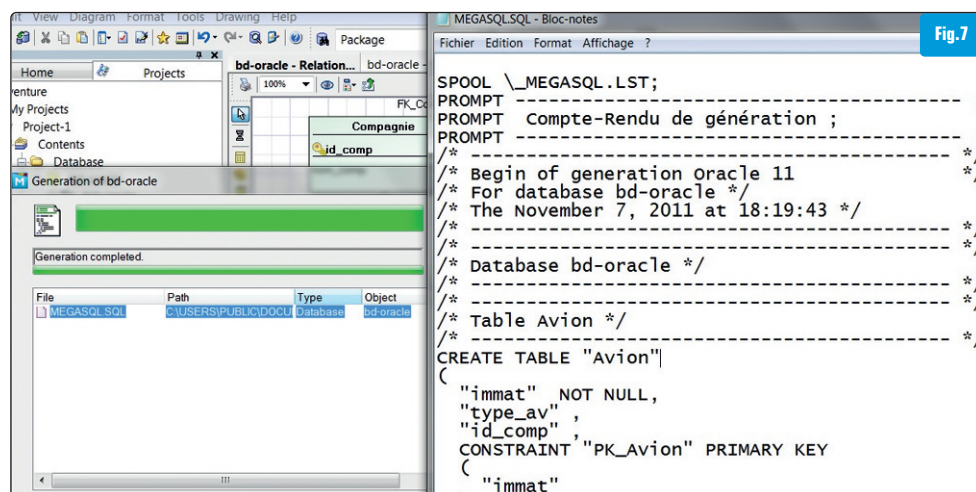
Associations

Chaque type d'association (classes-association, composition, etc.) dispose de son icône. L'annotation d'une association vous permet de fixer les multiplicités [Fig.10].

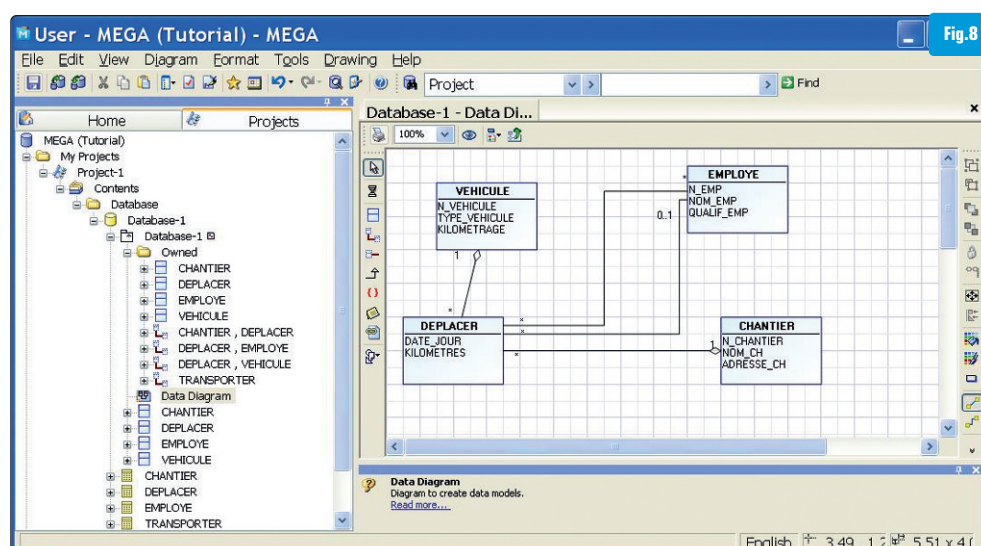
Génération du modèle relationnel

La génération du diagramme du modèle de données (que Modelio nomme diagramme de classes persistantes) se réalise au niveau du projet qui contient le diagramme (clic droit puis SQL Designer/Transformation : UML en modèle de données).

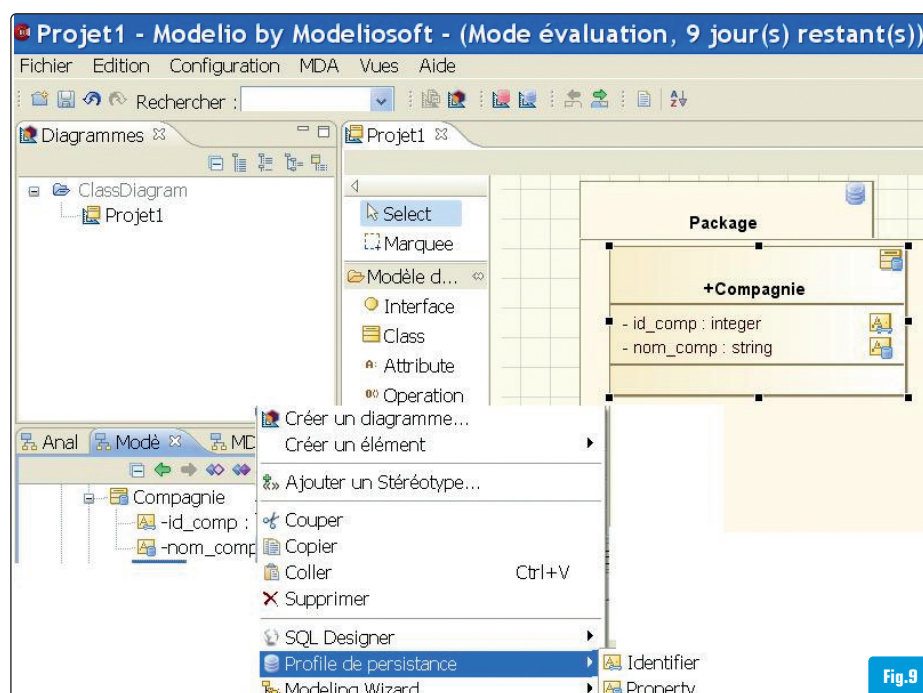
Le problème que j'ai rencontré se situe à ce niveau et concerne la transformation des



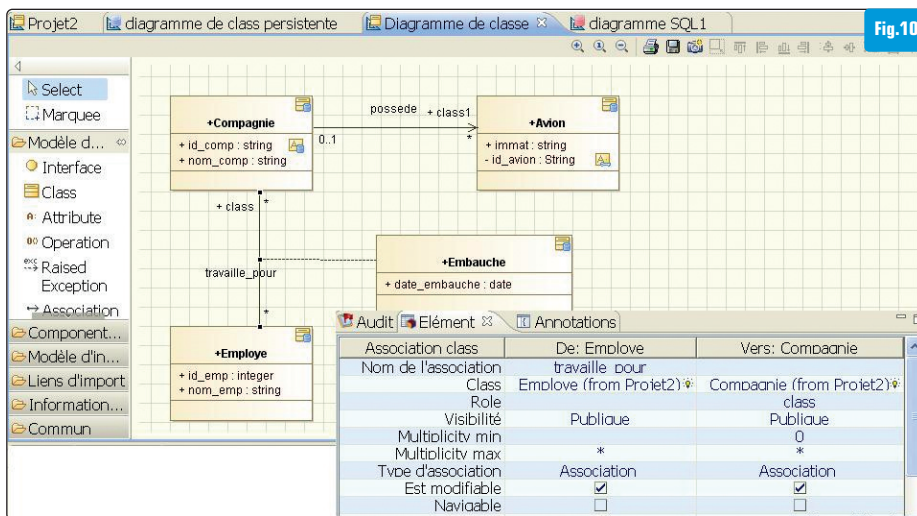
Génération de la base



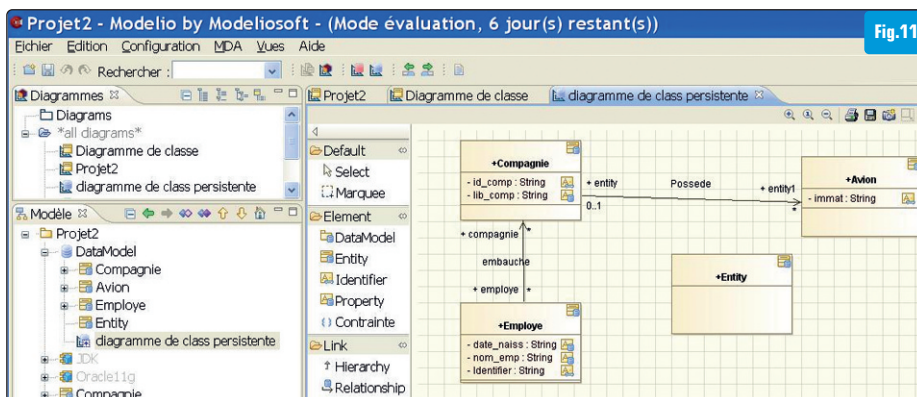
Rétro-conception d'une base



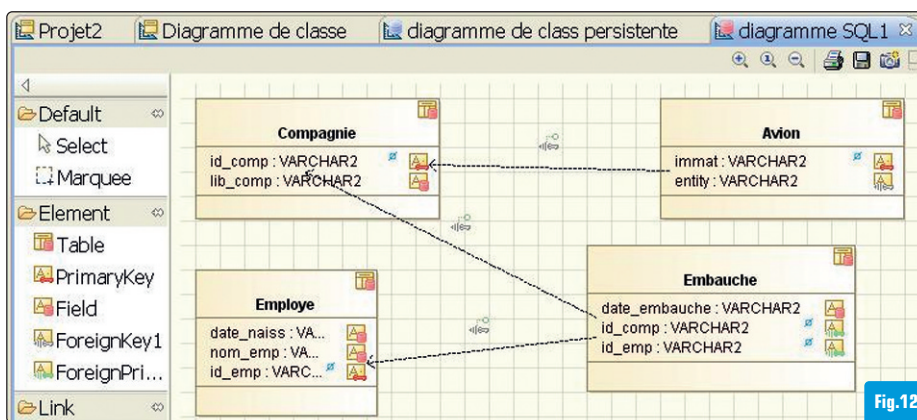
Identification des classes



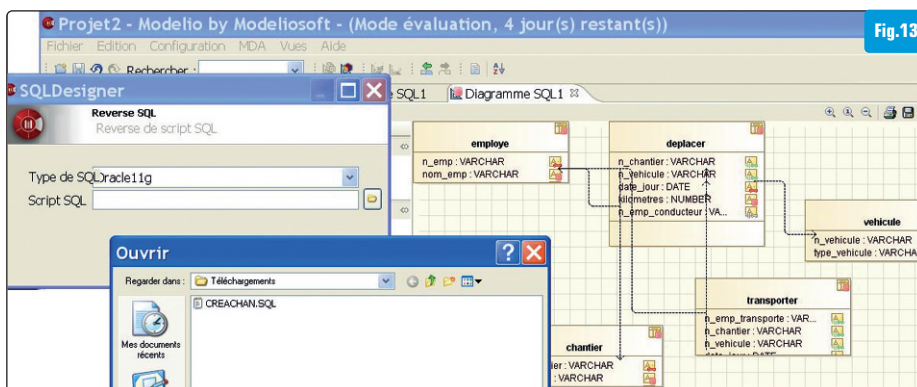
Associations



Génération du modèle de données



Génération des tables



Rétro-conception d'un script SQL

classes-association. Pour des questions d'intégration du module SQL avec le générateur Hibernate et Java, ce type de transformation n'est pas encore implémenté. Pour contourner ce problème, il faut agir soit au niveau du modèle de données, soit modéliser chacune de vos classe-association en tant que classe normale. Ce n'est pas satisfaisant mais ça vous évitera un blocage [Fig.11].

Génération des tables

La génération du diagramme des tables (que Modelio nomme diagramme SQL) se lance par un clic droit sur le modèle de données puis SQL Designer/Transformation : Modèle de données vers modèle de tables. Vous choisirez ensuite le SGBD cible (SQL92, Oracle, MySQL ou SQL Server) [Fig.12].

La génération du script SQL s'opère par un clic droit sur le diagramme SQL, puis SQL Designer/Générateur : Script SQL. Pour effectuer une rétro-conception à partir d'un script SQL, positionnez-vous au niveau du projet, puis clic droit SQL Designer/Reverse : Script SQL. Vous devrez ensuite choisir le type du script (SQL92, Oracle, MySQL ou SQL Server) et localiser le fichier contenant la création des tables et contraintes. Pour remonter dans les niveaux, positionnez-vous sur le diagramme SQL obtenu (qui est de bonne qualité), puis clic droit SQL Designer/Transformation : Modèle de tables vers modèle de données. La suite des opérations est plus hasardeuse lorsque vous désirez générer le modèle UML [Fig.13]. La simplicité des menus et l'ergonomie de cet outil sont appréciables de même qu'un forum réactif. En revanche, il manque un peu de fluidité dans la manipulation graphique des schémas. L'existence d'un modèle de données intermédiaire et la volonté d'intégrer Hibernate dans le même contexte entravent, à mon sens, la qualité de génération entre des modèles UML et SQL et inversement.

OBJECTEERING

Objecteering est un autre produit Français de la société SOFTEAM qui est un acteur majeur dans la communauté des technologies objet et premier membre Européen de l'OMG. Dans cet outil, le terme *physical model* désigne un schéma relationnel (niveau logique) et *logical model* désigne un script SQL. Avant tout, créez un projet puis configurez l'affichage des tags qui serviront à annoter un diagramme de classes pour sa transformation (Outils/Options graphiques...

gramme de classes en modèle relationnel. Des boîtes de dialogue permettent un certain nombre d'options, notamment de choisir son SGBD et les classes à traduire [Fig.22]. Le résultat s'affiche en tant que nouveau schéma relationnel, la fenêtre de gauche contient tous les détails de la composition des tables. A ce niveau, vous pourrez renommer des colonnes et clés étrangères mais surtout typer au mieux chaque colonne [Fig.23].

Génération des tables

Le choix SGBD/Générer la base de données... permet de transformer schéma relationnel en script SQL ou directement en tables par une connexion ODBC au SGBD. Des boîtes de dialogue permettent un certain nombre d'options notamment les relations à traduire, la forme du script SQL, etc. Un aperçu du code SQL est également présenté [Fig.24].

Rétro-conception

Le choix Fichier/Reverse Engineering/Base de données... permet de construire un schéma relationnel à partir d'un script SQL ou par une connexion au SGBD pour extraire des tables d'un schéma. Des boîtes de dialogue permettent un certain nombre d'options notamment le type du SGBD source, les tables à traduire après connexion au SGBD, le nom du script SQL, etc [Fig.25]. Un modèle physique des données est ensuite automatiquement généré.

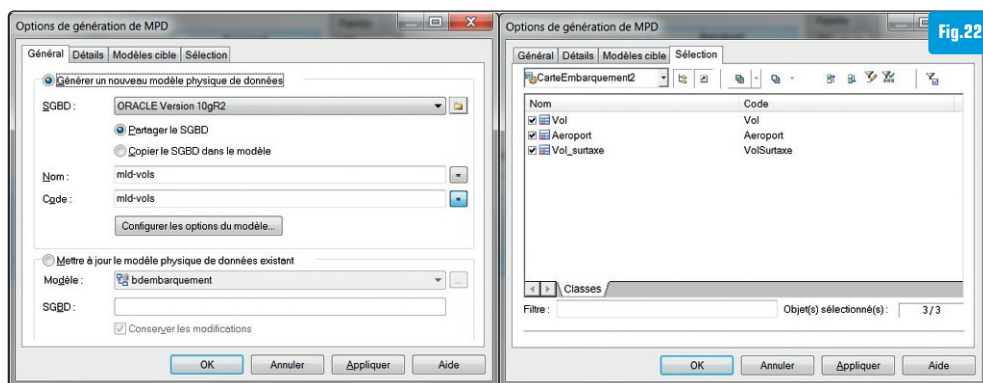
RATIONAL ROSE

Rational Software qui a été à l'origine de la standardisation d'UML est propriété de IBM depuis 2003. L'outil de conception qui est au catalogue de IBM s'appelle toujours Rational Rose Data Modeler.

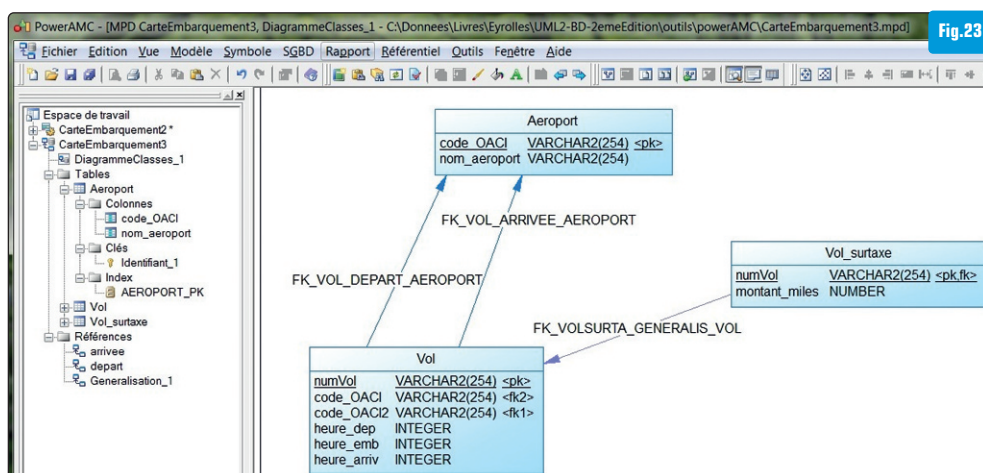
Cet outil est basé sur le formalisme UML et inclut un profil convenant aux bases de données. Un profil est une proposition d'une communauté et regroupe un ensemble d'éléments UML (composants, stéréotypes, icônes, propriétés...) qui s'appliquent à un contexte particulier et qui conservent le méta-modèle d'UML intact.

Quand un tel profil est valide au niveau de la notation, il est dit bien formé. Ce profil n'a pas été adopté par les autres éditeurs de logiciels [Fig.26].

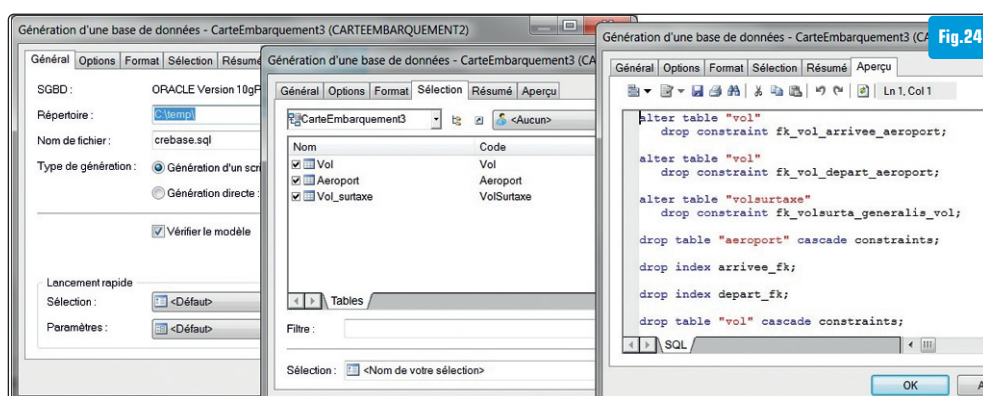
Il est nécessaire dans un premier temps de créer un composant Database dans le répertoire Component View (clic droit Data Modeler/New/Database). Nommez la base



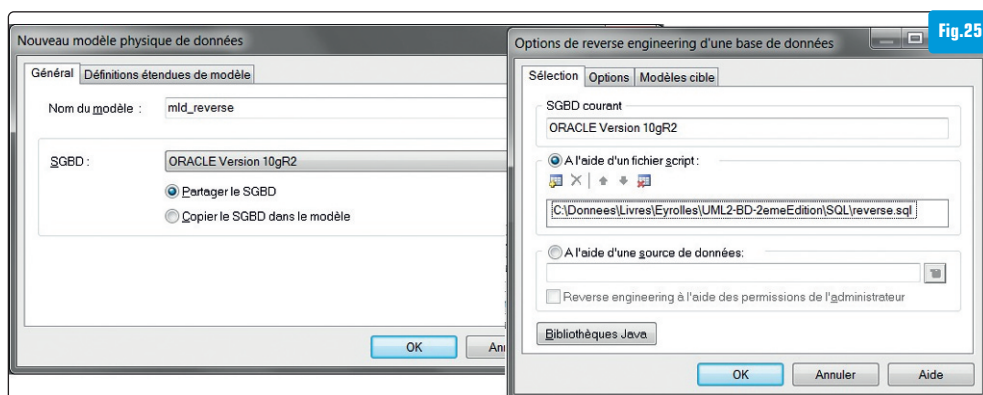
Génération du modèle relationnel



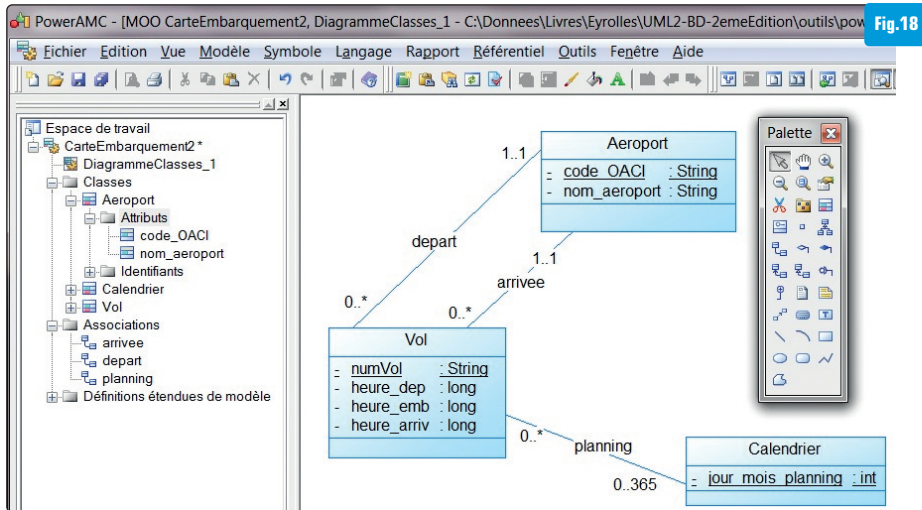
Modèle physique des données



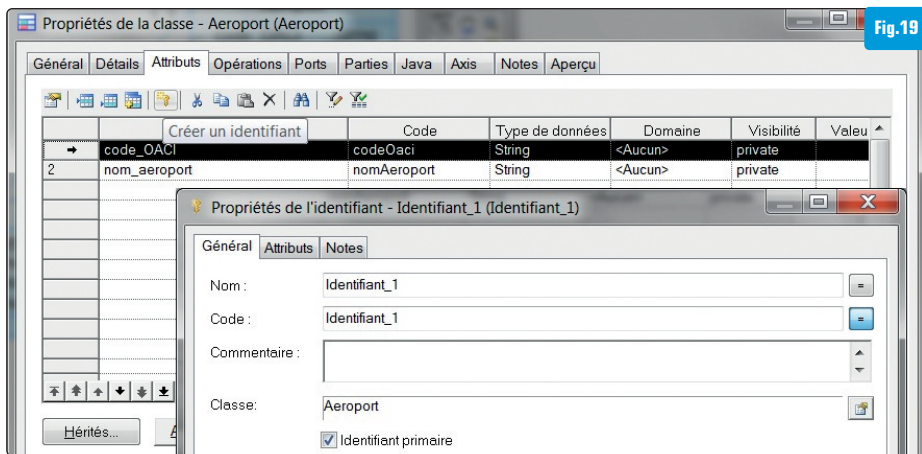
Génération de la base



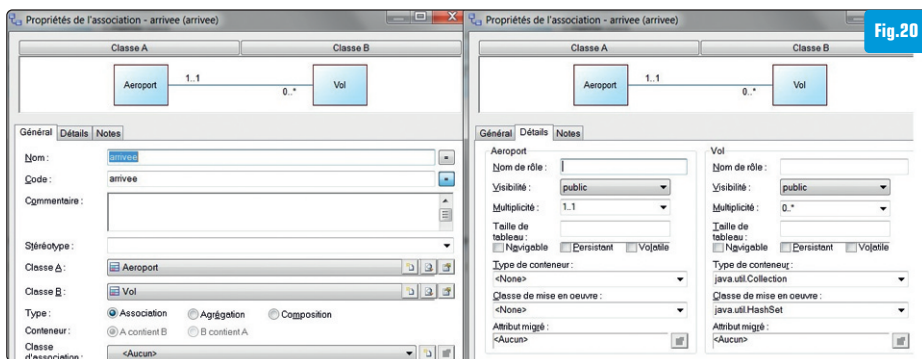
Options de rétro-conception



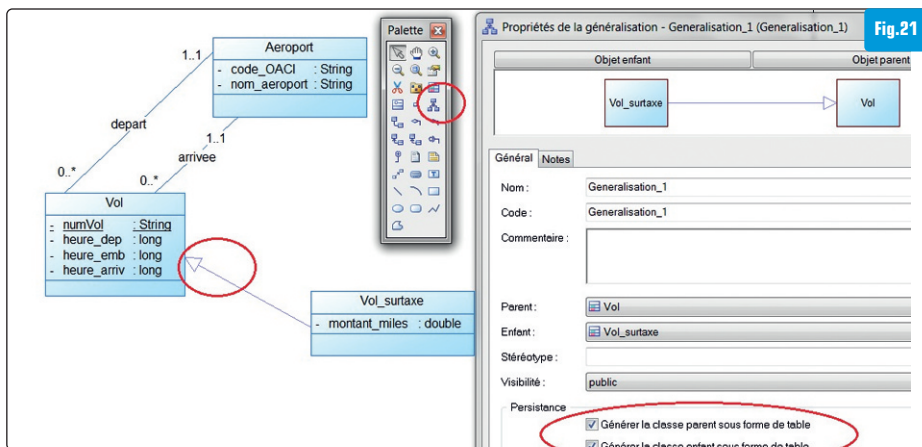
Environnement de travail



Création d'un identifiant



Détails d'une association



Détails des liens d'héritage

points négatifs concernant le faible nombre de SGBD pris en compte et l'absence de processus de rétro-conception.

PowerAMC

PowerAMC (anciennement AMC*Designor) est la version française de l'outil de modélisation PowerDesigner de Sybase. Depuis la version 8, la notation UML (appelée MOO : Modèle Orienté Objet) coexiste avec les notations Merise, entité-relation et IDEF1X. Pour débiter avec un nouveau diagramme de classes Fichier/Nouveau/Modèle Orienté Objet, en choisissant comme langage objet Analyse.

La palette graphique apparaît. Pour visualiser le nom des associations : clic droit sur le fond puis Préférences d'affichage/Association, cocher dans la case Nom [Fig.18].

La fenêtre de gauche contient tous les détails du diagramme de classes.

Identifiants

Pour chaque classe, vous devez définir un identifiant primaire. Le nom de l'identifiant importe peu, vous n'avez pas à le modifier. Sélectionnez l'attribut (ou les attributs), cliquez sur la clé et suivez le guide : [Fig.19]

Si l'attribut n'apparaît pas souligné après avoir validé cette étape, retaillez un peu la classe à la souris ça devrait fonctionner. Si vous désirez ajouter d'autres identifiants (alternatifs), le principe est le même mais vous décochez la dernière case.

Associations

En double cliquant sur un lien d'association, vous obtenez plusieurs fenêtres pour inscrire le nom de l'association, les rôles, multiplicités et composition. Décochez la case *Navigable* qui ne sert à rien [Fig.20]. Pour créer une classe-association, il suffit de rattacher une classe à une association existante par un nouveau lien d'association et le tour est joué.

Héritage

La palette contient un symbole relatif aux liens d'héritage, reliez ensuite la sous-classe à sa sur-classe [Fig.21]. En double cliquant sur un lien d'héritage, vous obtenez une fenêtre qui permet de choisir la traduction de l'héritage (distinction : par défaut, push-down : ne pas générer la classe parent, push-up : ne pas générer la classe enfant).

Génération du modèle relationnel

Le choix Outils/Générer un Modèle Physique de Données... permet de transformer le dia-

gramme de classes en modèle relationnel. Des boîtes de dialogue permettent un certain nombre d'options, notamment de choisir son SGBD et les classes à traduire [Fig.22]. Le résultat s'affiche en tant que nouveau schéma relationnel, la fenêtre de gauche contient tous les détails de la composition des tables. A ce niveau, vous pourrez renommer des colonnes et clés étrangères mais surtout typer au mieux chaque colonne [Fig.23].

Génération des tables

Le choix SGBD/Générer la base de données... permet de transformer schéma relationnel en script SQL ou directement en tables par une connexion ODBC au SGBD. Des boîtes de dialogue permettent un certain nombre d'options notamment les relations à traduire, la forme du script SQL, etc. Un aperçu du code SQL est également présenté [Fig.24].

Rétro-conception

Le choix Fichier/Reverse Engineering/Base de données... permet de construire un schéma relationnel à partir d'un script SQL ou par une connexion au SGBD pour extraire des tables d'un schéma. Des boîtes de dialogue permettent un certain nombre d'options notamment le type du SGBD source, les tables à traduire après connexion au SGBD, le nom du script SQL, etc [Fig.25]. Un modèle physique des données est ensuite automatiquement généré.

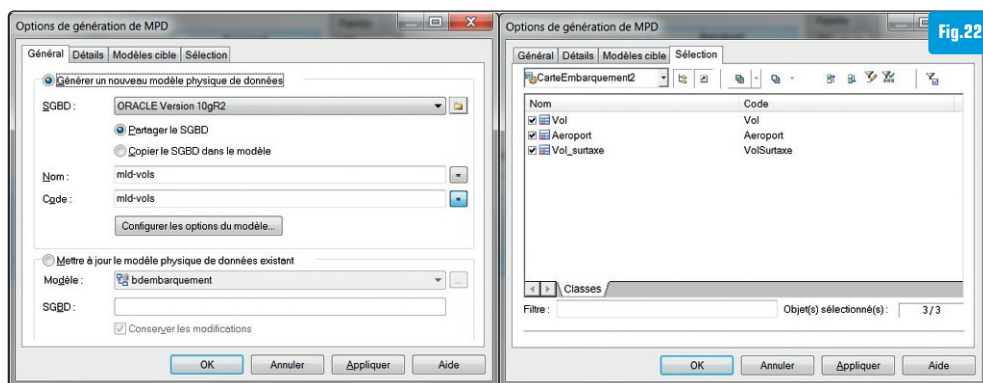
RATIONAL ROSE

Rational Software qui a été à l'origine de la standardisation d'UML est propriété de IBM depuis 2003. L'outil de conception qui est au catalogue de IBM s'appelle toujours Rational Rose Data Modeler.

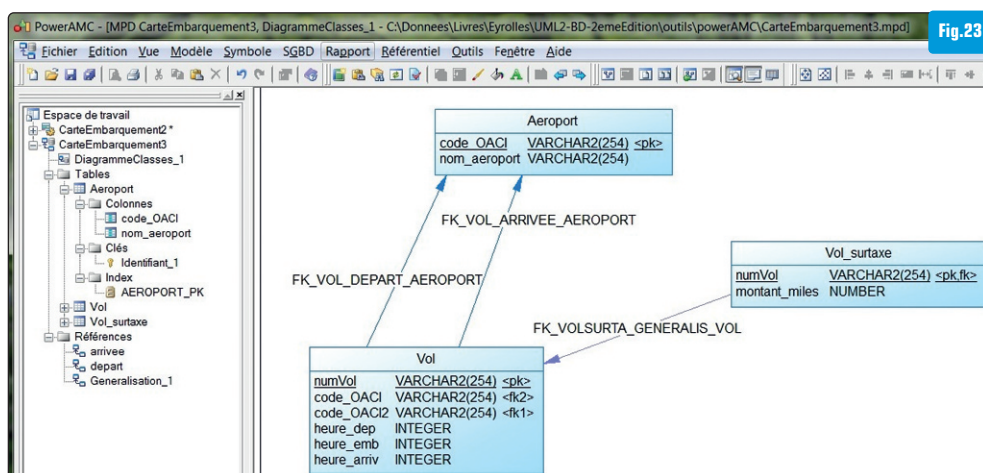
Cet outil est basé sur le formalisme UML et inclut un profil convenant aux bases de données. Un profil est une proposition d'une communauté et regroupe un ensemble d'éléments UML (composants, stéréotypes, icônes, propriétés...) qui s'appliquent à un contexte particulier et qui conservent le méta-modèle d'UML intact.

Quand un tel profil est valide au niveau de la notation, il est dit bien formé. Ce profil n'a pas été adopté par les autres éditeurs de logiciels [Fig.26].

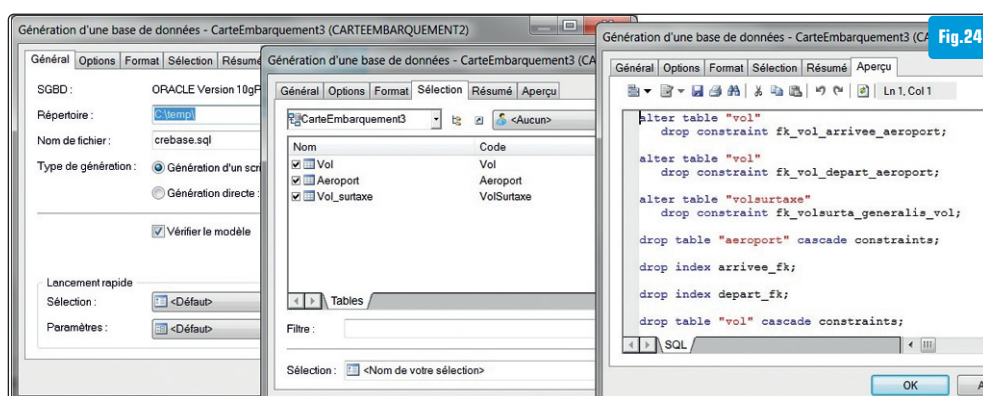
Il est nécessaire dans un premier temps de créer un composant Database dans le répertoire Component View (clic droit Data Modeler/New/Database). Nommez la base



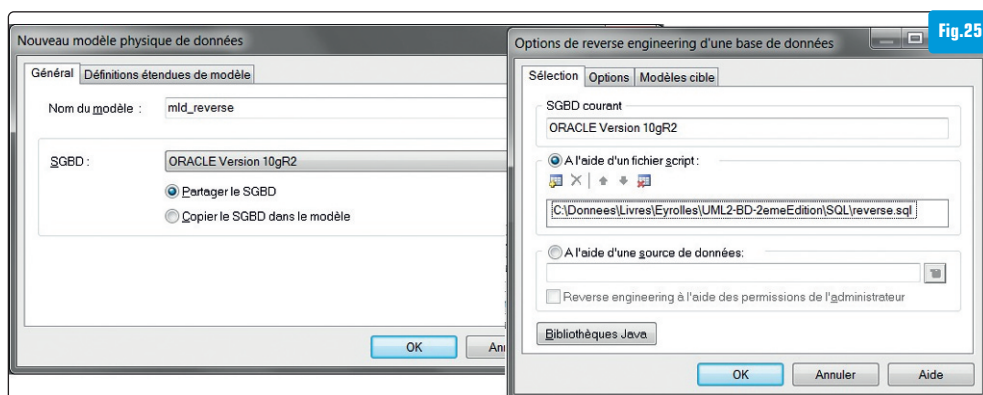
Génération du modèle relationnel



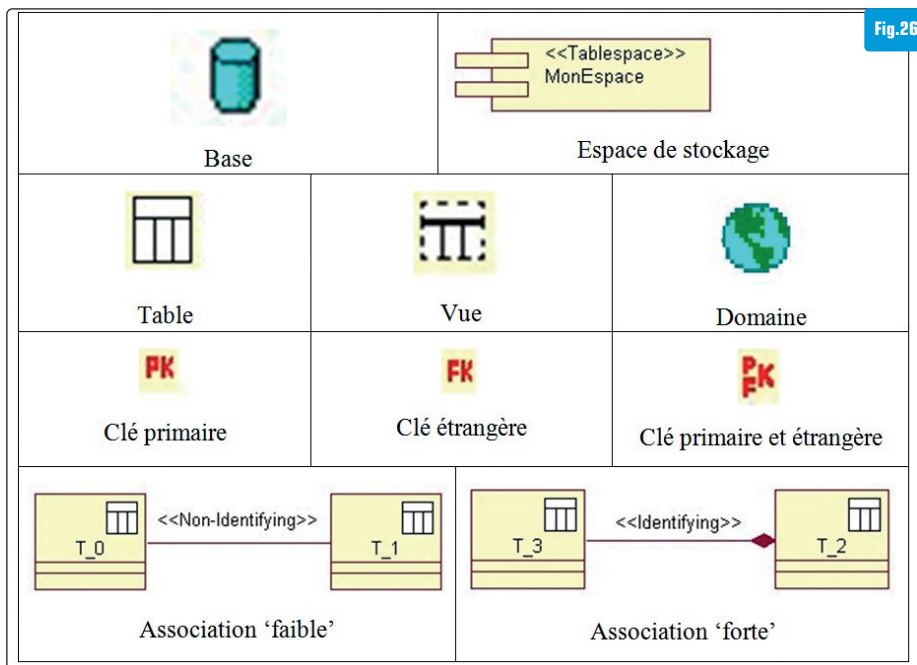
Modèle physique des données



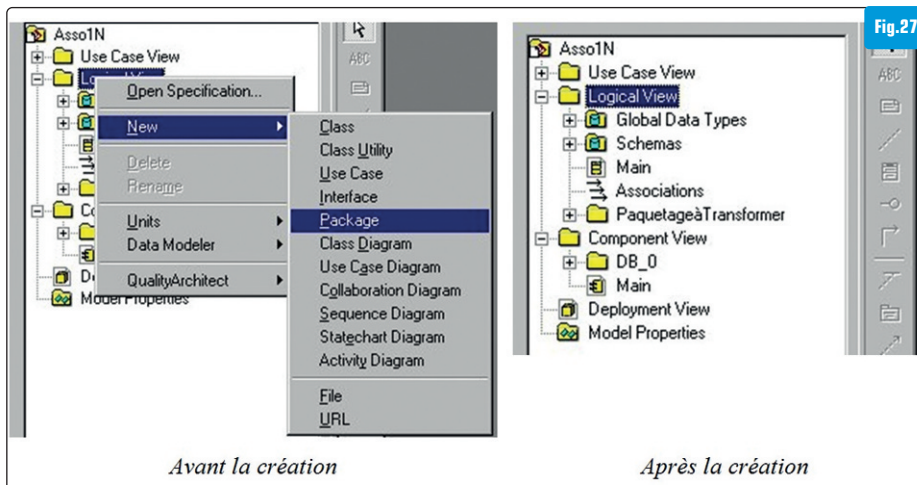
Génération de la base



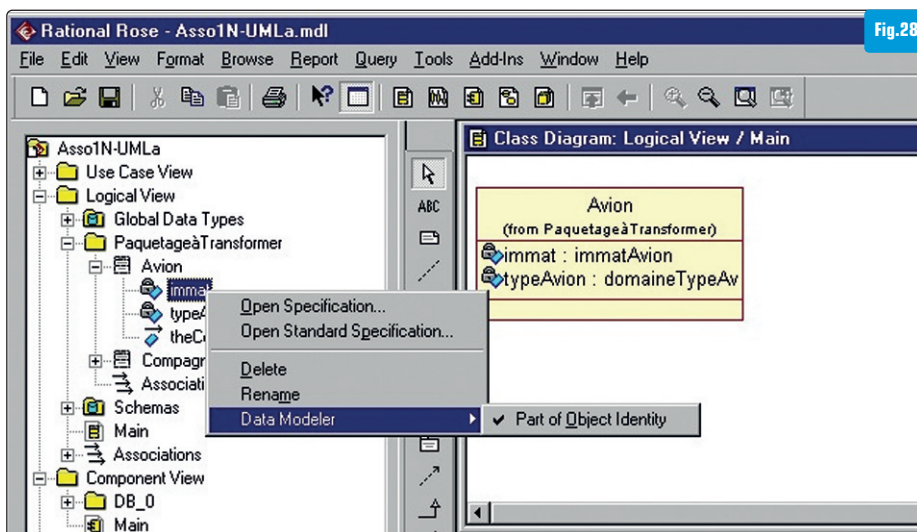
Options de rétro-conception



Icônes du profil UML de Rational



Création



Identification d'une classe

et choisissez le SGBD cible (Name et Target). Créez ensuite un schéma dans le répertoire Logical View (Clic droit Data Modeler/New/Schema) et associez ce schéma à la base de données. Pour créer un paquetage : clic droit sur le compartiment Logical View, New/Package [Fig.27].

Les classes du diagramme UML doivent présenter les caractéristiques suivantes :

- marquées Persistent (clic droit sur la classe, Open Specification, onglet Detail) ;
- groupées dans un paquetage du compartiment Logical View. L'outil transforme un diagramme au niveau du paquetage.

L'utilisation des domaines de valeurs est nécessaire pour affecter à chaque attribut le type de colonne. Pour créer un domaine : clic droit sur Global Data Types, Data Modeler/New/Domain Package.

Identifiants

Pour identifier une classe, vous devez marquer l'attribut en tant que Part of Object Identity. Pour un identifiant composé il faut répéter ce choix sur chaque attribut. Si aucun attribut ne vous convient, vous pouvez laisser l'outil ajouter un attribut de type ID-based column. Pour le SGBD Oracle, le type de cette colonne est NUMBER(10) [Fig.28].

Génération du modèle relationnel

A l'aide de la palette, en choisissant l'icône de l'association, il suffit de relier les classes entre elles, nommer l'association et préciser les multiplicités. Sélectionnez le paquetage (clic droit, Data Modeler/Transform to Data Model...) puis indiquez les nom du schéma et de la base. Si aucune base n'est sélectionnée la transformation sera de type SQL2. Il est possible de créer des index pour les clés étrangères [Fig.29].

Après la transformation, vous devez créer un diagramme pour visualiser le modèle logique (clic droit sur le schéma puis Data Modeler/New/Data Model Diagram). Vous pouvez modifier certaines caractéristiques avant de générer le script SQL (noms des contraintes et colonnes, etc.) [Fig.30].

Génération des tables

Pour générer le script SQL, sélectionnez le schéma (clic droit Data Modeler/Forward Engineer...), un assistant est alors lancé dans lequel vous devrez choisir parmi différentes options (chemin et nom du script SQL à générer, création de commentaires, de tablespaces, de directives DROP, etc.). Il est aussi possible d'exécuter directement le script par une connexion au SGBD.

Rétro-conception

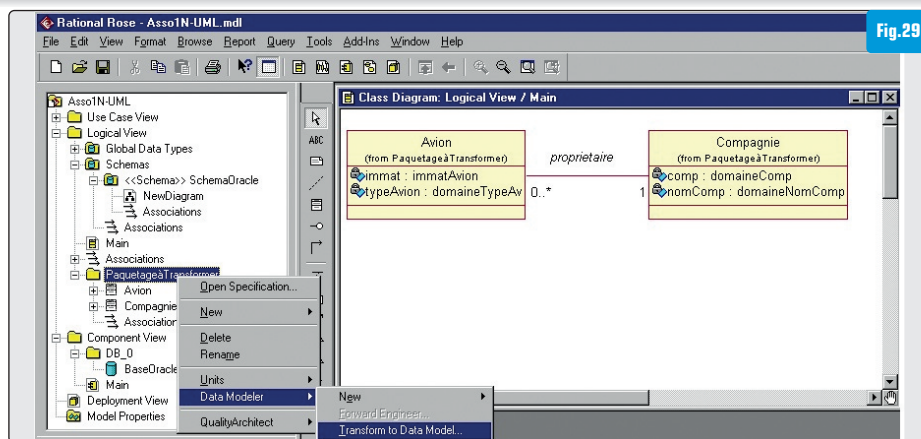
Vous devrez au préalable créer un paquetage vide. Le processus de rétroconception démarre du schéma (compartiment Logical View/Schemas) à sélectionner par un clic droit (Data Modeler/Transform to Object Model). L'assistant permet de sélectionner le nom du paquetage en sortie ainsi que d'autres options. Le fait de transformer un modèle de données dans un paquetage modifie les nouveaux éléments du paquetage mais ne détruit ni ne modifie les éléments mis à jour au niveau du modèle de données. Cet outil n'a pas évolué depuis la première version de l'article, ce qui prouve sa très grande robustesse. On peut toutefois regretter qu'il n'inclut pas de pilotes pour les SGBD open source.

VISUAL PARADIGM

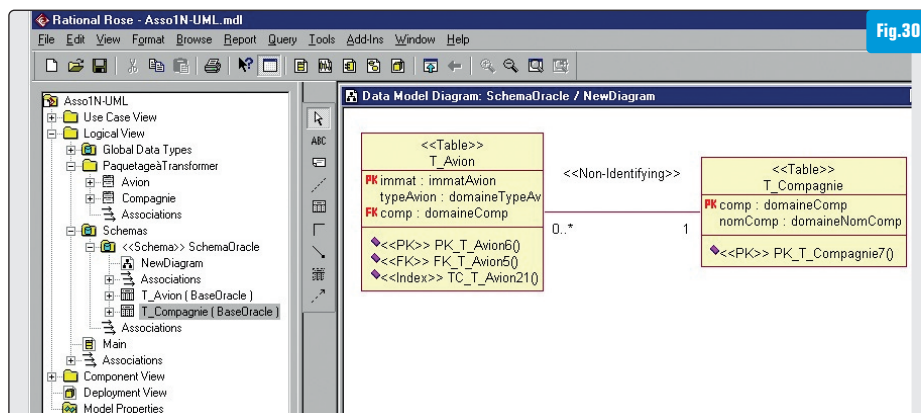
L'édition Enterprise déclare des fonctionnalités relatives à SQL. J'ai donc téléchargé la version d'évaluation qui permet l'enregistrement de diagrammes pendant 10 jours. Une fois saisi le diagramme de classes UML, vous constaterez qu'il n'est pas possible de générer du code SQL directement. La version évaluée en 2007 ne comportait ni identifiant de classes ni associations n-aires. Cette version supporte maintenant les associations n-aires sans inclure les identifiants de classes. Pile ce qu'il ne fallait pas faire [Fig.31]. Ce qui est possible de faire pour l'heure c'est générer un diagramme de classe à partir d'un modèle de type entity-relationship. Très pratique...

WIN'DESIGN

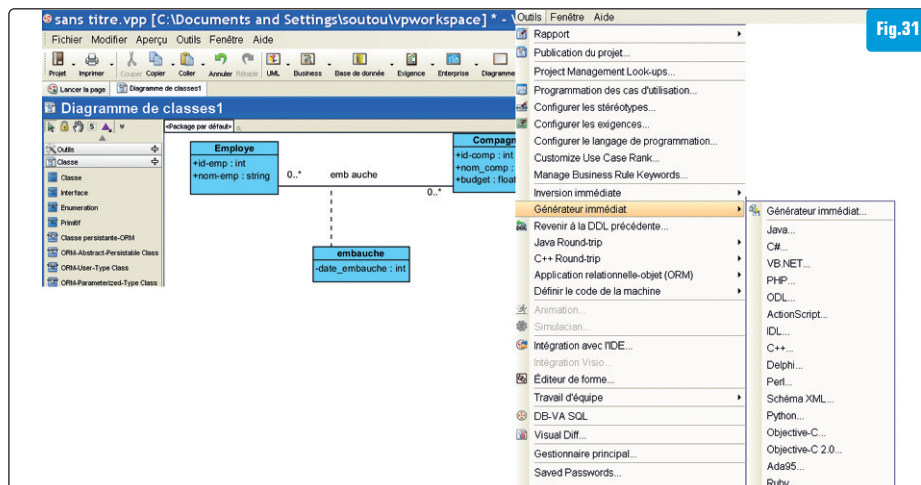
Win'Design est le produit de la société CECI-MA basée à Aix en Provence. Il est présent sur le marché français depuis 1995. Développé initialement pour Merise/2, la notation UML arrive en 2002 avec la version 5. Depuis, l'outil est en évolution constante. En 2011, la version 11 était disponible. La Gamme comprend trois modules autonomes et complémentaires, qui s'articulent autour d'un référentiel (Database pour la conception et le reverse des bases de données, Business Process pour la modélisation des processus métier et Object pour la modélisation UML). Vous devrez disposer du premier et du troisième module pour traduire des diagrammes de classes en script SQL. Win'Design est un outil facile à prendre en main et son interface est très intuitive. Le seul problème de ce produit est qu'il présente encore des lacunes de traduction d'un diagramme UML vers le modèle logique. Lacunes qui sont absentes si vous utilisez la notation



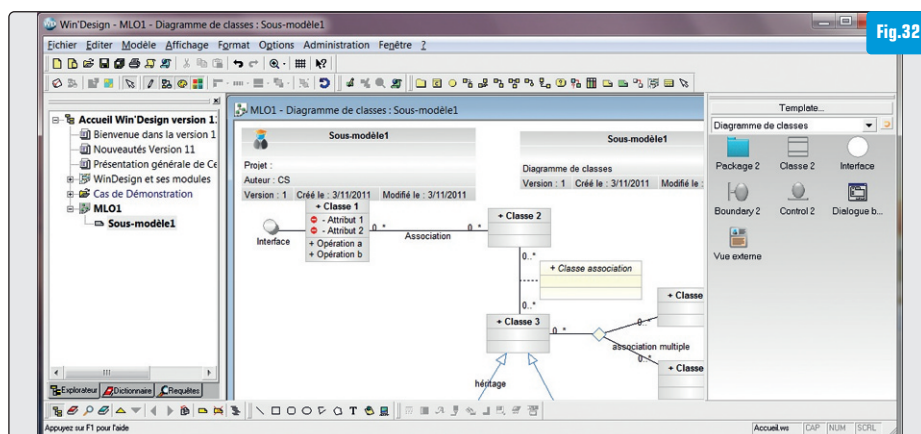
Génération du modèle relationnel



Modèle de données obtenu



Environnement de travail



Environnement de travail

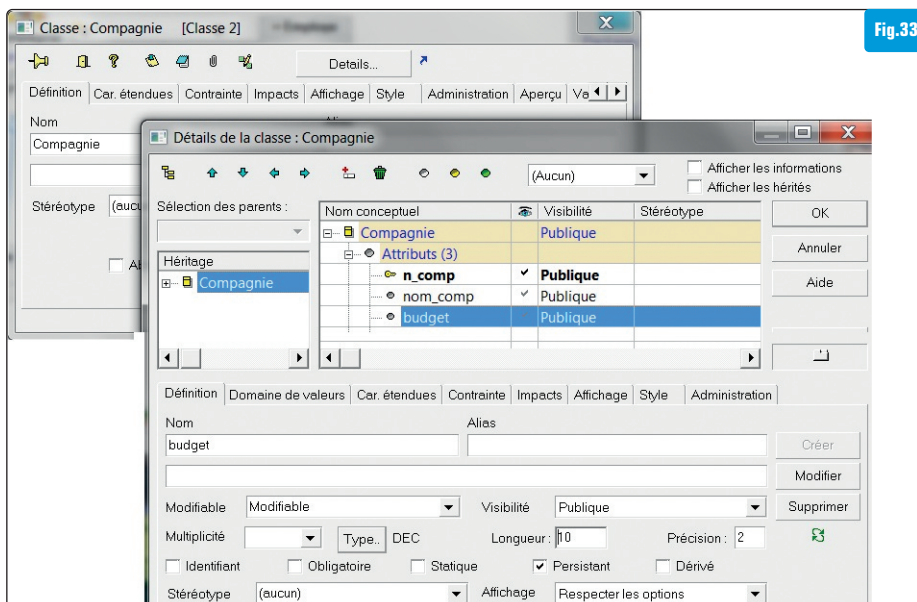


Fig.33

Création d'un identifiant

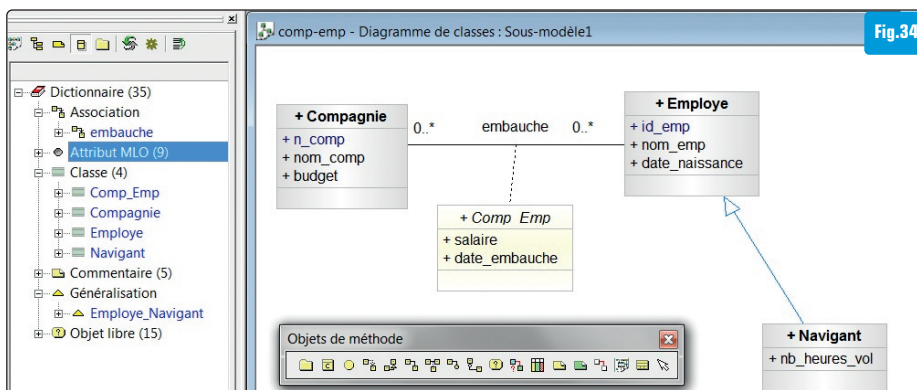


Fig.34

Référentiel et palette d'outils

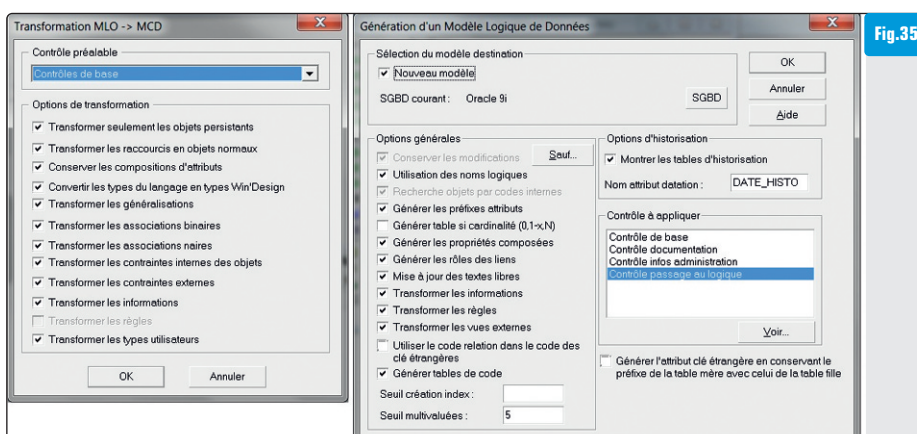


Fig.35

Génération du modèle relationnel

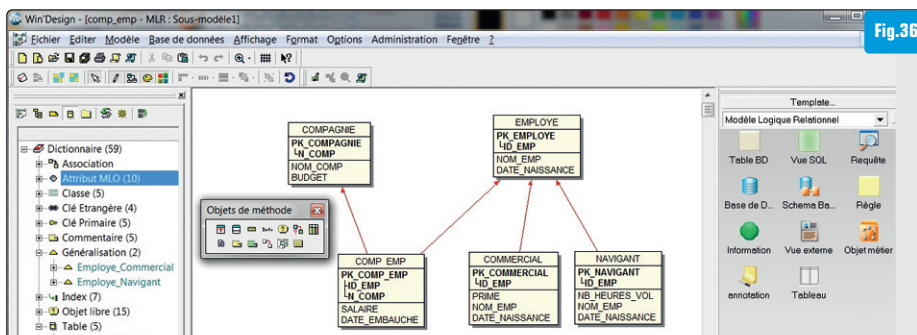


Fig.36

Modèle logique des données

Merise. Pour débiter avec un nouveau diagramme de classes Fichier/Nouveau/Module Objet. Cliquez sur le diagramme de classes et choisissez le langage cible Analyse. Un modèle vierge d'attribut mais contenant tous les types d'association apparaît, ainsi qu'une palette graphique [Fig.32].

Identifiants

Pour chaque classe, le bouton Detail... permet de saisir les attributs et leurs types. L'identifiant est à cocher dans la case associée [Fig.33]. L'attribut identifiant apparaît d'une autre couleur que les autres.

Associations

La fenêtre de gauche contient tous les détails du diagramme de classes. La palette permet de créer tout type classe, association, classe-association, lien héritage [Fig.34]. Pour visualiser le nom des associations, cliquez sur le lien, dans l'onglet Affichage, cochez Nom. Il est possible de créer des associations n-aires, mais il n'est pas possible de relier une classe-association à une autre classe par une association simple. Cette fonctionnalité n'est possible qu'au niveau de Merise en utilisant un lien identifiant.

Héritage

Il n'est pas possible de choisir la traduction de l'héritage (distinction, push-down ou push-up). Par défaut, la méthode push-down est adoptée. Ce choix de décomposition n'est possible qu'au niveau de Merise.

Génération du modèle relationnel

Le choix Modèle/Générer modèle logique permet de transformer le diagramme de classes en modèle relationnel. Des boîtes de dialogue permettent un certain nombre d'options, notamment le choix du SGBD [Fig.35]. Le résultat s'affiche en tant que nouveau modèle logique, le référentiel s'enrichit des détails des tables (clés et index). A ce niveau, vous pourrez renommer des tables, colonnes ou clés étrangères [Fig.36].

Génération des tables

Le choix Base de données/Générer script... permet de transformer un schéma relationnel en script SQL. Des boîtes de dialogue permettent un certain nombre d'options notamment la composition du script SQL.

Rétro-conception

Le choix Fichier/Reverse Engineering/Base de données... permet de construire un

modèle logique à partir d'un script SQL ou par une connexion au SGBD pour extraire des tables d'un schéma. Des boîtes de dialogue permettent un certain nombre d'options, notamment le répertoire qui contient le script SQL, les tables à traduire, etc. Un modèle logique des données est ensuite automatiquement généré.

Produit	Editeur	Prix indicatif (€)	Version évaluée	Note
PowerAMC	Sybase	6800 €	15.3	★★★★★
MEGA Designer	MEGA	3300 €	2009 SP5	★★★★★
Rational Rose Data Modeler	IBM	2100 €	7.0	★★★★★
Objecteering	SOFTEAM	1800 €	6.1	★★★★★
Win'Design	CECIMA	2980 €	11.0	★★★★★
Modelio SQL Solution	SOFTEAM	400 €	2.0	★★★★★
Visual Paradigm	Visual Paradigm	1400 €	8.3	★★★★★
MagicDraw	No Magic	1700 €	12.0	★★★★★

Fig.37

Et le gagnant est...

CONCLUSION

L'évolution d'UML, d'un point de vue normatif, vers les bases de données, ne se concrétise toujours pas car la RFP (Request For Proposal) nommée Information Management Metamodel proposée par l'OMG en 2006 est toujours au point mort. Cette spécification visait à définir un méta modèle incluant différentes sous-parties : Common Warehouse Metamodel, UML2 Profile for Relational Data Modeling, UML2 Profile for Logical (Entity Relationship) Data Modeling, UML2 Profile for XML Data Modeling et UML2 Profile for Record Modeling (COBOL). Cette absence de normalisation n'empêche personne d'utiliser UML pour concevoir une base de données

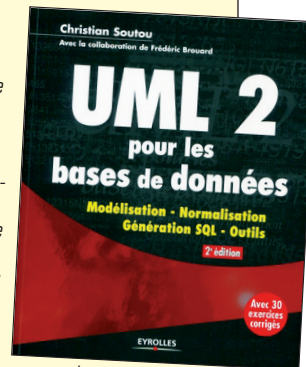
relationnelle, la preuve, bon nombre d'outils ont implémenté les principaux mécanismes. En fonction des tests réalisés, des manipulations relatives à l'installation, à la recherche de documentation, etc. voici le classement, qui n'engage que moi, bien sûr [Fig.37].

Christian Soutou

Maître de conférences à l'université Toulouse II et consultant indépendant. Rattaché à l'équipe MACAO du laboratoire IRIT et au département Réseaux et Télécoms de l'IUT de Blagnac, il intervient autour des bases de données et des technologies de l'information (XML et Web services) en licence et master professionnels, ainsi que pour le compte de la société Orsys. Il est également l'auteur des ouvrages SQL pour Oracle et Programmer avec MySQL, parus aux éditions Eyrolles.

UML 2 pour les bases de données (Édition Eyrolles)

Christian Soutou et Frédéric Brouard proposent la 2e édition de UML 2 pour les bases de données. Entièrement réécrit, l'ouvrage revisite largement la modélisation et les données avec de nombreux nouveaux exemples, les versions les plus récentes des outils. Ce livre est dédié aux architectes, développeurs, développeurs en bases de données.



Modeliosoft

Solutions d'entreprise basées sur l'atelier open source leader de modélisation
Modélisation et implémentation des Systèmes et Logiciels

Développeurs logiciels

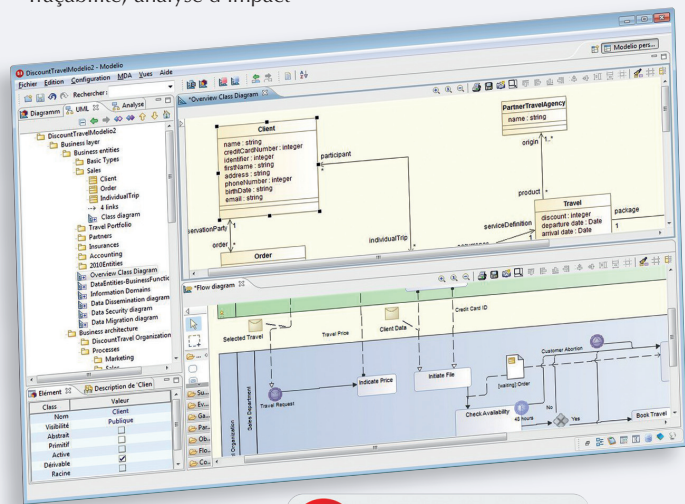
- UML pour Java, C++, C#, SQL, WSDL, XML
- Génération, roundtrip, reverse

Architectes métier

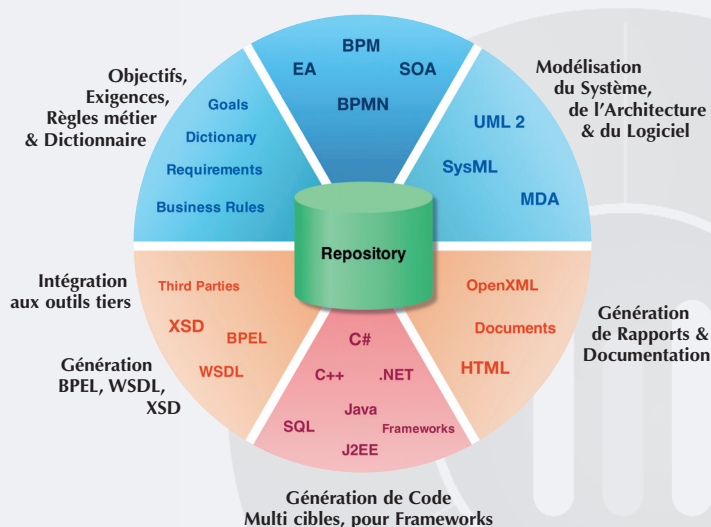
- Modélisation intégrée de UML, BPMN, Architecture d'Entreprise,
- Analyse intégrée des exigences, des objectifs, ...

Architectes système

- Modélisation Systèmes, Logiciels, Hardware, Exigences
- Traçabilité, analyse d'impact



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



Modelio.org

- L'atelier mature de modélisation UML/BPMN open source
- Le Store Modelio : modules et extensions pour modéliser et générer

Solutions d'entreprise Modeliosoft

- Solutions pour développeurs, architectes métier, architectes systèmes
- Génération documentaire Word, HTML : plans types dédiés
- Travail de groupe distribué, gestion de version et configuration

« Geek un jour, Geek pour toujours. »

Nabil Babaci est consultant SharePoint au sein de la SSII MCNEXT. Il intervient aussi bien dans le domaine du développement que sur l'IT. C'est aussi un grand passionné de sécurité informatique. Il publie régulièrement des articles dans la presse spécialisée. Mais le développement, et particulièrement .Net, reste sa passion numéro un.

Propos recueillis par François Tonic



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

Nabil : Je dois avouer que cela a été une pure coïncidence. Pour moi l'informatique a été une révélation assez récente. Voici mon parcours : je suis sorti d'un BAC STT ACC, et oui, pas de STI ni de S, mais bien du marketing. Je dois avouer que cette filière m'a permis d'acquérir de bonnes bases pour les présentations de produits ou de projets et de développer le côté rédactionnel, essentiel dans l'informatique aujourd'hui. Ensuite je suis allé dans une université d'Anglais, très important notamment pour la documentation 😊, puis comme cela ne marchait pas très fort, je me suis pris une année sabbatique et là je vous le donne en mille, j'ai commencé à faire de l'informatique non-stop, notamment pas mal de hacking (software et hardware). La révélation a été quand j'ai publié mon premier article en 2007. Ce qui fait que quelques mois plus tard j'ai intégré une école informatique, Exia.Cesi dans la filière développement. Dès lors, l'informatique a été pour moi une constante source d'amusement et de rencontres formidables. Ce qui m'a permis, par exemple, de créer des associations, de continuer dans le monde des publications, d'intégrer le programme Microsoft Student Partner où j'ai eu l'occasion de rencontrer des gens passionnés et passionnants. Pour résumer, je suis un peu un touche-à-tout.

Avant de m'orienter pleinement vers les technologies Microsoft j'évoluais dans le monde Open-Source, notamment en qualité de DBA MySQL/ SySAdmin. Ce qui me permet aujourd'hui d'avoir un recul sur le choix des architectures et des possibilités technologiques.

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

Nabil : Pour ma part, je pense que l'on ne peut pas se passer de développement. Nous sommes dans un secteur où les possibilités sont illimitées. Qui aujourd'hui n'est pas de près ou de loin dans l'informatique ? Ce qui fait qu'on aime toujours et encore le développement, c'est à mon sens, notre capacité à nous adapter aux différents langages, le goût du challenge et, bien sûr, la recherche de connaissances dans des technologies connues ou non. Une personne passionnée trouvera toujours le moyen de s'amuser dans ses projets.

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

Nabil : Geek un jour, Geek pour toujours. L'implication des technologies dans nos environnements quotidiens fait que nous cultivons sans le savoir un « geek style ». Cultiver ce côté geek est primordial, en ce

moment je suis à fond sur les « Visual Studio Achievements ». La veille technologique occupe une grande place et je lui consacre quotidiennement du temps.

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

Nabil : Aujourd'hui le job de développeur a (beaucoup) changé. Et les développeurs aussi. Nous faisons face à des développeurs communicants et communautaires. Il est devenu une pièce maîtresse dans les projets. Et je pense que bon nombre d'entreprise en sont conscientes. Ce qui fait qu'aujourd'hui les développeurs ont la cote. Côté personnel, je dois avouer que « la pression » n'est pas si présente que cela. Si vous venez avec un état d'esprit qui se veut positif, les projets se déroulent le plus simplement du monde.

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

Nabil : Ce que j'aime particulièrement faire en dehors du travail, c'est me rendre dans des expositions. J'accorde une place essentielle à la culture autre qu'informatique. Je trouve cela nécessaire dans la mesure où

Et le bureau de Nabil ?

Concernant mon bureau, je suis un adepte du minimalisme, avoir ce qui pour moi est le nécessaire, faire en sorte que tout tienne dans mon sac-à-dos. Je ne sors jamais sans smartphone. Concernant ma plateforme de travail, c'est juste un laptop de guerre, et comme tout le monde je code en musique.

cela me permet de m'évader et comble de l'ironie, il m'arrive de trouver des solutions à mes problèmes de développement lorsque j'admire une peinture 😊. Cependant, il est vrai que mes semaines sont assez chargées mais je trouve toujours du temps pour joindre l'utile à l'agréable.



Peux-tu nous présenter ton quotidien en quelques mots ?

Nabil : En réalité mon quotidien commence la veille, car en quittant mon travail j'ai toujours une idée de développement ou d'activité à mettre en oeuvre.

Si je dois définir cela de manière chronologique, le matin c'est veille technologique (vive les smartphones), puis ma journée est rythmée par mes développements projets (quelques achievements sous Visual Studio 😊), puis quelques réunions, des écritures d'articles, de blogging ou de présentation produits/projets et pour terminer, finaliser mes idées et faire en sorte de les réaliser (je

vous réserve une petite surprise dans les mois qui vont suivre). En somme des journées bien remplies.

Comment vois-tu ton job évoluer ?

Nabil : Aujourd'hui, je suis sur SharePoint (et j'adore) et je compte y rester. L'évolution logique de mon job serait Architecte logiciel. Au rythme où les technologies progressent, je préfère rester dans la technique. Non pas que les métiers de type Chef de projet ou Lead Technique me déplaisent, cela se fait plus au niveau de la création. Je préfère être dans un processus de réalisation de haut niveau avec des problématiques fortes.

Des conseils aux étudiants et dévs qui nous lisent ?

Nabil : L'un des conseils que je peux donner est de mettre à profit le temps dont on dispose pendant ses études pour construire des projets, se faire connaître au travers de blogs, partager son expérience. Choisir la technologie que l'on souhaite et la maîtriser à fond. Concernant la technologie, il est important de cultiver sa curiosité et de faire des comparatifs sur ce qui plus tard vous sera bénéfique. Ne pas hésiter à prendre contact avec les communautés, elles se feront un plaisir de vous aider dans vos projets. Et j'ajouterais enfin, joindre l'utile à l'agréable. #

Publication judiciaire

Par jugement du Tribunal de commerce de Nanterre du 27 janvier 2012, la société TLM COM a été déboutée des demandes qu'elle formulait à l'encontre des sociétés CONVIVIANCE et ICSIT dans les termes suivants :

par ces motifs :

Le tribunal, statuant publiquement par un jugement contradictoire, rendu en premier ressort,

- Déboute la SA TLM COM de sa demande de communication des 21 fichiers saisis par huissier le 14 Mai 2007, tels que listés dans sa demande du 29 Septembre 2011,
- Dit que la SA TLM COM n'apporte pas la preuve que la SARL CONVIVIANCE et la SARL ICSIT aient commis à son encontre des actes de concurrence déloyale et la

déboute de ses demandes de condamnation de la SARL CONVIVIANCE et de la SARL ICSIT à ce titre,

- Déboute la SA TLM COM de sa demande de condamnation de la SARL CONVIVIANCE et de la SARL ICSIT en réparation de son préjudice allégué pour atteinte à son droit de producteur de base de données,
- Ordonne à la SARL CONVIVIANCE de déposer ses comptes annuels au greffe de ce tribunal et ce sous astreinte de 50 € par jour de retard à compter du prononcé du présent jugement pour une durée de 3 mois, se réservant la liquidation de l'astreinte,
- Ordonne la publication judiciaire du dispositif de la présente décision dans deux revues ou journaux à caractère professionnel aux frais de la SA TLM COM, dans un délai de 3 mois du prononcé de la présente décision, dans la limite de 5000 € HT par publication,

- Condamne la SA TLM COM à payer à la SARL CONVIVIANCE et la SARL ICSIT la somme totale de 20 000 € au titre de l'application de l'article 700 du code de procédure civile, déboutant du surplus,
- Ordonne l'exécution provisoire du présent jugement sans contestation de garantie,
- Condamne la SA TLM COM aux entiers dépens y incluant les frais d'expertise judiciaire.

Liquide les dépens du Greffe à la somme de 260,88 Euros, dont TVA 42,75 Euros.

Délibéré par Messieurs BEAUDOUT, BRUGERE et THAUMIAUX.

Dis que le présent jugement est mis à disposition au greffe de ce Tribunal, les parties en ayant été préalablement avisées verbalement lors des débats dans les conditions prévues au deuxième alinéa de l'article 450 du C.P.C.

Notre sélection de livres

Jean engels

HTML 5 et CSS 3

Eyrolles

HTML 5 et CSS 3 sont désormais les deux éléments clés pour tout développeur web digne de ce nom. Ce livre est conçu pour les étudiants et pour tout ceux qui veulent apprendre et se mettre à niveau. L'auteur met en avant les bonnes pratiques de développement Web, dont xHTML. Il consacre une large place aux nouveautés de HTML5 et CSS3 : nouveaux éléments de structure sémantique, nouveaux éléments audio et vidéo, Media Queries pour l'adaptation des pages Web au terminal de



l'utilisateur (écran d'ordinateur, tablette, smartphone), stockage local, nouveaux sélecteurs CSS, nouveaux types de contours et d'ombrage, arrière-plans multiples et dégradés de couleur, multicolonnage et effets de texte, etc. Avec de nombreux exercices !

Jeremy Osborn

HTML 5 par la pratique

First

Ouvrage volontairement très didactique, vous y trouverez toutes les bases pour comprendre HTML 5. Suivez les instructions pas à pas proposées dans les quinze chapitres de ce livre pour vous former aux nouveautés de HTML5 et de CSS3, et créer des pages Web plus belles et plus efficaces que jamais. Un ouvrage très sympathique et plutôt facile à suivre. Une bonne manière de débiter en douceur avec les nombreux concepts de HTML 5.



COLLECTIF

Programmation Android

First

Android est avec iOS, la plateforme mobile la plus active en ce moment. Voici un nouveau livre pour apprendre la programmation Android. Les auteurs couvrent jusqu'aux versions 3.x uniquement, la v4 n'est pas abordée (malheureusement) ! Tous les aspects du système sont étudiés : SDK natif, SDK, les outils de développement, la partie matérielle, les API, la gestion des données, le réseau, l'architecture système et des applications. Bref, tout ce qu'il faut savoir pour bien débiter avec Android. De nombreux codes sont montrés et expliqués. A vous de jouer !

COLLECTIF

Processing

Pearson

Processing est un langage open source dont le postulat de départ est de produire du code source comme une œuvre d'art ! Créé par des artistes pour des artistes dans un esprit de simplicité et de cohérence, Processing est un couteau-suisse multimedia qui permet de réaliser par programmation toutes sortes d'applications dans les domaines du design graphique, de l'image



animée, du son, de la 3D ou de la communication interactive. Avec Processing, le code informatique devient un matériau des arts plastiques au même titre que l'argile, le fusain ou l'aquarelle. Essayez-le !

OLIVIER HEURTEL

PHP 5.4

éditions ENI

PHP 5.4 est sorti il y a quelques semaines et nous disposons déjà d'un livre dessus ! L'auteur revient rapidement sur PHP et ses concepts de base, puis il aborde l'ensemble

des nouveautés de la version 5.4 avec de nombreux exemples de code. Une bonne introduction à ce nouveau PHP.

JOËLLE DELACROIX

Linux programmation système et réseau 4e édition

Dunod, à paraître

L'auteur propose ici de revoir les fondations du système Linux, ce qui n'est jamais superflu quand on doit l'administrer, le dépanner



ou tout simplement, le déployer. Chaque chapitre s'ouvre sur une partie théorique. Puis une partie applicative, toujours basée sur Linux, décrit de manière simplifiée l'implémentation faite de ces concepts au sein du noyau Linux, ainsi que les primitives systèmes qui leur sont attachées. Des exemples de programmation illustrent l'emploi de ces primitives.

J-P ARCHIER

IPv6

éditions ENI, à paraître

IPv6 modifie en profondeur le réseau et le protocole fondamental. L'auteur revient tout d'abord sur les concepts d'IPv4 et ce qu'apporte réellement IPv6 et l'ensemble des fonctionnalités de cette évolution que l'on ne comprend pas toujours très bien. L'auteur revient notamment sur les conséquences d'IPv6 sur la sécurité, le routage réseau et la gestion réseau dans les applications.





User Group du mois

LeMug.fr : la communauté MySQL

Le MySQL User Group francophone ou « LeMug.fr » est une association loi de 1901, à but non lucratif. Le projet a été lancé en septembre 2008, par Pascal Borghino, Olivier Dasini, et Damien Seguy afin de répondre à une demande récurrente de la communauté MySQL française : fédérer les ressources et les compétences de la base de données open source la plus utilisée.

Le but principal de l'association est donc de promouvoir les technologies relatives à la base de données MySQL (les bases de données MySQL, MariaDB, Percona server, Drizzle,...) auprès des professionnels et de participer à son développement, en France et dans les pays Francophones. Les membres de l'association sont régulièrement présents sur les événements touchant son activité principale, en France et à travers le monde. La majorité de ces membres sont francophones et travaillent dans différentes sociétés ou groupes impliqués dans l'Open



Source ou des sociétés commerciales. LeMug.fr organise régulièrement des rencontres techniques (Meetups), Apéros, Barcamps... ouvertes à toutes et à tous (développeurs, administrateurs de bases de

données, administrateurs système, directeurs techniques,...). Le but étant d'allier l'utile à l'agréable afin de stimuler la communauté, favoriser les échanges et partager les retours d'expériences. En effet, être proche et à l'écoute des utilisateurs permet de tirer le meilleur parti de ces technologies, et de faire émerger les tendances lourdes pour les



prochaines années. Un programme ambitieux et à la taille de la communauté française.

Pour rester au fait de la riche actualité de l'écosystème MySQL, n'hésitez pas à consulter le site officiel du MUG: <http://www.lemug.fr>. Mais aussi les sites partenaires, comme: planète MySQL (<http://fr.planet-mysql.com>), dasini.net - Journal d'un expert MySQL: (<http://dasini.net/blog/>) ou Db newz: (<http://www.dbnewz.com/>). Une entraide est bien entendue présente entre les utilisateurs sur lesquels l'association s'appuie directement à travers un forum sur le club <http://www.developpez.com>.

Christophe Villeneuve & Olivier Dasini

Mai

Dev Camps de Microsoft

Microsoft organise les Dev Camps entièrement orientés développeurs et accessibles en ligne. Le format se veut court, 1 heure, et apporter un maximum d'informations.

Pour mai, voici les dates pour les Dev Camps :

- Windows 8 Camp : 2 mai à Nantes, 10 mai à Bordeaux, 15 mai à Rennes
- Developer road show : 10 mai à Rennes, 16 mai à Toulouse, 24 mai à Marseille
- Azure Series : 25 mai, live meeting

Le **24 mai prochain**, l'après-midi du développement au campus Microsoft près de Paris sur le thème, Hadoop + Azure. Tout pour démarrer avec Hadoop sur Windows Azure (modèle de développement, déploiement, SDK et API, etc.).

Pour en savoir plus :

<http://msdn.microsoft.com/fr-fr/events/hh824685>

Rendez-vous chez les tontons flexeurs

Le 10 mai à partir de 19h, les tontons flexeurs organisent une session spéciale « 3d pour le web, le mobile et nos interfaces ». A cette occasion, le créateur du moteur Minko sera présent.

Venez donc nombreux Jeudi **10 Mai à 19h** participer à cet événement.

Pour ne pas changer une équipe qui gagne, ce sera salle TD6 au premier étage de l'ENSEIRB-MATMECA.

Site : <http://ttfx7.eventbrite.com/>

Juin

Apps Foundry Contest : un grand concours de dev de 36 heures !

Du 17 au 19 juin, participez à un grand concours de développement, avec une partie conférence technique. Le but : créer une application en 36 heures incluant le code, le design, la créativité, la performance... Bref, aux développeurs de montrer leurs talents. Chaque équipe sera composée de 4 personnes. Le développement se fera pour iOS, Android, Blackberry, Tizen / Bada, Windows Phone.

Détail important : aucune application flash ne sera acceptée... Chaque équipe devra présenter son projet. Le premier prix est un accès à la conférence ICT Spring du 20 juin. La compétition aura lieu au Luxembourg. Attention : inscription payante (gratuit pour les étudiants)

Site : <http://www.apps-foundry-contest.org>

agenda

MAI

- Le 03 mai, siège Oracle France 15, Boulevard Charles De Gaulle 92700 Colombes **Atelier Oracle Database pour les développeurs**. Un atelier concret consacré aux derniers outils de développement pour les bases de données Oracle.

http://www.oracle.com/webapps/events/ns/EventsDetail.jsp?p_eventId=143892&src=7307403&src=7307403&Act=350

- Du 11 au 13 mai, Paris 1er à l'IESA Multimédia [Campus Opéra]. **Weekend BeMyApp spécial Android**. Il s'agira de coder des applications pour le premier autoradio connecté capable d'accueillir des apps : le Parrot's Asteroid <http://fr.bemyapp.com/blog/>

JUIN

- Du 05 au 06 juin, Cité Internationale Universitaire, **Forum PHP 2012** :

Le Forum PHP est le plus grand événement national consacré à la plate-forme PHP et aux technologies Web. <http://www.afup.org/forumphp>

- Le 14 juin, Paris 13e, MAS, **CMSday**. Le premier événement majeur, en France et en Europe, dédié à la gestion de contenu open source. Plus de 20 éditeurs et communautés de CMS ont été invités à participer à de nombreux échanges et présentations. <http://www.cmsday.fr/>

- Du 19 au 21 juin, CNIT, Paris La Défense, **Solutions Linux 2012**. L'événement annuel incontournable des professionnels de l'open source et du logiciel libre annonce sa quatorzième édition. <http://www.solutionslinux.fr>

Etes-vous un développ

© i-stockphoto/ Ricardo Infante Alvarez

peur écologique ?

Développement, économie d'énergie, et éco-responsabilité, sont-ils compatibles ? A Programmez !, ce n'est pas la première fois que nous mettons en avant le GreenIT ou la programmation « écologique ». Sujet trop immature il y a 2 ans, aujourd'hui, avoir une démarche éco-responsable dans le logiciel est devenu un critère, même s'il est encore loin d'être généralisé.

Cependant, si le matériel informatique a fait d'importants progrès dans la consommation énergétique en réduisant parfois drastiquement celle-ci, il reste encore un important travail à réaliser. Et la multiplication des terminaux, la puissance toujours plus grande des processeurs multi-cœurs, posent de nouveaux défis à cette volonté de réduction énergétique. Dans certains cas, il est facile de réduire son impact écologique, par exemple avec une impression plus réduite des documents.

Le greenIT est-il encore lié aux grandes entreprises ? Devoteam, qui a publié récemment une nouvelle étude complète sur le sujet, est sans équivoque : « *Par ailleurs, 77% des personnes travaillant en PME avouent ne pas connaître le Green IT contre seulement 4% des personnes travaillant dans une grande entreprise. De fait, les PME, de par leur taille et leurs enjeux disposent de moins de*

capacités de retour sur investissement sur ces sujets... »

Il faut se dire que le logiciel reste le point faible. Malgré quelques cures d'amincissement, le logiciel reste toujours trop gros, trop gourmand. Mais cette course à la puissance, à la ressource, est poussée par les utilisateurs qui en veulent toujours plus. Or, chaque nouveauté majeure (interface 3D, vidéo HD, technologie vocale) nécessite de la puissance machine et donc de l'énergie. Or, recharger son smartphone chaque jour consomme plus que le faire une fois tous les trois jours... La lourdeur de certains langages, nous pensons aux langages non natifs, pose aussi un réel problème sur les ressources exigées et donc la consommation de celles-ci. Mais on peut aussi se demander si le greenIT atteint le développement et le logiciel en général ?

Globalement, la réponse est non avec de fortes nuances, car il existe des bonnes pratiques, des outils, pour tracer les consommations excessives et pour développer plus « vert ». Malgré tout, point de miracle. A chaque développeur d'être plus rigoureux dans son code, à utiliser plus strictement les ressources et à mettre en œuvre celles nécessaires, ni plus, ni moins. Mais ces réflexes sont étrangers à la plupart des développeurs.

Ce mois-ci, nous vous proposons de découvrir quelques méthodes et outils pour être moins gourmand !

François Tonic

Eco-conceptions des logiciels et Green Patterns



Les logiciels subissent la loi inexorable de l'entropie : Plus il y a de ressources, plus le logiciel va les occuper. Double impact pour l'environnement : toujours plus de consommation et obsolescence programmée du matériel accélérée. Mais que peut faire le développeur ?

OBÉSICIEL ET IMPACT DU LOGICIEL

> Les constats

L'empreinte écologique des équipements informatiques se concentre lors de leur fabrication et de leur fin de vie. La fabrication d'un ordinateur émet par exemple 70 à 100 fois plus de CO2 qu'un an d'utilisation en France. Le geste essentiel du Green IT consiste donc à allonger la durée d'utilisation des équipements. Au-delà d'une durée de garantie trop courte, la couche logicielle est le principal facteur d'obsolescence des équipements informatiques. En effet, la puissance nécessaire pour exécuter un logiciel double d'une version sur l'autre. Si bien qu'en 25 ans, la durée moyenne d'utilisation d'un ordinateur a été divisée par 3, passant de 10 à 3 ans. Réduire les besoins en ressources matérielles (cycles CPU, mémoire vive, etc.) des logiciels est incontournable pour réduire leur consommation électrique (et donc les émissions de CO2 liées) et allonger la durée de vie active des équipements informatiques (qui concentrent les nuisances environnementales). Ce phénomène que suit le logiciel s'appelle l'obésiciel et on peut le voir dans tous les logiciels. Mais est-ce vraiment de la faute des développeurs ? Et si le logiciel suivait une loi d'évolution non maîtrisable. Il faut regarder du côté du matériel pour comprendre ce qui se passe dans le logiciel. Depuis 1965, une loi prédit correctement l'évolution du matériel : la loi de Moore. Elle se vérifie année après année. La Loi de Moore est simple : « Quelque chose » (le nombre de processeurs, la performance, la vitesse...) double tous les 18 mois. Cette loi a été critiquée, reprise, interprétée et modifiée, mais toujours vérifiée. Cependant avant d'être une loi technologique, c'est une loi économique. C'est ce que Gordon Moore appelle lui-même une prophétie auto réalisatrice. La Loi de Moore est un leitmotiv pour l'industrie électronique... qui met les moyens marketing et techniques pour atteindre les objectifs de la loi. L'impact pour les développeurs est que comme on sait que de toute manière on aura toujours plus de puissance, il n'y a pas besoin d'optimiser le code. Et pour fermer ce cercle non vertueux, cela fournit aux fabricants de matériel une excuse pour suivre la loi de Moore. Cette évolution continue du matériel amène un autre problème et renforce l'obésiciel : même si le matériel nous offre des capacités de plus en plus importantes (comme la parallélisation avec des processeurs multi-cœurs ultra efficaces), pourrions-nous utiliser toute la capacité de ces technologies ? Aurons-nous le temps et les moyens d'optimiser notre code ?

> Obésiciel

Dans tous les cas, le logiciel continue à croître, à devenir plus lourd et à consommer plus. La loi de Lehman énonce que l'évolution d'un logiciel est inévitable, pour différentes raisons. Il a énoncé 8 lois entre 1964 et 1996, voici 3 lois qui nous concernent :

- 1 – Modifications perpétuelles** : Un logiciel doit constamment être adapté, sinon il devient de moins en moins utile.
 - 2 – Complexité croissante** : Au fur et à mesure que le logiciel évolue, sa structure a tendance à se complexifier.
 - 3 – Continuelle augmentation** : le contenu du logiciel doit continuellement augmenter pour satisfaire les besoins des utilisateurs.
- Est-ce encore une fois une prophétie auto-réalisatrice ? Ce qui est clair c'est que pour l'instant aucun projet (Microsoft Windows, GNU/Linux, Firefox ...) n'a cassé ces lois. Structure trop complexe et trop importante, contraintes économiques, intérêt à réutiliser du code déjà développé ...

> Processus de développement efficient vs Quick & Dirty

Si l'on regarde de plus près, de nombreux développeurs (moi compris !) appliquent des méthodes que l'on peut appeler « quick and dirty » : rapides et sales. Ces solutions permettent de programmer rapidement car l'on ne s'occupe pas d'appliquer les bons design patterns, de réfléchir à l'architecture... Solutions très répandues, elles permettent d'arriver aux résultats voulus rapidement et de corriger des problèmes en cas de crise. L'exemple le plus célèbre est celui de MS-DOS, dont le nom initial signifiait tout simplement : Q-DOS : « Quick&Dirty Operating System ».

De nombreuses discussions dans le monde des développeurs font rage sur l'intérêt du quick and dirty. Pourquoi passer du temps à réfléchir sur des bonnes pratiques alors que le client veut une simple application ? D'autant plus que certains développeurs ont eu la malheureuse expérience d'utiliser des bonnes pratiques sans que le logiciel ne satisfasse le client.

Le problème de cette pratique est que cela crée une dette technique. La comparaison de cette dette technique à une dette financière montre qu'un code mal conçu requiert le paiement d'intérêts dans un délai plus ou moins long. Quels intérêts ? Un effort de maintenance, un effort d'évolution ou même un travail pour nettoyer le code. Sans parler de la nécessité de changer les ordinateurs (poste de travail notamment) à chaque nouvelle version. Dans tous les cas, cette dette contractée dès l'écriture du code devra être payée par l'entreprise ou le particulier qui utilise le logiciel. Et il en va de même pour l'environnement.

ECO-CONCEPTION ? GREEN PATTERNS ?

Selon l'institut Green Software Engineering, un logiciel durable est un logiciel qui n'a pas d'impact négatif direct (énergie, ressources pour produire / supprimer le logiciel) ou indirect (phase d'utilisation du logiciel) sur l'économie, la société, l'homme et son environnement. L'ingénierie logicielle durable est ainsi l'art de définir et développer des logiciels de manière à ce que les impacts positifs et négatifs sur



le développement durable tout au long de son cycle de vie soient évalués, documentés et optimisés.

Eco-concevoir un logiciel, c'est donc développer une application en maîtrisant son impact sur l'environnement mais aussi sur la société et sur l'économie. L'impact sur l'environnement du logiciel n'est jamais direct car le logiciel fonctionne obligatoirement avec du matériel. Cependant son effet sur le matériel et donc l'environnement n'est pas négligeable : obsolescence ressentie, consommation électrique, ondes radios... L'éco-conception du logiciel analyse donc tous ces éléments et préconise des bonnes pratiques pour les maîtriser. Les aspects sociaux ne doivent pas être négligés dans cette conception. Si le logiciel peut être un vecteur de lien social important (amélioration des liens et meilleure communication par exemple), il peut à l'inverse être une cause de création d'inégalités sociales et d'exclusion. En effet, même si des normes et des bonnes pratiques existent quant à l'accessibilité des sites et des logiciels, peu de logiciels les intègrent. Sur l'exclusion des populations du numérique, même si l'accès au matériel et à internet s'est amélioré, il reste une barrière pour accéder aux mondes numériques. Le logiciel en est la principale (interface complexe, notions abstraites...).

Le dernier pilier du développement durable, souvent oublié, voire opposé au développement durable lui-même, est l'économie. L'équité du modèle économique entre dans l'analyse qui doit être faite dans l'éco-conception, avec par exemple une réflexion sur des pratiques de coûts de licence (parfois prohibitifs), pour les utilisateurs. La productivité du processus de développement logiciel est un autre axe de travail. Dans un domaine où plus de 50 % des projets sont retardés, annulés ou qui dépassent les budgets, on peut se poser la question d'une viabilité économique.

LANGAGE ET ARCHITECTURE

> Natif ou managé ?

La guerre fait rage en ce qui concerne les langages : on pourrait penser qu'un langage natif comme le C serait plus performant car plus proche de la machine. Oui et non... Oui car il est certain que les langages natifs sont optimisés par rapport à la machine et contrairement à d'autres langages, ils ne passent pas par un runtime ou un interpréteur. Non, car tout dépend de l'application que vous allez développer. Par exemple certaines boucles de calcul seront tout aussi performantes, voire plus, en java qu'en C. De la même manière, le PHP, même si peu performant, permet de mettre en place des applications très rapidement et donc à faible coût.

L'important est de choisir une technologie adaptée à l'utilisation : une application qui sera déployée à grande échelle nécessitera un langage optimisé comme le C++ par exemple. De plus, dans le cas de langage managé ou interprété, le développeur devra connaître les mécanismes des couches basses et du runtime. Sa façon de coder en sera améliorée ! Ce n'est pas parce que l'on vous donne une Ferrari que vous ne devez pas regarder la consommation et avoir une éco-conduite ! Vous pourrez par exemple utiliser des API de marshalling pour mieux gérer le comportement du logiciel.

Exemple de Facebook

Sans quoi, si votre logiciel rencontre le succès (ce que je vous souhaite !), vous ressentirez la dette technique. Ça été le cas pour Facebook avec le PHP. L'impact de PHP était énorme en termes de performance et demandait des machines supplémentaires. En effet, PHP n'est pas le plus performant. Certains modules comme

opcache permettent d'améliorer la performance mais n'améliorent pas la vitesse d'exécution. En effet, ils permettent uniquement de stocker les opcodes Zend lors de la phase de compilation avant chaque requête. La solution adoptée par Facebook a été de transformer le code PHP en code C++. Le projet HipHop for PHP est alors né ! HipHop fonctionne de la manière suivante :

- Analyse statique pour détecter les déclarations et les dépendances,
- Transformation des types
- Génération du code en C++

Le projet n'est pas encore très utilisé et impose des contraintes (pas de support d'Apache par exemple). Si vous souhaitez l'évaluer : <https://github.com/facebook/hiphop-php/wiki/>

> Architecture : comment répartir les calculs

Au-delà du langage, la répartition des calculs est un axe d'amélioration très important en ce qui concerne l'éco-conception. En effet, les calculs sont de plus en plus répartis sur différents matériels : serveur web, architecture client/serveur, mobile... Il faut placer les calculs dans des plateformes qui ont une efficacité énergétique la plus forte. Dans le cas des architectures web et client/serveur, il faut mettre le plus de calcul possible côté serveur. Les serveurs sont en effet plus maîtrisés en ce qui concerne l'énergie. En parallèle, on va devoir limiter le nombre de requêtes entre les éléments.

ON OUBLIE LE % CPU

> Allez jusqu'au niveau du tick processeur

Dans les architectures processeurs actuelles, des mécanismes permettent aux processeurs d'entrer dans des états de veille plus ou moins profonds. Le gain global sur la consommation de la plateforme en est amélioré. Cependant, plus le processeur sera dans un état de veille profond, plus la sortie de cette veille sera coûteuse (appelée transition C-State) Il est donc nécessaire d'appliquer certaines bonnes pratiques :

- Pour les états où le processeur est en Idle, éviter toute activité (polling, tâche de fond, timers...)
- L'activité doit être rassemblée le plus possible pour permettre au processeur d'entrer en veille profonde.
- Éviter les transitions fréquentes C-state

Voici un exemple des consommations dans différents états pour un processeur Intel Core 2 Duo

C-state	Max Power Consumption
C0 (busy wait)	35 Watts
C1	13.5 Watts
C2	12.9 Watts
C3	7.7 Watts

L'impact des transitions C-state est d'autant plus important sur des architectures multicœurs si le parallélisme n'est pas traité correctement. En effet, si le traitement est géré par les cœurs sans réelle concurrence, les cœurs vont alternativement être en veille et le nombre de transitions sera important. La consommation pourra être encore plus grande que sur un simple cœur. Il faut donc maximiser les exécutions concurrentes sinon il faut exclusivement utiliser un cœur unique.

Les timers précis ont un impact sur ce fonctionnement. En effet, sur Windows par exemple, le timer périodique cadence le scheduler à 15,6 ms. Certaines applications (comme les applications multimé-

dias) augmentent ce timer pour améliorer la performance. Le nombre de transitions est alors augmenté de façon très importante et donc la consommation aussi. Il est donc préférable de ne pas utiliser des timers précis (ou seulement si nécessaire).

Sur la déclaration des timers, Windows propose à partir de Seven une nouvelle API Timer Coalescing permettant aux logiciels de préciser qu'ils n'ont pas de contraintes fortes sur la période du timer. Ceci permet au système de regrouper les traitements et d'éviter des transitions C-state trop nombreuses.

```
SetWaitableTimerEx(
    __in      HANDLE hTimer,
    __in      const LARGE_INTEGER *lpDueTime,
    __in      LONG lPeriod,
    __in_opt  PTIMERAPCROUTINE pfnCompletionRoutine,
    __in_opt  LPVOID lpArgToCompletionRoutine,
    __in_opt  PREASON_CONTEXT WakeContext,
    __in      ULONG TolerableDelay
);
```

Il est conseillé de mettre le paramètre TolerableDelay à 32 ms, soit deux périodes de tick Windows. Cependant si votre timer à une période de plusieurs secondes, vous pouvez augmenter la tolérance. Le scheduling et la gestion des timers d'une application est très importante pour la consommation. Deux applications avec la même charge CPU pourront avoir des gestions totalement différentes et donc des consommations non égales. Au vu du fonctionnement des C-states et des ticks, il est préférable d'effectuer des traitements plus longs et moins régulièrement que des traitements courts et à chaque tick système. Vous aurez la même charge CPU mais pas la même consommation !

> Evaluation du comportement sous Intel et Windows

Voici une méthodologie pour analyser le comportement C-state et tick sous Windows et pour Intel 5 Series Mobile Chipset et Intel Core i7/i5/i3 Processor.

Vous avez besoin du logiciel Intel Battery Life Analyzer http://download-center.intel.com/Detail_Desc.aspx?agr=Y&DwnldID=19351. Lancer l'application en tant qu'administrateur. Pour lancer le profiling, cliquez sur **Capture All**. Laissez le logiciel effectuer les mesures. La fenêtre CPU C-state vous donne le pourcentage de temps passé dans les différents états [Fig.1]. Dans le cas présent, le processeur passe beaucoup de temps dans les états actifs (CO-C1). Une analyse plus précise est nécessaire pour savoir pourquoi le pourcentage n'est pas inférieur à 5%. La fenêtre software activity analysis permet d'avoir plus de détails : [Fig.2]. La colonne Timer Tick, indique les applications qui ont demandé un changement de timer. Ici Chrome et svchost. Sous Windows 7, vous pouvez faire la même analyse en lançant en mode administrateur la ligne PowerCfg /energy. Un rapport va être généré pour vous indiquer qui a demandé un changement de timer. Cela vous permettra de voir plus précisément le process responsable. La colonne Cswitches from Idle indique les processus responsables d'un changement important de C-state. La fenêtre Idle Windows Analysis vous permettra de plus d'analyser combien de temps votre application passe en idle. Une fois l'application consommatrice identifiée, il faut effectuer une analyse plus précise avec **Microsoft Windows Performance Analyzer**. Ouvrez la

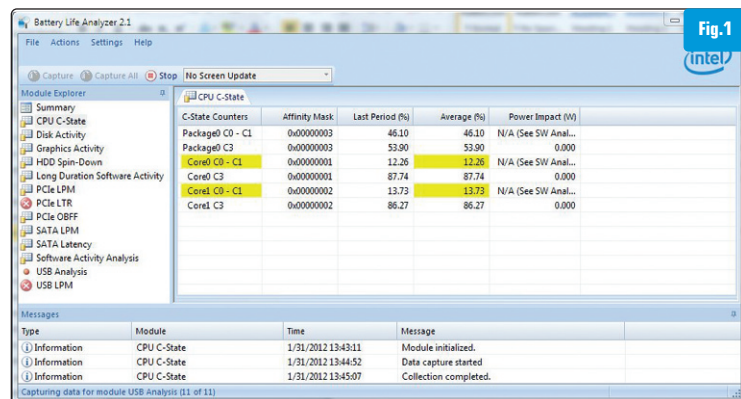
fenêtre de commande en mode administrateur. Lancez la commande **xperf -on DiagEasy -stackwalk CSwitch**. Laissez le système en idle. Arrêtez l'enregistrement avec **xperf -d output.etl**. Ouvrez le log avec **xperfview output.etl**. Activez le décodage des symboles "Trace" menu > "Load Symbols". Vous pourrez ainsi analyser précisément votre application en fonction des process, des CPU... [Fig.3]. Si vous êtes sous Linux, l'outil Ptop vous permettra de réaliser une analyse des transitions Cstate. L'application vous donnera le temps passé dans les différents C-state et le top 10 des applications réveillant le processeur [Fig.4].

EFFICACITÉ DES DONNÉES : LOCALITÉ ET COÛT DES DONNÉES

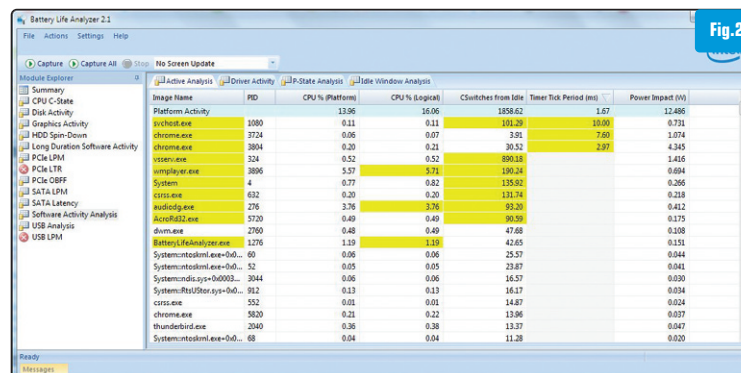
> Diminuer le coût des données

L'explosion des données est une des causes de l'obésité des logiciels. Il y a de nombreuses raisons, par exemple, la taille des données n'est pas forcément celle que l'on croit quand on développe. Les contraintes imposées par les architectures matérielles obligent les compilateurs et les environnements d'exécution à ajouter des données aux données utiles et à les réorganiser en mémoire. L'alignement et la taille d'adressage sont par exemple des contraintes dimensionnantes. Le padding en est un exemple. Les compilateurs doivent suivre les restrictions d'alignement des données définies par les microprocesseurs. Cela signifie que les compilateurs doivent ajouter des octets de bourrage (padding) entre les données définies par l'utilisateur afin que les restrictions imposées par le microprocesseur ne soient pas violées. Voici un exemple sur la taille des structures :

```
struct {
    bool b;
    double d;
```



Battery Life Analyzer / Cstate



Battery Life Analyzer / Software

```
short s;
int i;
};
```

La structure va occuper: 1 (bool) + 7 (padding) + 8 (double) + 2 (short) + 2 (padding) + 4 (int) = 24 bytes. Alors que pour :

```
struct {
double d;
int i;
short s;
bool b;
};
```

La taille sera de 8 (double) + 4 (int) + 2 (short) + 1 (bool) + 1 (padding) = 16 bytes.

Ce surcoût de taille est encore plus important pour les architectures 64 bits. Les applications ont généralement de nombreux objets de petites tailles et des pointeurs et cela oblige à ajouter des bits de bourrage, des données supplémentaires et donc à augmenter la mémoire nécessaire. Cette pratique ne s'applique pas uniquement aux langages natifs mais aussi aux langages managés. Mais cela va nécessiter d'utiliser des API spécifiques. En .net par exemple, on utilisera la classe Marshal pour accéder à des fonctions d'allocation non managée de la mémoire si l'on veut optimiser la déclaration des structures :

```
[StructLayout(LayoutKind.Sequential, Pack = 1)]
[StructLayout(LayoutKind.Explicit)] avec des champs [FieldOffset(0)] devant chaque élément.
```

On retrouve les mêmes contraintes dans les langages comme java avec des machines virtuelles. Selon IBM TJ Watson Research Center, le surcoût des données (data overhead) est entre 50 et 90% :

- pour un double de 8 octets, JVM impose 16 octets d'overhead. la donnée ne compose donc que 33% des 24 octets.
- pour un treemap de 100 entrées, le premier overhead est de 48 octets et de 40 pour chaque entrée. La taille est donc 3,9kB
- Pour TreeMap<Double, Double> (100 entrées), l'overhead atteint 82% Une implémentation alternative serait 2 doubles et l'overhead ne serait que de 2%. Cela ne permet plus d'avoir les mêmes fonctionnalités mais cela montre bien les gains possibles.

> Localité des données

Un autre principe d'efficacité est la distance aux données (principe de localité). Plus la donnée va être éloignée du processeur, plus l'accès à cette donnée sera coûteux en temps et en énergie. Cette dis-

tance est une distance virtuelle que l'on peut évaluer en nombre d'instructions du processeur et de temps de latence pour accéder à cette donnée. La distance va dépendre de l'architecture et de la technologie. Accéder à une donnée en cache sera par exemple moins coûteux qu'une donnée sur disque dur.

Voici quelques distances classiques :

- Registre - 1 cycle - Jusqu'à 10 mots par cycle - CPU
- Cache L1 - 3 cycles - Jusqu'à 2 mots par cycle - CPU
- Cache L2 - 10 cycles - 1 mot par cycle - circuit externe
- Mémoire RAM - ~100 cycles - 0,5 mot par cycle - circuit externe
- Disque dur - ~ 1 million de cycle - 0,1 mot par cycle - périphérique
- Bande externe - long - peu de mots par cycle - périphérique
- Donnée réseau - très long - très peu de mots par cycle - périphérique

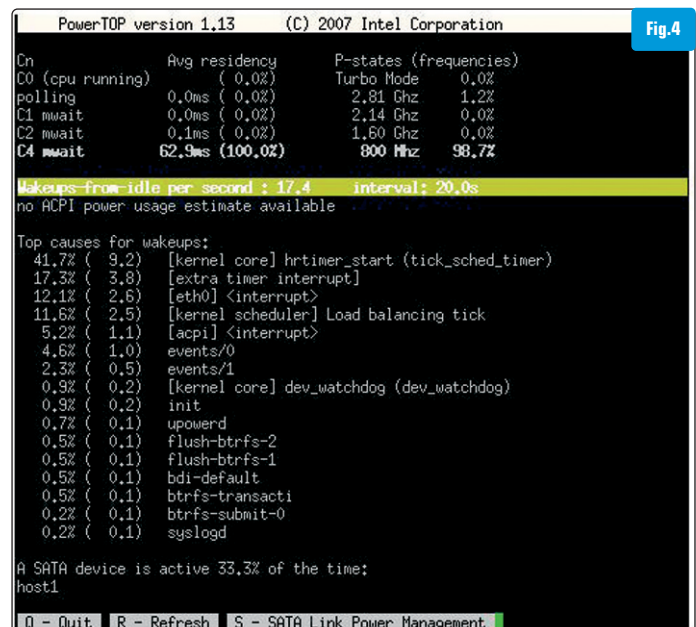
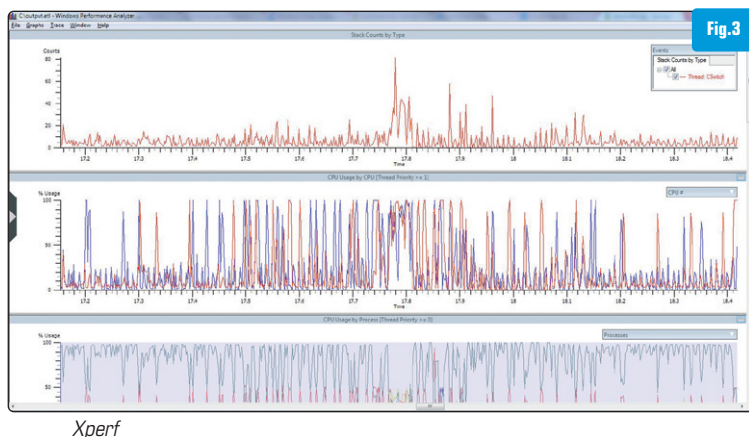
L'architecture de la mémoire est donc une des clés pour optimiser l'efficacité de la gestion des données. Plus les données seront proches, moins le calcul consommera d'énergie. Pour cela, plus les données seront traitées dans le cache, moins le code passera du temps à aller chercher les données plus loin (en ram par exemple). Dans ce cadre, il faut éviter ce que l'on appelle les pertes de caches (Cache misses). Lorsque le CPU recherche une donnée dans un des caches et qu'elle n'y est pas, alors le processeur recharge dans le cache la donnée. Cela est alors coûteux. Les bonnes pratiques peuvent être utilisées en se calquant sur le développement durable et le principe des 3 R (http://www.research.scea.com/research/pdfs/GDC2003_Memory_Optimization_18Mar03.pdf) :

- Réarranger : modifier les patterns pour augmenter la localisation spatiale
- Réduire : des formats plus petits et plus intelligents, de la compression de données
- Réutilisation : augmenter la localisation temporelle (et spatiale)

EFFICACITÉ DE CALCUL : BOUCLE ET AUTRES Puits D'Énergie

Efficacité d'un processus

L'efficacité d'un processus ou d'un travail est la caractéristique à effectuer la même tâche avec moins de ressource. Dans le cas d'un calcul informatique la ressource utilisée est le processeur. Elle peut être caractérisée par différentes notions : temps processeur,



nombre de ticks d'horloge, charge CPU... mais dans tous les cas, elle représente une information sur la quantité de ressource du processeur utilisée par le calcul. En utilisant moins de ressource CPU, la tâche sera réalisée en moins de temps et consommera donc moins d'énergie. Le deuxième effet bénéfique sera que les ressources du processeur seront libérées plus tôt soit pour effectuer un autre calcul, soit pour passer en veille.

Boucle de calcul

Les boucles de programme sont des mécanismes très répandus car ils permettent de réaliser de nombreuses opérations simplement. Cependant, ils sont très consommateurs en ressources car ils monopolisent le processeur. De plus le nombre de boucles à réaliser est souvent très important et même parfois infini (et cela peut causer des bugs). La simplicité des boucles doit être maniée avec précaution car l'impact de l'implémentation d'une boucle peut être très négatif. Voici quelques moyens d'optimisation :

Sortie de la boucle : Il faut prévoir au plus tôt une condition de sortie permettant d'arrêter l'itération. Code d'erreur, actions utilisateurs... Toute condition permettant de sortir de la boucle au plus tôt permettra d'éviter des traitements inutiles. Exemple simple mais efficace :

```
For (i=0 ;i<5000 ;i++)
{
// action
If (sortie = true) // condition de sortie
{i = 5000}
}
```

Fin de boucle optimisée : Le nombre d'itérations maximales d'une boucle est souvent choisi sans réflexion d'optimisation. La prise en compte du contexte et du besoin est donc à prendre en compte dans le choix du nombre d'itérations. Pourquoi mettre une boucle de surveillance au maximum possible (255 par exemple) alors que d'autres données peuvent être prises en compte (temps maximum de saisie, timeout existant...)

L'utilisation de boucles pour surveiller des données est à proscrire. Il faut privilégier l'utilisation d'interruptions ou de multitâche. Dans cette optique, on pourra par exemple mettre en place un microprogramme qui traite de la surveillance d'un périphérique et un autre qui traite les données. Cela permettra ainsi à l'OS de gérer de façon plus flexible l'ensemble de la fonction. Un code avec des `thread.wait(x secondes)` est un code pauvre !

Plus globalement, il est difficile de généraliser des bonnes pratiques sur l'utilisation des boucles. Il est donc nécessaire d'étudier la consommation de ressource et d'énergie pour les boucles avec des grosses itérations et des calculs importants.

Une technique est d'extraire le plus de traitement de l'on peut des boucles de calcul. La technique appliquée sur la déclaration des structures en Hot/Cold spitting en est l'exemple Il est en effet possible d'améliorer la performance de la boucle et de réduire les pertes de cache en divisant les structures selon la fréquence d'accès : hot (fréquemment accédé) et cold (rarement accédé). Cela permet aux données « hot » de rester plus souvent dans le cache et d'être traitées ensemble. La structure S suivante :

```
Struct S
{
void *key;
S *pNext;
```

```
int count[10]; // Tableau de compteur d'erreur
S2 *pCold;
}
```

Sera divisée en hot field :

```
Struct S1
{
void *key;
S *pNext;
S2 *pCold;
}
```

Et Cold field

```
Struct S2
{
int count[10];
}
```

La technique de loop unrolling peut être intéressante car elle augmente la performance des boucles et évite les pertes de caches (cache misses) :

```
for (int i=0; i<n; i++)
{
somme += data[i];
}
```

Deviendra :

```
for (int i=0; i<n; i+=4)
{
somme1 += data[i+0];
somme2 += data[i+1];
somme3 += data[i+2];
somme4 += data[i+3];
}
somme = somme1 + somme2 + somme3 + somme4;
```

Attention cependant car la plupart des processeurs et des compilateurs réalisent ces optimisations. Il est donc nécessaire de bien connaître les techniques d'optimisation du processeur. Une autre manière est de tester les pratiques dans votre code, et si l'optimisation n'est pas concluante de l'oublier sur votre plateforme ! D'autres techniques sont très intéressantes : Par exemple avec les processeurs Intel Core 2 Duo, la fonctionnalité Loop Stream Detector (LSD) rend le loop unrolling inutile. Le LSD permet de détecter quand le logiciel est dans une boucle et annule la prédiction de branche. L'utilisation de cette fonction se fait cependant sur des boucles de faible taille.

APPLICATION « ENERGY-AWARE »

> Principes

Les applications doivent prendre en compte la gestion de l'énergie implémentée par le système d'exploitation. Votre application sera alors "Power-Aware" (consciente de la gestion d'énergie). Si cette gestion n'est pas optimisée, le logiciel perturbera la gestion de l'énergie du PC et la consommation d'énergie sera très importante. De plus, cela pourra créer un sentiment négatif de l'utilisateur envers le système global et envers la gestion de l'énergie.

Le programme doit donc être capable de :

- Supporter les événements de changement de mode d'énergie : par exemple prendre en compte le passage en mode veille prolongée

- Adapter le comportement à la configuration de la gestion d'énergie et l'état courant du processeur : par exemple si le processeur diminue sa performance, le logiciel doit s'adapter.
- Intégrer la politique de gestion énergétique du système

> Application sous Windows

Sous Windows, les messages PBT_APM permettent de connaître les événements de la gestion de l'énergie. Voici par exemple des événements disponibles :

PBT_APMBATTERYLOW – Batterie faible

PBT_APMQUERYSUSPEND – Demande de permission pour la veille.

PBT_APMQUERYSUSPENDFAILED – Demande de veille refusée

PBT_APMSUSPEND – Mise en veille

Sous .Net en C#, voici un exemple pour prendre en compte le contexte :

```
const int WM_POWERBROADCAST = 0x0218;
const int PBT_APMQUERYSUSPEND = 0x0000;
const int PBT_APMQUERYSTANDBY = 0x0001;
const int PBT_APMQUERYSUSPENDFAILED = 0x0002;
const int PBT_APMQUERYSTANDBYFAILED = 0x0003;
const int PBT_APMSUSPEND = 0x0004;
const int PBT_APMSTANDBY = 0x0005;
const int PBT_APMRESUMECRITICAL = 0x0006;
const int PBT_APMRESUMESUSPEND = 0x0007;
const int PBT_APMRESUMESTANDBY = 0x0008;
const int PBT_APMBATTERYLOW = 0x0009;
const int PBT_APMPOWERSTATUSCHANGE = 0x000A;
const int PBT_APMOEMEVENT = 0x000B;
const int PBT_APMRESUMEAUTOMATIC = 0x0012;
protected override void WndProc(ref Message m)
{
    base.WndProc(ref m);
    if (WM_POWERBROADCAST == m.Msg)
    {
        // Réalisez les actions en fonction des messages.
    }
}
```

En fonction des événements, vous pouvez alors agir : enregistrer le contexte, sauvegarder un fichier, avertir l'utilisateur... Afin de debugger les changements du système, vous pouvez utiliser Spy++ sous Visual Studio :

Allez sous **Espionner > Enregistrer des messages**. Sous l'onglet Messages, désélectionnez tous les messages et sélectionnez **WM_POWERBROADCAST**.

Vous obtenez alors tous les messages associés à la gestion de l'énergie et pouvez déboguer le comportement de votre application en fonction [Fig.5].

Le fonctionnement est le même sous Linux. `acpi_listen` par exemple permet de surveiller les événements des mécanismes de gestion de l'énergie.

APPLICATION « CONTEXT-AWARE »

> Principes

Un logiciel efficace dans la prise en compte du contexte est un logiciel qui s'adapte à son environnement. Sans cette adaptation, le logiciel répondra difficilement aux sollicitations extérieures et ne se mettra pas

dans un bon état de consommation. L'exemple concret est le changement de la luminosité de l'écran lors du branchement/débranchement de l'alimentation. Une application doit donc répondre aux changements de l'environnement et cela demande de prendre en compte le plus possible les informations venant de l'extérieur du logiciel :

- Données issues d'autres logiciels
- Capteurs et actionneurs
- Données du système
- Hypothèses ou données sur l'utilisateur.

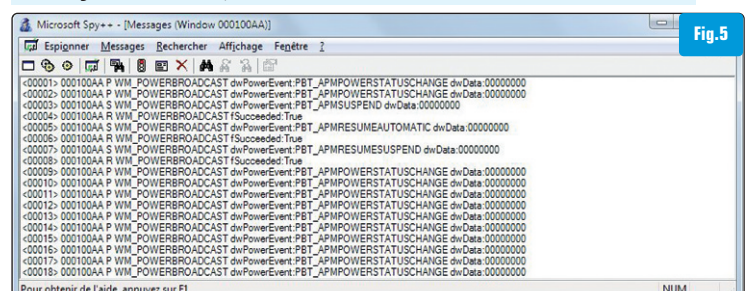
> Application aux fenêtres sous Windows

L'exemple concret est la minimisation d'une fenêtre. Lorsque mon application est active je fais des traitements (calcul, affichage...). Mais si l'utilisateur minimise la fenêtre, il n'a certainement plus besoin de tous les traitements (encore moins de la gestion de l'affichage). Le logiciel doit donc bien gérer ce changement de contexte. Sous Windows et .net on pourra surveiller le broadcast des messages **WM_WINDOWPOSCHANGED**, indiquant les événements sur les fenêtres. Pour java, voici une solution :

```
private void formWindowStateChange(java.awt.event.WindowEvent evt) {
    if(this.getState()==1){// Minimisé
        // Arrêter tout !
    }else if(this.getState()==0){// Maximisé
        // Relancer tout !
    }
}
```

Cette bonne pratique, peut-être appliquée dans les navigateurs internet. De nombreux script java, mises à jour et autres traitements fonctionnent en effet quand l'utilisateur minimise le navigateur ou visualise un autre onglet. Il est possible d'utiliser les événements `onblur` et `onfocus`. Voici un code qui permet de changer le background, à vous d'adapter en fonction du contexte !

```
function onBlur() { document.body.className = 'blurred'; };
function onFocus(){ document.body.className = 'focused'; };
if (/*@cc_on!@*/false) { // Si Internet Explorer
    document.onfocusin = onFocus;
    document.onfocusout = onBlur;
} else {
    window.onfocus = onFocus;
    window.onblur = onBlur;
}
</script>
<style type="text/css">
.focused #focused, .blurred #blurred { display: inline }
.focused #blurred, .blurred #focused { display: none }
.focused {
    background: white;
```



Spy++


```
color: black;
}.blurred {
  background: black;
  color: white; }
</style>
```

> SDK pour prendre en compte le contexte

Pour aider le développeur, des environnements de développement permettent de facilement récupérer ces contextes et ensuite adapter le fonctionnement. On peut citer à titre d'exemple les outils suivants chez Intel :

- Mobile Platform Software Developer Kit (MPSDK)
- Laptop Gaming Technology Developer Kit (Gaming TDK)
- Web 2.0 Technology Development Kit (Web 2.0 TDK)

CAS DE SITE WEB : QUE FAIRE ?

Les sites web ont un impact important sur la consommation, côté serveur mais aussi côté client, sans compter la consommation induite sur les éléments réseaux.

L'explosion des technologies comme le PHP ou des CMS comme Drupal n'a pas améliorée la situation. Une des premières recommandations est d'adapter la technologie ou l'outil que vous mettrez en place en fonction du besoin : A-t-on besoin de mettre DRUPAL en place pour afficher 5 pages ? Un site statique sera préférable et vous diminuerez la consommation par 6. Voici quelques bonnes pratiques parmi tant d'autres !

> Eviter de retailer les images

N'utilisez pas des images plus grosses que ce dont vous avez besoin dans le HTML. Cela va en effet nécessiter de retailer l'image pour rien.

```

```

Si votre image taille 500x500px, il faut la redimensionner en 100x100px

> Embarquer les images

L'encodage des images en base64 est une bonne pratique pour réduire les requêtes

```

```

Dans ce cas classique, lors du chargement de la page, le navigateur va envoyer une requête au serveur pour ensuite télécharger l'image. L'encodage de l'image sera alors :

```

```

Il n'y aura pas de requête supplémentaire et l'image sera embarquée dans le HTML. Cette pratique n'est valable que pour les petites images car elle augmente la taille de 50%. Voici un site permettant de convertir vos images en base64 : <http://www.motobit.com/util/base64-decoder-encoder.asp>

> Eval is evil

Eviter l'utilisation d'eval(). En effet, eval() évalue une chaîne de caractère en code. Cela amène donc une recompilation et une réinterprétation du code lors de l'exécution. Il en va de même pour setTimeout que l'on utilise parfois avec une chaîne de caractère plutôt qu'une fonction :

```
window.setTimeout('window.parent.alerte()', 1000);
```

sera plutôt utilisé comme cela :

```
window.setTimeout('window.parent.alerte()', 1000);
```

Idem pour la création de fonction :

```
new Function(...)
```

sera transformé en déclaration littérale

```
function() { ... } ou function myfunc() { ... }
```

Il en va de même pour preg_replace.

> Eviter les variables globales

Eviter les variables globales car elles seront dans des scopes différents et elles seront donc accessibles moins rapidement.

Adapter les sites mobiles aux mécanismes matériels. Pour les sites pour mobile, il est intéressant de connaître le comportement des caches des smartphones. Par exemple pour iPhone, les objets de plus de 15ko ne sont pas cachés. Cela veut dire qu'il est intéressant de dimensionner ses images et autres objets en fonction de cela.

> Maîtriser l'intégration

L'intégration des logiciels est une phase importante. Par exemple, dans le cas du couple PHP+Apache, compiler et linker PHP et APACHE en utilisant des bibliothèques liées statiquement améliorera de 30% la performance (sauf que ce n'est plus possible avec Apache 2...)

> Mesurer vos applications

La mesure physique est importante (voir plus loin dans l'article), mais une mesure simple de la performance est possible. <http://gtmetrix.com/> offre par exemple une solution web gratuite.

> Pour aller plus loin dans l'efficacité : exemple de node.js

Optimiser l'utilisation des architectures et des technologies avec les pratiques précédentes est une chose importante mais on va être confronté à des barrières technologiques et des goulots d'étranglement. En effet, l'architecture des serveurs avec en particulier la gestion des entrées-sorties et les modèles de communication concurrente limite les performances. Des solutions existent bien sûr pour contourner ces limitations en utilisant moins de ressource et en consommant moins d'énergie comme le framework Node.js ou les serveurs web légers tels que Nginx, lighttpd... [Fig.6].

Source : <http://blog.webfaction.com/a-little-holiday-present>

Vous avez deux solutions si vous souhaitez mettre en place un serveur tout simple : installer une stack complète apache+php... ou passer par Node.js. Voici par exemple le code du fichier server.js permettant de mettre en place le serveur :

```
var http = require('http');
http.createServer(function (request, response) {
  response.writeHead(200, {'Content-Type': 'text/plain'});
  response.end('Hello World\n');
}).listen(8124);
console.log('Server running at http://198.168.1.1:8124/');
```

Vous n'avez plus qu'à lancer le programme sous le programme node.js :

```
node server.js
```

Dans votre explorateur, vous avez alors à l'adresse <http://localhost:8888/> votre serveur. L'avantage de cette technologie est une mise en place très rapide et un serveur peu consommateur. Vous pourrez même le déployer sur des PC de type plugPC ne consommant que quelques watts.

MESURE : PROFILING ET MESURE DE LA CONSOMMATION

> Mesure physique

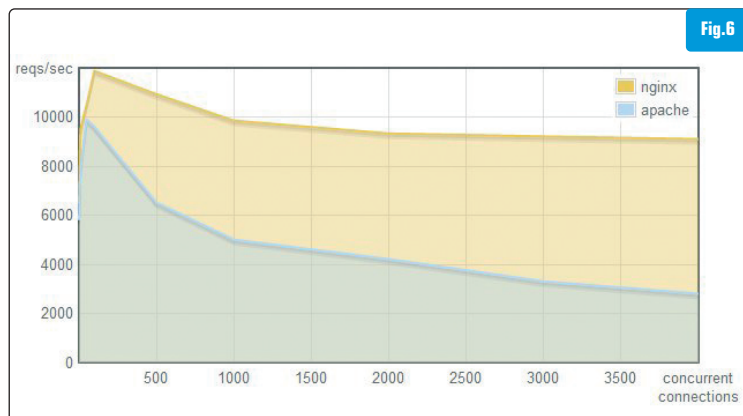
Il est important pour le développeur de connaître les unités physiques de mesure. En effet, ses métriques habituelles de travail sont plutôt les octets et les lignes de code. Pour la mesure de consommation d'énergie, il va falloir mettre en jeu d'autres données. Voici les métriques à prendre en compte :

Ampère : Il s'agit de l'unité de mesure de l'intensité de courant mesurée en charge électrique par seconde. Cette unité n'est pas utilisée pour mesurer la consommation mais plutôt dimensionner les composants.

Watt : Le Watt est l'unité mesurant la puissance, c'est-à-dire le taux d'énergie consommée par l'appareil. La puissance est obtenue en multipliant la tension par l'intensité pour le courant continu. Pour le courant alternatif, la formule est un peu plus complexe car elle fait intervenir le déphasage du courant. La puissance est une mesure instantanée, c'est-à-dire qu'elle donne le taux d'énergie consommé à un instant donné.

Wattheure : Le Wattheure mesure une quantité d'énergie. Un Wh est la quantité d'énergie consommée par un équipement d'une puissance d'un Watt pendant une heure. Un appareil d'une puissance de 100W, consommera 50 Wh d'énergie pendant 30 mn ou 200 Wh pendant 2 heures. Contrairement à la puissance, l'énergie consommée fait toujours intervenir une période de temps définie. On estime par exemple la consommation électrique d'un poste de travail en nombre de kWh par an. La puissance et l'énergie sont souvent confondues. Comparer deux puissances ne permet pas de donner une conclusion sur l'efficacité énergétique car il faut faire intervenir l'utilisation dans le temps.

La mesure des données de consommation nécessite des instruments spécifiques : Les wattmètres. Ils se branchent entre la prise de courant et l'appareil. Ces appareils sont peu coûteux (environ 30 €). Voir [Fig.7]. Les wattmètres ne vont pas donner précisément l'impact d'un logiciel uniquement mais de tout le système. Cependant, ils vont donner des informations importantes. Je vous invite à mettre un wattmètre derrière votre PC, vous aurez des surprises.



Nginx vs Apache



Wattmètre

Ceci n'est rien d'autre qu'un classique profiling. Le profiling est utilisé pour analyser le temps passé par différentes parties du logiciel. On peut citer : Netbean profiler, X Debug... Un logiciel de mesure doit effectuer la même analyse. La problématique pour lui est d'évaluer la consommation à partir du profiling réalisé. Entrent alors en jeu des modèles de consommation. Ces modèles caractérisent la consommation d'énergie en fonction de mesures : charges CPU, nombres d'instructions... On obtient alors la consommation d'énergie d'une tâche du programme. Tel est le fonctionnement des logiciels de mesure existants :

Intel Energy Checker : c'est un SDK qui est fourni par INTEL et qui permet au développeur d'instrumenter son logiciel. Le SDK se connecte à certains instruments de mesure physique et peut intégrer d'autres matériels.

Powertutor : c'est un logiciel développé par l'université du Michigan qui permet de fournir la consommation globale du mobile et des applications pour les téléphones Android. Des fichiers de log peuvent être enregistrés et rapatriés sur PC.

Green Fox : module pour Firefox permettant de mesurer la consommation d'une page ou d'une application web [Fig.8].

> Autres logiciels

Afin d'analyser l'impact des logiciels, vous pourrez utiliser des logiciels classiques de profiling commun. En effet, l'occupation mémoire, la consommation CPU et autres indicateurs classiques sont complémentaires à la consommation d'énergie. Attention cependant à ne pas utiliser que la charge CPU car comme nous l'avons vu, elle n'est pas forcément représentative de la consommation.

CONCLUSION

L'intégration de bonnes pratiques et de green patterns est un nouveau domaine pour les développeurs. De plus dans un contexte où il faut livrer toujours plus vite, l'intégration des green patterns semble contradictoire. Cependant comme nous avons essayé de le montrer, les marges de manœuvres sont énormes. Sans revenir à un développement en assembleur, chaque développeur peut agir pour un développement plus respectueux de l'environnement et des utilisateurs.

Olivier Philippot

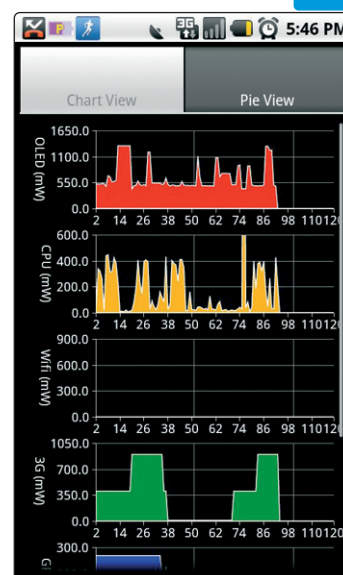
Consultant Green IT et éco-conception des logiciels

Fondateur du Green Code Lab, groupe travaillant sur l'éco-conception greencodelab.fr. Auteur de « Green IT » aux éditions ENI et « Green Patterns ». <http://greencodelab.fr/Livre>

> Logiciel de mesure

La mesure de la consommation d'une partie du logiciel nécessite d'être intrusif. Le logiciel de mesure doit en effet mettre des sondes pour mesurer les ressources utilisées.

Fig.8



Powertutor

Devoteam : une étude 2012 sur le GreenIT

Devoteam a dévoilé fin mars dernier la nouvelle étude autour du GreenIT, « Devoteam Green IT survey 2012 ». Plus de 270 organisations dans 22 pays ont répondu à l'étude.

Plusieurs chiffres clés sont révélateurs de l'informatique verte : 78% des répondants l'envisagent comme un investissement important pour le développement durable (contre 67% l'année dernière), en moyenne, 63% des entreprises ont initié un programme Green IT. Ces projets se font à 82 % dans les grandes entreprises, à peine 44 % dans une PME. Et surtout, 77 % des personnes travaillant dans une PME connaissent le GreenIT et ce que cela signifie réellement. En réalité, on peut opposer deux choses : le Green for IT (réduire l'impact de l'informatique sur l'environnement) et le IT for Green (comment l'informatique contribue au développement durable).

Que faire des déchets électroniques ?

Un problème récurrent se pose aux entreprises (et même aux particuliers) : que faire des déchets d'équipements électroniques et électriques (DEEE) ? L'étude pointe un manque d'information criant et de toute politique : 69 % des entreprises n'ont aucune connaissance des volumes de déchets générés. Pis, 40 % des entreprises ne proposent aucune initiative sur les DEEE et leur gestion.

« Seule l'Allemagne semble sortir du lot : une organisation sur deux a déclaré maîtriser le volume de déchets élec-

troniques et une entreprise sur trois a indiqué en plus suivre leur fin de vie [notion de traçabilité]. A l'opposé, les entreprises marocaines paraissent encore loin de ces préoccupations [19%] », précise l'étude. La durée moyenne de l'utilisation d'un ordinateur est estimée à 3,5 ans et 2,5 ans pour un téléphone portable (un chiffre en baisse régulière avec l'arrivée des smartphones). Pour la France, Devoteam indique : un volume d'équipement électronique mis en service d'environ 24 kg par an par habitant, et un volume collecté de 6,5 kg par an et par habitant.

La tentation de l'achat vert ?

Achète-t-on des produits informatiques et électroniques respectant (plus ou moins) les critères environnementaux ? Grosso modo, 2/3 des entreprises disent regarder ces critères pour les politiques d'achats. Cependant, quand on regarde de près les chiffres, seuls 10 % des entreprises sont très regardantes (pour +20 % des achats), 23 % le sont moyennement (entre 10 et 20 %). Malgré tout, les écolabels sont regardés.

« Les écolabels se divisent en deux catégories pouvant éclairer les responsables informatiques sur l'impact environnemental des produits qu'ils achètent :

- les certifications : les produits ou services sont préalablement testés.*
- les autodéclarations : les organisations enregistrent elles-mêmes leurs produits et services », recadre l'étude.*

Le label Energystar concerne presque 4000 marques et 15 550 équipements. Il s'agit du label le plus répandu.

Le télétravail comme source de GreenIT ?

« L'essor des nouvelles technologies de l'information et de la communication, l'accès à internet à haut débit, la pression de la compétitivité dans les entreprises et la conscience écologique ont fait émerger un nouveau mode de gestion du travail : le télétravail. Il comporte de multiples facettes : travailler à domicile, dans les transports ou dans des espaces collectifs. En travaillant depuis

son domicile 2 jours par semaine en moyenne, un employé pourrait diminuer son empreinte carbone de 10% », précise l'étude. L'Allemagne et le Royaume-Uni sont à la pointe de ce mouvement. La France est largement derrière avec à peine 51 % des personnes travaillant régulièrement depuis leur domicile. Le télétravail ne va pas de soi et pose de multiples problèmes aux entreprises et responsables mais aussi aux salariés qui doivent s'organiser quotidiennement et éviter la confusion entre travail et vie personnelle. D'autre part, il n'est pas toujours simple de régler certains aspects : l'assurance, en cas d'accident, le matériel à fournir, le remboursement de certaines consommations comme l'électricité, etc.

La notion de bilan carbone

On parle souvent de bilan carbone (on l'observe pour tout et un peu n'importe quoi) mais qu'en est-il réellement dans les entreprises ? 30 % des entreprises interrogées disent avoir dressé un bilan carbone. Et là encore, il existe une grande disparité : 30 % utilisent des outils pour suivre et gérer le bilan carbone, 27 % l'ont fait une seule fois et 43 % le font régulièrement...

« ... La responsabilité n'incombe pas uniquement aux Etats. Les organisations peuvent être à la manœuvre. Avec moins de 10% des répondants à utiliser un outil de gestion du carbone, la trajectoire vers une maîtrise des émissions ne se généralise pas. Pourtant, investir aujourd'hui dans la mise sous contrôle de ses émissions, c'est anticiper et mieux se préparer aux enjeux environnementaux et économiques de demain. En outre, l'entrée en vigueur du Grenelle II implique dès la fin de l'année l'obligation d'un bilan carbone pour toutes les entreprises françaises de plus de 500 salariés », souligne l'étude.

Absence des terminaux et du développement

Aussi utile soit cette étude, il manque la dimension purement technique du GreenIT : la programmation, les ressources matérielles. Or, il y a là un vaste chantier à peine effleuré par le GreenIT. Le prochain défi ?



VITE, ABONNEZ-VOUS !

OFFRE
PROLONGÉE
JUSQU'AU 30 JUIN

Jusqu'à -50%



Exceptionnellement, jusqu'en juin, vous pouvez bénéficier, pour 1 euro de plus, d'un des "livres numériques" au choix, sélectionné par la rédaction de Programmez.

(accès html au contenu intégral de l'ouvrage sur la Librairie numérique des Editions ENI, d'une valeur de 21 à 31€ selon les titres)

Pour 1€ de +, Un "livre numérique" des Editions Eni

Html 5 CSS3 (Prix Version numérique : 23,92€)

Google Analytics (Prix version numérique : 21,20€)

Développez une application Android (Prix Version numérique : 31,20€)



Valeur des 11 numéros du magazine : 65,50€ + Valeur du livre : 21 à 31€
= Valeur totale : 86€ à 96,70€

Offre : 50€ seulement (ou 60€ avec accès illimité aux archives).

TOUTES LES OFFRES EN LIGNE : www.programmez.com

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

OUI, je m'abonne

Code,
gestion de projets,
développement web, mobile.

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

à retourner, avec votre règlement à :
Groupe GLI, 17 chemin des Boulangers 78926 Yvelines cedex 9

- ☐ Abonnement 1 an au magazine + 1 « Livre Numérique ENI » : 50 € (au lieu de 65,45 €, prix au numéro + valeur du « livre numérique ») ☐ Abonnement seul : 49 €
- ☐ Abonnement Intégral : 1 an au magazine + Archives Internet et PDF + 1 « Livre Numérique ENI » : 60 € ☐ Abonnement Intégral seul : 59 €
- ☐ Abonnement 2 ans au magazine + 1 « Livre Numérique ENI » : 79 € (au lieu de 130,9 €, prix au numéro + valeur du « livre numérique ») ☐ Abonnement 2 ans seul : 78 €
- ☐ Abonnement intégral 2 ans au magazine + Archives Internet et PDF + 1 « Livre Numérique ENI » : 89 € ☐ Abonnement intégral 2 ans seul : 88 €

Livres à Choisir : ☐ Html 5 CSS 3 ☐ Google Analytics ☐ Développez une application Android

Détails sur www.programmez.com/abonnement.php

Tarifs France métropolitaine

☐ M. ☐ Mme ☐ Mlle Entreprise : _____ Fonction : _____

Prénom : _____ Nom : _____

Adresse : _____

Code postal : _____ Ville : _____

Tél : _____

E-mail : _____ @ _____

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

(Attention, e-mail indispensable pour le livre numérique et les archives sur internet)

Enquête Programmez ! **Emplois et rémunérations informatiques en 2012**

Programmez continue d'enquêter sur les métiers de l'informatique : après notre dossier sur les développeurs aujourd'hui, dans notre précédent numéro, nous vous proposons une nouvelle enquête sur les tendances des emplois et rémunérations de l'informatique en 2012.

avertissement

Les tableaux présentés sur cette enquête sont extraits de la seconde Etude de Rémunération Nationale - Tendances générales du recrutement et des rémunérations en 2012 – du Cabinet Hays et de Cadre Emploi. Les rémunérations sont exprimées en K€.

La stratégie de recrutement des entreprises est depuis plusieurs années marquée par une prudence que d'aucuns jugent excessive. L'environnement économique et la crise de l'euro ne sont pas favorables à l'emploi. La période électorale en ce début d'année ne fait qu'accentuer le phénomène ! La prudence constatée est donc reconduite en 2012, voire accentuée. C'est ce qui ressort de l'étude annuelle nationale menée par le cabinet Hays et Cadre Emploi, qui ne s'affiche pas franchement optimiste. Ainsi, si 83% des entreprises interrogées pour l'étude ont recruté en 2011, elles seront moins de 70% à embaucher de nouveaux collaborateurs en 2012. Qu'en est-il de l'embauche dans le secteur de l'informatique ? De plus en plus présente dans toutes les strates et les produits de l'entreprise, l'informatique s'impose comme un incontournable. Pour autant, et comme tous les autres secteurs, la prudence reste de mise. Certes, le marché est en croissance et en perpétuelle évolution, ce qui permet d'afficher un certain optimisme, mais l'informatique ne recrute certainement pas au même rythme que le marché lui-même. Pourquoi ?

Le secteur informatique affiche depuis de nombreuses années un réel déficit de postulants dans les nouvelles technologies, entraînant une inévitable pénurie de candidats présentant des compétences recherchées. Elle est le fruit du rythme effréné de l'évolution des technologies, qui entraîne un décalage permanent entre

l'offre et la demande, entre la formation et les qualifications recherchées. La résultante également du retrait (retraite) de la génération du papy boom qui vide l'entreprise de ses compétences les plus anciennes (Cobol). Et enfin de la segmentation des secteurs qui recrutent, entre de séduisantes SSII qui parfois jouent la surenchère et qui sont un tremplin vers des postes d'encadrement, et les entreprises consommatrices, dont le message de stabilité et de sécurité a du mal à séduire face aux offres des premières, en particulier pour les nouveaux arrivants sur le marché du travail.

Le secteur des IT optimiste sur l'emploi, mais...

Paradoxalement, malgré une baisse de la confiance générale – 46% seulement des entreprises imaginent des perspectives sectorielles positives dans les prochains mois... elles étaient 61% en 2011 ! - que vient confirmer l'enquête annuelle Eurocom Worldwide, les diri-

geants du secteur des nouvelles technologies demeurent optimistes. 71% d'entre eux prévoient même une hausse de leur chiffre d'affaires au cours des 12 prochains mois. En revanche, ils ne sont que 60% à s'attendre à ce que leur entreprise recrute ! Il est plus rassurant de noter que 5% seulement d'entre eux s'attendent à des licenciements, deux fois moins qu'il y a un an !

« On peut observer deux phénomènes, commente Catherine Bienbeau, directrice de l'agence

Hardcom, membre français du réseau Eurocom Worldwide. D'une part, des technologies prometteuses, notamment dans les domaines du cloud computing et de la mobilité, incitent plus que jamais à des investissements dans de nouveaux outils. D'autre part, les organisations qui veulent maintenir leur compétitivité ne peuvent plus repousser les investissements auxquels elles ont renoncé précédemment. »

LES TENDANCES ET RÉMUNÉRATIONS PAR SECTEUR **Systèmes et Exploitation**

La demande est en hausse sur les systèmes et réseaux. Les moteurs du secteur s'appellent virtualisation, réseaux, sécurité et cloud computing. Autant de domaines stratégiques qui poussent les entreprises, malgré la pénurie, à être attentives au niveau d'exigences qu'elles attendent des candidats. La maîtrise de l'anglais, par exemple, est un incontournable. Parmi les profils les plus recher-

“Il est plus difficile qu'il y a un an d'embaucher, faute de talents disponibles.”

Enquête annuelle Eurocom Worldwide, 2012

Profils de candidats

C'est la grande tendance qui s'affiche aujourd'hui, comme nous l'ont confirmé les experts que nous avons rencontrés pour réaliser notre enquête : le profil personnel du candidat influe de plus en plus fortement dans la prise de décision lors du recrutement. Certes, la formation niveau Bac +5 et le diplôme d'ingénieur sont des clés pour ouvrir les postes de cadres. Et pour les jeunes diplômés, ils sont le sésame vers un accès rapide à un premier emploi. Mais les entreprises recherchent de plus en plus de profils aptes à communiquer, à écouter et à dialoguer avec les métiers et les opérationnels. Des compétences managériales pour fédérer les équipes sont un plus. De même que la pratique de l'anglais courant. A noter que le cumul d'une spécialisation certifiée et de l'anglais courant se valorisent avec un salaire supérieur d'environ 10% à ceux qui sont évoqués dans les tableaux des rémunérations.

chés figurent également les personnalités les plus à même de se rapprocher des métiers et de dialoguer avec les opérationnels.

Ancienneté	0 à 2 ans	2 à 5 ans	5 à 10 ans	+10 ans
Hotlineur	18 / 22	22 / 25	25 / 27	NS
Technicien support	20 / 24	24 / 28	28 / 35	NS
Administrateur systèmes réseaux	26 / 30	30 / 34	35 / 40	40 / 50
Ingénieur systèmes réseaux	30 / 35	35 / 40	40 / 45	45 / 50
Analyste d'exploitation	24 / 28	28 / 32	32 / 36	36 / 40
Superviseur hotline	NS	30 / 35	35 / 40	40 / 45
Responsable informatique	NS	35 / 40	40 / 50	50 / 60
DBA	30 / 35	35 / 45	45 / 60	60 / 70

Source : Cabinet Hays et Cadre Emploi

Etudes et Développements

Après s'être affiché en nette hausse début 2011, avec en particulier des salaires en progression, le secteur des études et du développement a marqué le pas en fin d'année, une tendance qui se prolonge en 2012. Là aussi la personnalité du postulant joue énormément dans le processus de recrutement. Malgré la réserve des entreprises, qui se traduit par un allongement du processus de recrutement, le secteur va continuer de recruter. Figurent, parmi les profils les plus recherchés, les compétences MOE et experts Java, les architectes et MOA. Les candidats de niveau Bac +5 qui présentent ces profils sont assurés de bénéficier d'une évolution rapide en interne.

Ancienneté	0 à 2 ans	2 à 5 ans	5 à 10 ans	+10 ans
Analyste-programmeur	25 / 30	30 / 35	35 / 40	40 / 45
Ingénieur développeur	30 / 38	38 / 43	43 / 55	55 / 65
Ingénieur qualité / méthodes	30 / 33	33 / 38	38 / 42	42 / 45
Ingénieur intégration	32 / 38	38 / 43	45 / 55	NS
Responsable R&D	NS	NS	50 / 55	55 / 65
Chef de projet études et développement	32 / 38	38 / 45	45 / 55	55 / 60
Architecte	NS	38 / 45	45 / 60	60 / 70
Directeur technique	NS	NS	65 / 75	80 / 100

Source : Cabinet Hays et Cadre Emploi

Jeunes diplômés : l'obstacle du premier emploi

La première embauche reste une étape délicate pour les jeunes diplômés. Et comme le constate le cabinet Hays, les années qui passent ne changent rien, voire accentuent la difficulté ! « En 2011, plus d'un tiers de nos clients ne recrutait aucun jeune diplômé ; de même, un autre tiers environ recrutait 1 à 10% de jeunes diplômés par rapport à leurs embauches totales. En 2012, ils seront plus de 40% à ne recruter aucun nouvel entrant sur le marché de l'emploi et 35% à recruter entre 1 et 10% de jeunes diplômés. »

Le phénomène est moins prégnant dans les nouvelles technologies lorsque les candidats sortent des écoles spécialisées. Diplôme d'ingénieur Bac +5 en poche, le jeune diplômé a généralement été approché durant sa formation, et pour plus de 90% d'entre eux, un poste les attend en sortie d'école, généralement dans une SSII. Et une riche carrière, à condition de ne pas chercher dans la foulée à atteindre systématiquement un poste de chef de projet...

Les chiffres le prouvent, une spécialisation technologique judicieuse est plus recherchée et ouvre des perspectives de poste et de rémunération certainement plus riches, à condition de saisir les opportunités qui se présentent.

Conseil et MOA

Les entreprises ne peuvent continuer longtemps de reporter les projets. Le secteur du conseil et de la MOA devrait en profiter et le recrutement repartir à la hausse.

Hormis, comme le souligne avec inquiétude le cabinet Hays, dans la banque qui reste tendue ! Démonstration du retour à l'optimisme avec le jeu de la concurrence que recommencent à jouer les candidats, en particulier sur les profils les plus rares. Un phénomène en revanche qui pourrait accentuer la pénurie de candidats qualifiés. Le

Recrutement, sites d'emplois et réseaux sociaux

Internet est désormais un incontournable de la recherche d'emploi. Parmi la profusion des sites qui lui sont dédiés, Pôle emploi demeure la plateforme en ligne préférée des Français. 73% des internautes ont visité pole-emploi.fr pour consulter une offre, 57% plusieurs fois. Suivent Monster, Keljob, etc. Un sondage CSA a permis de mesurer l'efficacité ressentie de ces sites dans la recherche d'un emploi sur internet. Pole-emploi.fr est cité en premier à hauteur de 23%. Viennent ensuite les autres sites emploi, les espaces recrutement des sites entreprises, les cabinets de recrutement, puis les réseaux sociaux, blogs et Twitter. Le sondage relativise également le rôle des réseaux sociaux – Vimeo et LinkedIn en particulier - dans la recherche d'emploi, sur lesquels seuls 10% des internautes déclarent

posséder un profil. Leur rôle n'est cependant pas à négliger, tous les cabinets de recrutement passent par eux dans leur chasse à la perle rare. C'est d'ailleurs généralement la mission qui leur est soumise, trouver 'le' profil technique qui répond à une attente précise de l'entreprise ou de la SSII. Pour cela, les consultants exercent une veille proactive sur les réseaux sociaux, et vont jusqu'à consolider un fichier de candidats potentiels.

A ne pas négliger, surtout pour les métiers de l'IT... Qu'en est-il du côté des dirigeants des entreprises qui recrutent ? Et bien une étude Eurocom Worldwide vient de révéler qu'environ 40% des entreprises consultent le profil des postulants sur différents médias sociaux. Mais surtout qu'on a rejeté la candidature d'un cadre sur cinq, suite à une visite

de son profil sur un média social. La déconvenue du candidat qui doit affronter son image (et ses erreurs...) véhiculée par les médias sociaux est bien une réalité ! « Dans les années à venir, beaucoup d'entre nous seront confrontés à ce qu'ils rendent public aujourd'hui sur différentes plateformes sociales », constate Mads Christensen, directeur du réseau Eurocom Worldwide. « Le fait qu'un recruteur sur cinq écarte des postulants sur la base de ce qu'il découvre sur internet est un indicateur très fort de l'impact de ces nouveaux médias dans la vie de tous les jours. »

L'étude Eurocom Worldwide, principalement menée auprès d'entreprises européennes, a également confirmé que le media social le plus populaire dans le secteur IT reste LinkedIn (74%). Viennent ensuite Twitter (67%), Facebook (65%), puis YouTube (56%). Vimeo est un épiphénomène franco-français.

relationnel, l'écoute, la compétence rédactionnelle et l'aptitude à fédérer les équipes sont attendus des candidats qui postulent pour ces postes et seront appréciés.

Ancienneté	0 à 2 ans	2 à 5 ans	5 à 10 ans	+10 ans
Consultant	33 / 38	38 / 43	43 / 50	50 / 60
Chef de projet MOA	NS	40 / 50	50 / 60	60 / 70
Directeur de projet	NS	NS	55 / 70	70 / 85
Consultant en stratégie / organisation	35 / 40	40 / 50	50 / 60	60 / 80
Directeur informatique	NS	NS	60 / 80	80 / 100
Directeur de systèmes d'information	NS	NS	70 / 90	> 90

Source : Cabinet Hays et Cadre Emploi

ERP et BI

Le secteur de l'ERP (gestion) et de la BI (Business Intelligence et analytique) est très proche dans ses tendances en matière de recrutement du précédent, Conseil et MOA. Une proximité logique, les projets de l'un concernent l'autre... Le cabinet Hays souligne la vigilance de ses clients sur les compétences techniques et fonctionnelles des candidats. Une formation en école d'ingénieurs généraliste, complétée d'une option finance, vente ou comptabilité, figurent parmi les profils les plus appréciés. Les candidats seniors affichant des compétences techniques et une spécialisation sur un métier sont également très demandés, en raison de leur connaissance du marché de la MOE.

Ancienneté	0 à 2 ans	2 à 5 ans	5 à 10 ans	+10 ans
Développeur ABAP	28 / 33	33 / 38	38 / 43	43 / 50
Consultant ERP	33 / 38	38 / 45	45 / 55	55 / 65
Consultant fonctionnel	NS	38 / 45	45 / 55	55 / 70
Chef de projet ERP	NS	35 / 42	45 / 55	55 / 70
Directeur de projet	NS	NS	50 / 65	65 / 85

Source : Cabinet Hays et Cadre Emploi

Femme et ingénieur : la longue marche vers l'équilibre

Femme et ingénieur... Longtemps ces deux expressions n'ont pu s'associer, à l'exception notable de quelques pionnières qui défilerent certains 14 juillet. Les formations scientifiques et d'ingénieurs ne séduisaient pas le public féminin, nous affirmait-on ! Aujourd'hui, les promotions des grandes écoles se sont féminisées. « En R&D, biotech, chimie, laboratoire pharmaceutique et télécommunications, la population de cadres issus de formations techniques type Bac +5 est féminine à 60% », affirme Laurent Hürstel, Directeur associé de Robert Walters. Les femmes profiteraient de la consumérisation, asexuée, mais également « de l'image studieuse associée aux femmes : écoute, conciliation, retenue et rigueur ».

Chez les ingénieurs, la tendance serait donc la parité, voire de dépasser l'équilibre. « Cette tendance est largement confirmée. Nos clients sont de plus en plus sensibles à la parité. Preuve en est : plus de la moitié des candidats approchés sur les métiers en R&D, Bureaux d'études et fonctions 'support' (Supply Chain, Procurement, QHSE) sont des femmes avec 5 à 15 ans d'expérience ». Quant aux salaires, ils tendent globalement au nivellement. « Nos clients veulent d'abord et avant tout des 'excellents' candidats et sont prêts à payer le prix pour s'attacher leur service. Et ce, au-delà du sexe, de l'environnement social ou de l'origine ».

Les seniors sont toujours sur la brèche

C'est une tendance inattendue, la demande de profils seniors s'affiche en croissance. En fait, les entreprises doivent faire face à deux phénomènes majeurs qui pèsent sur leur stratégie RH et militent pour l'emploi des seniors : le vieillissement de leur informatique et l'efficacité des personnes en place.

Le premier phénomène place l'entreprise dans une position délicate, car une partie de ses équipements tourne sur des environnements de programmation et/ou de pilotage qui ont depuis longtemps disparu des cursus d'enseignement et qui ne sont plus opérés que par une génération d'hommes et de femmes qui s'approchent à grands pas de la retraite. La programmation en langage Cobol sur le Mainframe, toujours aussi vivace, en est l'exemple emblématique.

Une réponse moderne invite ces entreprises à s'équiper avec du neuf, qu'il s'agisse du matériel ou des hommes. Mais c'est oublier que ce phénomène va plus loin que le franchissement d'un seuil technologique... Pourquoi l'entreprise changerait-elle son parc alors que le précédent n'est parfois pas encore amorti, ou que lorsqu'il l'est, il n'est plus une source de coût ? Voire plus prosaïquement, les équipements en place satisfont l'entreprise et ses clients, alors pourquoi changer ? Et bien plutôt que de jouer la carte du changement, l'entreprise mise sur les seniors qui sauront maintenir le statu quo.

Le second phénomène rejoint certainement le premier. Les entreprises ont depuis longtemps mesuré la difficulté qu'elles éprouvent à rentabiliser rapidement un projet, et le poids que peuvent peser les juniors dans cette épreuve. Qu'ils soient recrutés par les entreprises ou fournis par les SSII, les juniors coûtent chers pour une efficacité... souvent réduite !

L'entreprise a-t-elle le choix ? Et bien oui, et lorsqu'il se présente, elle préférera employer un senior, qu'elle rémunérera certainement plus cher, mais qui disposera des compétences et de l'expérience accumulées pour aller plus vite sur des projets mieux maîtrisés, qui réduira le risque de dérapage, et qui au final assurera un « time to market » plus rapide, donc plus rentable. « Les entreprises sont prêtes à dépenser davantage pour s'offrir des profils expérimentés dont elles ont besoin pour mener à bien les grands projets », remarque Katharina Dalka, consultante chez Pierre Audoin Consultants (PAC).

« Les acheteurs ainsi que les donneurs d'ordre sont désormais conscients que faire uniquement appel à des profils juniors équivaut souvent à un prolongement du projet et revient in fine plus cher. Dans cette même dynamique, les profils débutants perdent en grande partie de leur valeur. »

Voilà pourquoi la demande de profils seniors s'affiche en croissance et pourquoi ces derniers justifient un salaire plus élevé, comme l'indique l'étude sur les salaires de l'informatique.

Et puis, des perles peuvent émerger, des seniors qui affichent des compétences recherchées, comme la maîtrise de langages « disparus ». Dans ce cas, leur rémunération se situera hors cadre, et pourra parfois dépasser les limites du raisonnable !

Yves Grandmontagne

Ces entreprises qui recrutent

2000 postes à pourvoir

Programmez ! a de nouveau repéré pour ses lecteurs près de 2 000 nouveaux postes de Développeurs, Ingénieurs, Chefs de projets, Consultants, etc., à pourvoir sur toute la France en 2012.

> Ingénieurs

Sogeti High Tech propose plus de 700 postes d'ingénieur dans ses cinq métiers : le conseil, l'ingénierie système, l'ingénierie physique, l'ingénierie logiciel et le testing.

La SSI recrute des ingénieurs dans les domaines d'activité suivants : conception, étude et développement, calcul, analyse/simulation, tests logiciels, essais systèmes ou structuraux, validation, intégration, industrialisation, gestion de projet, sécurité.

www.sogeti-hightech.fr

> Techniciens et ingénieurs

La SSII européenne **Computacenter** continue d'embaucher (150 embauches ont déjà été réalisées depuis le début 2012) avec 90 nouveaux postes à pourvoir immédiatement dans ses 3 domaines d'activité : l'infogérance de transformation et les services de support, le conseil et l'évolution technologique des infrastructures, les services de distribution. Les postes à pourvoir : Technicien informatique débutant, Technicien informatique expérimenté, Service delivery manager, Service manager, Ingénieur informatique, Architecte, Chef de projet, Consultant.

www.computacenter.fr

> Techniciens Service Desk

La SSII d'infrastructure **SCC** annonce un plan de recrutement de près de 200 collaborateurs d'ici fin 2013, tous métiers confondus, dont 70 postes de techniciens Service Desk. SCC recherche des candidats au profil technique, acquis soit dans le cadre de leur cursus académique, soit au travers d'expériences professionnelles. Elle s'engage également à offrir une formation aux environnements techniques et informatiques de la société et de ses clients.

www.scc.com

Consultants

EquiTime, éditeur de solutions progiciels pour les plannings des collaborateurs sur le secteur Hospitalier, ainsi qu'après de grands comptes industriels et de collectivités, recrute plus de 10 consultants fonctionnels sur Paris et Région Parisienne. Profil Bac +2/4 en informatique/gestion RH, avec une expérience significative sur la mise en place opérationnelle d'un progiciel de gestion des temps.

www.equitime.fr

> Ingénieurs

Hexaglobe, spécialiste des technologies et services pour les nouveaux médias et diffuseurs, cherche à recruter 5 nouvelles personnes pour atteindre un effectif de 20 salariés d'ici fin 2012. La société recherche 2 chargés de projet dont 1 technico-commercial, 2 ingénieurs et 1 ingénieur commercial. www.hexaglobe.com

> Consultants gestion des risques IT

Provadys, cabinet français agréé certificateur auprès du PCI SSC et de l'Arjel, recrute au cours du 1er semestre 2012 plus d'une vingtaine de professionnels de la gestion des risques IT. Pour travailler sur des projets liés aux problématiques de conformité.

www.provadys.com

> Accounts Manager, chargé de clientèle, etc.

Le groupe **CCA International**, spécialiste de la relation client à distance, crée 150 emplois en Poitou-Charentes, à Poitiers sur la zone d'activité du Futuroscope, avant la fin mai, pour ses activités de vente dans le secteur des médias et des télécoms.

www.ccainternational.com

> Ingénieurs

La SSII **OPEN** confirme sa prévision de recrutement de plus de 700 nouveaux collaborateurs en 2012 afin de soutenir ses ambitions de croissance organique, sur tout le territoire national étendu à la Belgique et au Luxembourg. OPEN recrute en Ingénierie Applicative (ingénieur d'études, ingénieurs concepteurs Java, .Net, Open Source, C++), sur les Infrastructures (Ingénieurs Systèmes Unix, Linux, Supports Applicatifs, Ingénieur en Sécurité, chef de projet technique), et l'accompagnement des métiers clients (experts métiers, consultants AMOA et architectes).

Dans le domaine du Conseil, OPEN intégrera des profils de haut niveau en Conseil en Sécurité et Banking, ainsi que de jeunes consultants issus de grandes écoles d'ingénieurs et de Management. Plus de 15% des recrutements porteront sur des candidats débutants ou juniors (- de 2 ans d'expérience) avec un niveau de formation majoritairement Bac+5 (Ecole d'ingénieurs ou de management).

Les postes à pourvoir sont notamment des Ingénieurs Etudes .Net, Java, C++, Multimédias, Ingénieurs Systèmes & Réseaux : Unix, Windows, Cisco, Ingénieurs Informatique Embarquée, Temps Réel, Consultants Métiers, Architectes/Urbanistes, Administrateurs Bases de Données et Systèmes, Chefs de projet, Directeurs de projets, Consultants en Organisation, Conduite du changement, Transformation du SI, Ingénieurs commerciaux SSII tous secteurs économiques.

www.open-groupe.com

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



Introduction à la programmation GPU

Rechercher plus de performances

4^e partie



Dans les articles précédents, nous avons présenté les fondamentaux de la programmation parallèle avec **CUDA C**. Dans cette dernière partie consacrée à la technologie **CUDA C**, nous allons nous intéresser aux performances en exploitant les caches des microprocesseurs installés sur les cartes graphiques. Contrairement à la programmation CPU, les caches GPU sont entièrement gérés par le développeur, il est donc très important de comprendre comment les exploiter, car comme vous pourrez le constater le code est un peu plus compliqué à écrire, mais c'est le prix à payer pour obtenir de meilleures performances.

> Retour sur l'organisation de la mémoire

Précédemment, nous avons esquissé l'organisation de la mémoire en acceptant quelques simplifications didactiques. Notre première implémentation mérite un peu plus d'explications sur l'organisation mémoire de l'architecture CUDA. Pour faciliter notre compréhension, nous avons représenté sur le schéma [Fig.1] une grille contenant deux blocs de threads (0, 0) et (1, 0).

Au bas du schéma de la [Fig.1], deux morceaux de mémoire sont représentés : la mémoire globale (Global Memory) et la mémoire constante (Constant Memory) qui n'apparaissaient pas sur le schéma du précédent article. On remarque que la partie host (code CPU) communique directement via l'interface matérielle PCI Express avec ces deux types de mémoire. La mémoire constante est visible par tous les threads de la grille, mais elle est à lecture seule, offrant de ce fait des performances meilleures que la mémoire globale. La mémoire constante permet de stocker les variables « constantes » pour toute la durée d'exécution de l'application. Les variables constantes sont souvent représentées par les valeurs fournies en entrée des méthodes « noyau ». Les variables constantes sont stockées dans la mémoire globale, mais aussi cachées dans la mémoire constante pour offrir un accès plus rapide. Avec ce mécanisme de mise en cache, l'accès à la mémoire constante peut se révéler très intéressant sur le plan des performances lorsque plusieurs lectures se déclenchent en parallèle. La mémoire globale est visible par tous les threads de la grille en lecture et en écriture, mais elle est relativement lente. Son principal avantage est sa grande taille, dans la majorité des cas, vous ne vous poserez pas de question sur la taille des données que vous souhaitez copier. Les échanges entre l'host et la mémoire globale sont simples d'accès via les méthodes réalisant des copies entre l'host et la mémoire globale côté GPU. Pour les threads, l'accès à la mémoire est naturel (peut-être trop) et ne diffère pas du code traditionnel sur CPU.

En haut du schéma, deux blocs de threads sont représentés. Ces blocs possèdent chacun deux threads CUDA avec leurs registres respectifs, ainsi qu'une petite ligne de cache, appelée Shared Memory en dialecte CUDA. Cette mémoire partagée pour tous les threads du bloc courant est extrêmement rapide (de l'ordre de 10 fois plus rapide que la mémoire globale), mais relativement petite (de l'ordre de quelques kilooctets). Les accès à la mémoire partagée sont explicites et réclament de la part du développeur un développement

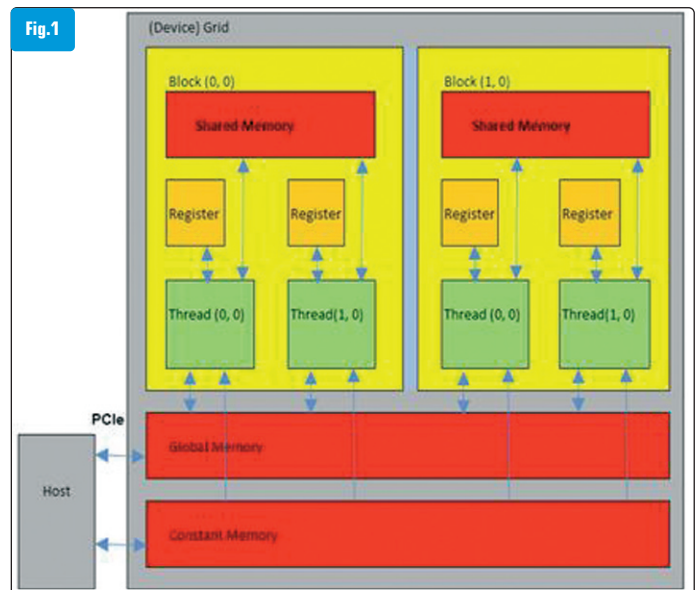


Figure 1 – Organisation de la mémoire vis-à-vis des threads GPU

spécifique. Enfin, notez que cette mémoire est partagée par tous les threads d'un bloc, permettant aux threads de collaborer pour accéder en lecture et en écriture aux données stockées en mémoire partagée avec d'excellentes performances. Cependant, nous pouvons déjà imaginer que les threads d'un même bloc devront se synchroniser pour ne pas aboutir à des traitements incohérents. Enfin, vous pouvez remarquer que les autres threads de l'autre bloc ne peuvent pas accéder à la mémoire partagée d'un autre bloc. La taille de la mémoire partagée étant très faible, le développeur devra généralement procéder à un découpage des données pour profiter des performances offertes par ce type de mémoire.

> Retour sur notre algorithme de multiplication matricielle

Si nous analysons le code de la multiplication matricielle, nous pouvons observer un défaut majeur. A votre avis, dans le code ci-dessous (Fragment de code 1) y-a-t-il des recouvrements de valeurs dans le calcul matriciel ? En d'autres termes, les indices des matrices A et B sont-ils réutilisés plusieurs fois au cours du calcul ?


```
__global__
void MatrixMultiply(float* A, float* B, float* C, int size)
{
    int i = blockIdx.y * blockDim.y + threadIdx.y;
    int j = blockIdx.x * blockDim.x + threadIdx.x;

    if (j < size && i < size)
    {
        float sum = 0;
        for (int k = 0; k < size; ++k) {
            sum += A[i * size + k] * B[k * size + j];
        }
        C[i * size + j] = sum;
    }
}
```

Fragment de code 1 : code du calcul matriciel simple

Pour simplifier, prenons le cas de quatre threads traitant en parallèle le calcul du produit matriciel.

	C(0,0) Thread(0,0)	C(1,0) Thread(1,0)	C(0,1) Thread(0,1)	C(1,1) Thread(1,1)
Ordre d'accès ↓	A(0,0)*B(0,0)	A(0,0)*B(1,0)	A(0,1)*B(0,0)	A(0,1)*B(1,0)
	A(1,0)*B(0,1)	A(1,0)*B(1,1)	A(1,1)*B(0,1)	A(1,1)*B(1,1)
	A(2,0)*B(0,2)	A(2,0)*B(1,2)	A(2,1)*B(0,2)	A(2,1)*B(1,2)
	A(3,0)*B(0,3)	A(3,0)*B(1,3)	A(3,1)*B(0,3)	A(3,1)*B(1,3)

Figure 2 : analyse de calcul matriciel vis-à-vis de l'accès mémoire

Sur une ligne horizontale sur la [Fig.2], que remarquez-vous ?

	C(0,0) Thread(0,0)	C(1,0) Thread(1,0)	C(0,1) Thread(0,1)	C(1,1) Thread(1,1)
Ordre d'accès ↓	A(0,0)*B(0,0)	A(0,0)*B(1,0)	A(0,1)*B(0,0)	A(0,1)*B(1,0)
	A(1,0)*B(0,1)	A(1,0)*B(1,1)	A(1,1)*B(0,1)	A(1,1)*B(1,1)
	A(2,0)*B(0,2)	A(2,0)*B(1,2)	A(2,1)*B(0,2)	A(2,1)*B(1,2)
	A(3,0)*B(0,3)	A(3,0)*B(1,3)	A(3,1)*B(0,3)	A(3,1)*B(1,3)

Figure 3 : analyse de calcul matriciel montre un chargement multiple des valeurs

On remarque que chacune des valeurs est systématiquement chargée deux fois ! Les threads chargent plusieurs fois la même valeur depuis la mémoire globale, ce qui provoque une surcharge substantielle en termes de trafic de la mémoire globale, ce qui ralentit forcément notre traitement. Comment éviter ce problème ? Naturellement, l'idée serait d'utiliser la mémoire partagée qui est bien plus rapide, mais malheureusement très petite.

Imaginons que toutes les données soient déjà présentes en mémoire partagée, les quatre threads pourraient charger plusieurs fois les mêmes valeurs sans problème, car le coût d'accès à la mémoire partagée est très rapide (10 fois moins coûteux).

Si nous résumons nos idées :

- La mémoire partagée est réduite en taille, il va donc falloir fractionner nos données en petit rectangles de données adaptées à la capacité de la mémoire partagée.
- Pour chaque rectangle, nous devons recopier ses données dans la mémoire partagée.
- Une fois que toutes les données du rectangle sont recopiées, nous pouvons traiter notre calcul réparti sur tous les threads GPU correspondant à notre rectangle qui puiseront leurs données depuis la mémoire partagée.
- Les calculs intermédiaires d'un rectangle doivent être placés dans une variable temporaire. Au final, la variable temporaire est recopiée dans la cellule correspondante de la matrice C.

Il est usuel en programmation GPU de comparer ce découpage en rectangles au recouvrement d'une surface avec des tuiles. L'élément clef de cette répartition est de permettre aux calculs de s'exécuter sans surcoût.

Nous pouvons remarquer que l'usage de la mémoire partagée est conditionné par l'algorithme. Ici le produit matriciel utilise plusieurs fois les mêmes valeurs au cours du traitement c'est donc un excellent candidat pour utiliser la mémoire partagée.

La technique de découpage en tuiles n'est pas toujours la bonne solution

Il n'est pas toujours possible d'utiliser la technique des tuiles pour exploiter la mémoire partagée. En effet, si votre algorithme ne comporte pas d'utilisation multiple des mêmes données, il n'y a pas de gain de performance possible dans cette direction.

Par exemple dans le cadre d'une simple addition de vecteurs, cela n'aurait aucun sens, car les données ne sont jamais utilisées plusieurs fois durant le calcul. Mais dans le cas d'addition de deux vecteurs, une autre technologie basée sur les techniques vectorisation comme SSE (Streaming SIMD Extensions) ou dernièrement AVX (Advanced Vector Extensions), serait sans doute la meilleure solution sur le plan des performances.

Lorsque vous décidez de paralléliser un morceau de code, vous devez rester vigilant sur l'algorithme afin de déterminer comment le paralléliser efficacement.

> Révision de l'algorithme du code noyau

Après avoir découvert l'organisation de la mémoire, vous vous demandez peut-être comment exprimer qu'une variable soit définie dans la mémoire partagée. Le compilateur **CUDA C** expose un mot clef `__shared__` permettant de définir de la mémoire partagée sur des variables. Dans notre réflexion, nous considérons que les traitements sont marqués en étapes : la copie des données dans la mémoire partagée, puis le calcul matriciel et enfin la copie de la valeur dans la matrice résultat. Pour s'assurer que toutes les étapes sont bien orchestrées, nous utiliserons une nouvelle fonction **CUDA C**, `__syncthreads()`. La fonction `__syncthread()` permet d'attendre que tous les threads au sein d'une tuile aient terminé leurs accès mémoire.

Comment définir la taille de vos tuiles ?

Nous n'avons pas expliqué comment définir la taille des tuiles (TILE_WIDTH dans le code). La réponse est simple, vous devez expérimenter vous-même les performances de votre programme en fonction des tailles utilisées. En général pour le produit matriciel et vis-à-vis de la génération des cartes actuelles, une taille de tuile, 16 est souvent optimum. Mais ce chiffre pourra évoluer en fonction des matériels.

Performances de calcul matriciel en fonction de la taille des tuiles sur une carte NVIDIA Quadro 2000M. La ligne « Performance » est exprimée en millisecondes.

Taille	2	4	8	16
Performance	3 464	523	92	52

On remarque que la taille à une conséquence importante vis-à-vis du programme.

Je vous encourage à tester les performances de vos traitements reposant sur la technique des tuiles afin d'évaluer la taille adéquate des tuiles vis-à-vis de votre programme.

Le code ci-dessous (Fragment de code 2) illustre nos idées précédentes.

```
__global__
void MatrixMultiplySquareCachedTiled(float* A, float* B, float
* C, int size)
{
    __shared__ float As[TILE_WIDTH][TILE_WIDTH];
    __shared__ float Bs[TILE_WIDTH][TILE_WIDTH];

    int bx = blockIdx.x;
    int by = blockIdx.y;
    int tx = threadIdx.x;
    int ty = threadIdx.y;

    int i = by * TILE_WIDTH + ty;
    int j = bx * TILE_WIDTH + tx;

    float sum = 0;

    for (int m = 0; m < size/TILE_WIDTH; ++m)
    {
        // Phase 1
        As[ty][tx] = A[i * size + (m * TILE_WIDTH + tx)];
        Bs[ty][tx] = B[(m * TILE_WIDTH + ty) * size + j];
        __syncthreads();

        // Phase 2
        for (int k = 0; k < TILE_WIDTH; ++k) {
            sum += As[ty][k] * Bs[k][tx];
        }
        __syncthreads();

        // Phase 3
        C[i * size + j] = sum;
    }
}
```

Fragment de code 2 : code du calcul matriciel tuilé

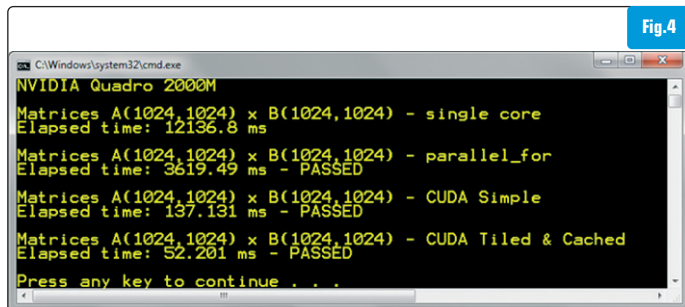


Figure 4 - Exécution de tous nos calculs matriciels exprimés en millisecondes

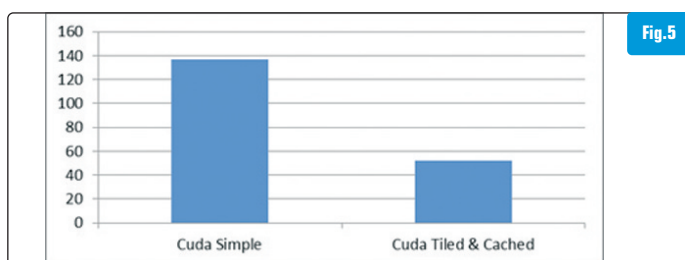


Figure 5 - Comparatif CUDA avec et sans cache exprimé en millisecondes

On retrouve le fractionnement des données et les trois phases dans le traitement d'une tuile :

- Le chargement des données depuis la mémoire globale vers la mémoire partagée.
- Calcul matriciel pour la tuile courante.
- Recopie de la variable temporaire dans la cellule correspondante à matrice C.

> Exécution du produit matriciel en séquentiel, en parallèle, en CUDA simple et en CUDA tuilé

En programmation parallèle, la performance est sans aucun doute un des buts premiers. Une comparaison avec toutes formes de calculs matriciels étudiées dans cette série d'articles permet de nous montrer les gains obtenus pour chaque mode d'exécution [Fig.4].

Pour éviter d'avoir un effet réducteur, nous ne comparons ici que les implémentations **CUDA** [Fig.5].

Cette nouvelle implémentation nous apporte un gain de 38% de performance au prix d'une complexité substantielle, mais qui ne devrait pas rebuter le passionné de performance qui caractérise les programmeurs GPU en général.

> En conclusion

La programmation **CUDA C** exige une bonne compréhension de l'architecture **CUDA**. En effet, les capacités de votre carte graphique seront toujours à prendre en compte vis-à-vis de votre appétit à accélérer vos calculs. La maîtrise du nombre de blocs et du nombre de threads, des indices et du nombre de tuiles de votre algorithme en mode noyau constitue un élément essentiel pour réussir d'excellents programmes **CUDA**. Ces calculs peuvent sembler compliqués pour le néophyte en programmation GPU, mais une pratique régulière de ces calculs vous permettra de ne plus être pénalisés par cette petite gymnastique intellectuelle. Sur le plan de la programmation, le Framework **CUDA C** impose une bonne organisation du code afin de ne pas sombrer dans des mélanges douteux entre le code **CPU** et le code **GPU**. Si votre algorithme s'y prête, l'utilisation de la mémoire cache permet d'obtenir des performances extrêmes, mais au prix d'un code un peu plus complexe. Il est sans doute déconseillé de se lancer dans la programmation **CUDA C** trop rapidement. Ce type de programmation reste un peu rugueux, mais les performances sont littéralement exceptionnelles lorsque vos algorithmes sont orientés données (traitements orientés boucles).

Ressources

Si vous souhaitez aller plus loin sur le développement **CUDA** je vous conseille les ouvrages ci-contre. Le premier est sans doute le plus didactique pour une première approche. Le second est beaucoup plus complet et offre des explications très détaillées, c'est donc un ouvrage de référence.

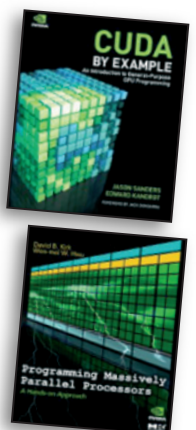


Bruno Boucard

Expert indépendant en parallélisme.

MVP Technical Computing

<http://blogs.msdn.com/devpara/>



Prenez une longueur d'avance : créez une application Windows 8 avec Visual Studio & TFS 11

Partie 2 : Agilité avec TFS11

Depuis la disponibilité de la version Bêta de Visual Studio 11, Team Foundation Server 11 et Windows 8 en Consumer Preview, certains éléments du premier article ont changé. Nous allons revenir dessus et aller un peu plus loin.

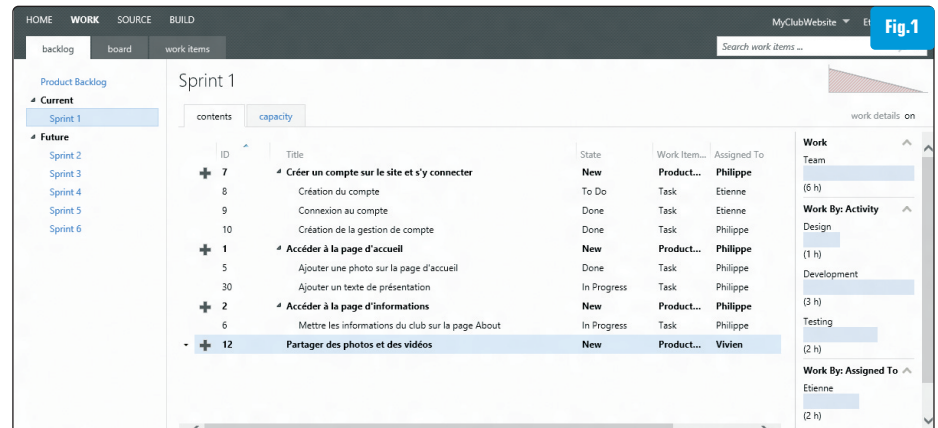
L'outil de StoryBoarding souffrait de l'absence de formes spécifiques Metro pour Windows 8. Ce manque est maintenant comblé depuis la sortie de la bêta qui propose tout un ensemble de contrôles et icônes permettant de designer des applications Metro pour Windows 8. La Grid créée dans le précédent article n'est cependant pas inutile car il ne s'agit pas d'un élément proposé parmi les Shapes de base.

Autre nouveauté et pas des moindres car elle transfigure tous les outils, depuis Visual Studio en passant par Team Explorer jusqu'au Team Web Access : le nouveau thème de l'interface, qui dans une volonté de se rapprocher de Metro, est devenu très monochrome. Ce tournant radical a été pris dans un souci d'épurer les écrans et d'améliorer les performances, notamment par l'utilisation d'une police de caractères dédiée aux icônes de l'application. Ce concept, également présent dans Windows 8 permet de limiter le chargement d'images et libère des ressources en ne chargeant qu'une seule fois la police de symboles. Bien d'autres modifications sont apparues dans cette version Consumer Preview, mais celles-ci touchent des outils qui n'ont pas été présentés lors du précédent article et qui le seront dans celui-ci ainsi que dans les suivants.

> Les méthodologies de développement

Lors du développement d'une application, l'équipe va généralement suivre des processus et une méthodologie particulière. Ces méthodologies sont nombreuses, et avec la nouvelle version de Team Foundation Server, c'est notamment les méthodologies dites Agiles qui sont mises en avant.

Dans Team Foundation Server, une méthodologie se matérialise sous la forme de ce que l'on appelle un Process Template. Ce dernier définit la configuration de tous les outils utilisés tout au long d'un développement d'un projet d'équipe, tels que la sécurité au niveau du contrôleur de code source,



la configuration de la gestion d'éléments de travail (Work Items), les documents disponibles sur le portail d'équipe SharePoint, etc. Ces Process Template contiennent des informations concernant la gestion du travail dans un projet d'équipe, matérialisées par des définitions de type Work Items. Dans le cas des Process Template Agile et Scrum, on va trouver des éléments avec un nom familier pour des utilisateurs de ces méthodologies tels que les User Story, les Tasks, les Bugs, les Issues, les Test Cases, etc. C'est au travers de ces Work items que l'on va coordonner les différentes interactions avec le projet d'équipe.

> Les méthodologies agiles de développement

Les méthodologies de développement dites agiles sont un ensemble de méthodes de gestion de projets visant à améliorer la qualité logicielle, et à améliorer la réactivité des équipes de développement avec notamment une plus grande implication du client.

En 2001 l'Agile Manifest formalise les principes des méthodes agiles en définissant 4 valeurs et 12 principes communs à celles-ci. Un des principes fédérateurs de ces méthodologies est l'organisation des cycles de développement en itérations incrémentales avec un découpage fonctionnel impliquant le client. Le travail d'équipe, la responsabilisation des membres de l'équipe

et l'auto-organisation du travail de l'équipe sont également des piliers des méthodes agiles. La version 11 de Visual Studio et Team Foundation Server offre de nouveaux outils très clairement orientés vers ces types de méthodologies et notamment vers la méthodologie Scrum.

> La méthodologie Scrum

La version 2010 de Team Foundation Server fournit dès l'installation deux Process Template : MSF for Agile Software Development v5.0 et MSF for CMMI Process Improvement v5.0. De plus, un nouveau Process Template, appelé Visual Studio Scrum 1.0, est disponible en téléchargement sur le site internet de Visual Studio Gallery et vient compléter le panel de Process Template disponibles, en proposant cette fois-ci une configuration beaucoup plus focalisée sur la méthodologie Scrum. Ce Process Template est intégré en standard, en version 2.0, dans la version 11 de Team Foundation Server. Dans ce Process Template, on obtient des définitions de Work Items épurées, ne demandant aux participants du projet que le strict minimum pour l'application de la méthodologie Scrum (exemple : Ici pas de champs de saisie du **temps passé** à compléter une tâche, seul le champ de **temps restant** subsiste et est vraiment utile). Pour rappel, on distingue dans un projet Scrum généralement 4 grands rôles : les développeurs, les tes-



teurs, le Product Owner et le Scrum Master. Si la fonction des développeurs et des testeurs fonctionnels ne sont plus à expliquer, le rôle de Product owner et de Scrum Master sont bien souvent méconnus du public.

Le Product Owner est la personne dont le rôle est de faire le lien entre l'équipe de développement et le client final de l'application. C'est un rôle assez difficile car son travail aura une influence directe sur la qualité de l'application et sur l'efficacité des autres membres de l'équipe. Dans un premier temps, le Product Owner se charge de récupérer toutes les fonctionnalités demandées par les utilisateurs pour pouvoir alimenter le Product Backlog. Ce dernier se présente souvent sous la forme d'un document contenant toutes les fonctionnalités demandées par les utilisateurs avec leur description la plus complète possible ainsi que leur ordre de priorité. D'un point de vue Team Foundation Server, ces fonctionnalités sont matérialisées par des Work Items de type User Story, et c'est à partir de celles-ci que l'équipe peut diviser en tâches techniques (Task) l'implémentation d'une nouvelle fonctionnalité dans l'application. Le Product Owner doit alors être capable de répondre à toutes les questions que pourrait se poser l'équipe durant le découpage et la planification, voire même la réalisation, de ces fonctionnalités qui impliquent un point de vue utilisateur.

Si dans la version 2010 de Team Foundation Server le Product Owner manipule généralement un fichier Excel, dans cette nouvelle version de Team Foundation Server 11, une interface dédiée à la gestion du Product Backlog est disponible depuis le Team Web Access et permet de créer, prioriser et définir rapidement une User Story ou encore un Bug. Le Scrum Master quant à lui est le garant du bon déroulement du projet. Il a

principalement un rôle de "facilitateur" pour l'équipe dans l'application de la méthodologie Scrum au travers de l'orchestration des quatre principaux meetings: Le Sprint Planning, Le Daily Scrum, Le Sprint Review et le Sprint Retrospective. C'est lui qui veille à la bonne gestion du temps de chacun de ces meetings, ainsi que la focalisation sur les bons sujets de la part de tous les participants des meetings. Au quotidien, le Scrum Master est également la personne en charge de régler tous les soucis rencontrés susceptibles de freiner l'évolution du projet, notamment tous ceux liés à des problèmes d'infrastructure, technologiques, etc.

> Démarrage d'un projet Scrum

Lors du démarrage d'un projet Scrum, après avoir créé un projet d'équipe et réalisé le Storyboarding de l'application, il faut commencer à formaliser toutes les fonctionnalités demandées par les utilisateurs, soit toutes les User Story, puis commencer avec l'équipe par le meeting de Sprint Planning.

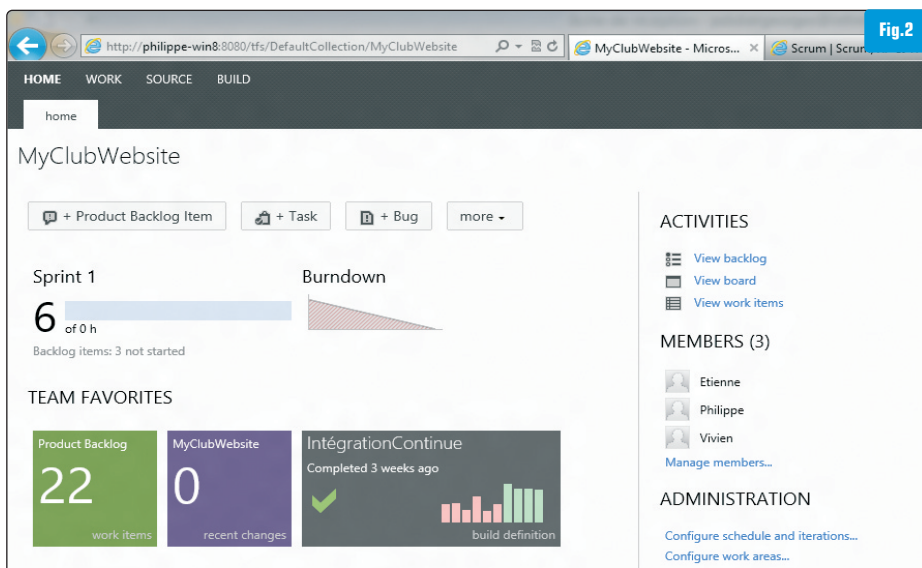
Les premiers éléments à définir sont les dates de début et de fin des différents Sprint. On peut également regrouper plusieurs Sprint dans un conteneur plus global appelé Release qui correspond généralement à une version de l'application destinée à être livrée en production. Ces dates sont un élément essentiel aux différents calculs effectués dans Team Foundation Server, notamment au niveau de l'affichage des nombreux rapports disponibles.

Dans le meeting de Sprint Planning, toute l'équipe commence à évaluer les efforts demandés par l'ajout d'une fonctionnalité dans l'application. Celles-ci sont mesurées en Story Points, correspondant à une estimation de la charge de travail, et permettent

ensuite de déterminer quelles fonctionnalités seront implémentées durant le premier Sprint, notamment en tenant compte de la vélocité (le nombre de Story Points pouvant être réalisé par l'équipe lors de chaque Sprint) de l'équipe. Le rapport de vélocité présent sous forme de graphique contenu dans un fichier Excel dans la version 2010 de Team Foundation Server est dorénavant présent directement à partir du Team Web Access.

Une fois que les membres de l'équipe se sont mis d'accord sur les User Stories qui seront implémentées durant le premier Sprint, ils peuvent ensuite procéder au découpage de celles-ci en tâches techniques à accomplir pour les réaliser. De nouveau, une estimation en heures du temps pris pour la réalisation des tâches est effectuée par l'équipe, afin d'évaluer si l'estimation initiale correspond bien à la durée d'un Sprint. Si la durée cumulée de toutes les tâches dépasse le temps de disponibilité de tous les membres de l'équipe, on retire alors une ou plusieurs User Story pour éviter de surcharger le Sprint. À l'inverse, si la durée cumulée des tâches est largement inférieure au temps de disponibilité des membres de l'équipe, on rajoute des User Stories pour utiliser le temps disponible. À la fin du Sprint Planning, il ne faut évidemment pas oublier de répartir les tâches entre les différents acteurs du projet, et également vérifier que la durée cumulée des tâches qui leur sont assignées est inférieure à leur disponibilité durant ce sprint. La disponibilité des différents membres du projet, auparavant paramétrable par le biais d'un fichier Excel dans la version 2010 de Team Foundation Server, l'est maintenant directement au travers du site Team Web Access [Fig.1].

Au quotidien, durant le Daily Scrum, les membres de l'équipe revoient tous ensemble les tâches accomplies par chacun, les tâches restant à accomplir ainsi que les potentiels problèmes (Issues) qui auraient pu être rencontrés. Les Issues sont alors renseignées directement dans Team Foundation Server et sont ensuite traitées par le Scrum Master durant la journée. Pour avoir une vision globale de l'avancement du projet, l'équipe utilise non seulement des graphiques (Burndown chart et Remaining Work) mais également le Scrum Dashboard pour suivre l'évolution des tâches sous forme de Post-it. Tous ces outils sont dorénavant directement disponibles depuis le Team Web Access.



> Le nouveau portail d'équipe

Le Team Web Access, ou portail d'équipe, prend encore plus d'importance dans cette nouvelle version 11 de Team Foundation Server et permet de définir plus facilement les membres d'une équipe participant à un même projet. S'il était nécessaire dans la version 2010 d'utiliser les groupes de sécurité pour cela, la nouvelle version de Team Foundation Server offre cette fois-ci une interface web qui permet de gérer beaucoup plus facilement les permissions de ces équipes.

Ainsi dès l'arrivée sur la page d'accueil du site du projet d'équipe, on voit en un coup d'œil la situation actuelle dans laquelle se trouve le projet. Sont affichés les dates de l'itération (Sprint) en cours, l'avancement du travail par un graphique type Burndown Chart simplifié avec le travail restant avant la fin de l'itération, ainsi qu'un accès direct au Backlog de produit, au tableau de sprint (Board) et à la liste des éléments de travail. À noter également sur cette interface, l'apparition de tuiles Metro qui correspondent aux différentes requêtes de Work Items, Définitions de Build ou chemins de Code source qui ont été marqués comme "Favoris" par l'équipe [Fig.2].

L'ouverture de l'onglet Work affiche le Product Backlog contenant la liste des exigences et permet de gérer et ordonner ces dernières. À ce niveau, on retrouve la possibilité de préciser la charge de réalisation (Effort) de chaque exigence et récupérer ainsi, après avoir saisi la capacité de charge de l'équipe, lors de quel Sprint il sera prévu de l'implémenter (Forecast).

En plus de proposer une interface épurée pour la saisie des Product Backlog Item (On n'a besoin de saisir que son titre pour pouvoir en créer un), le site web offre une manière très simple d'ordonner par priorité les Product Backlog Item en faisant des glisser-déposer.

À partir du Product Backlog, on possède également, en haut à droite, un accès vers un rapport, concernant la Vitesse de l'équipe, visible et mis à jour en temps réel depuis le Team Web Access.

L'itération en cours et les itérations prochaines sont listées sur le côté gauche et il est alors possible d'assigner les éléments du backlog aux itérations auxquelles on souhaite les associer par un simple glisser-déposer. Une fois positionné sur un Sprint, c'est cette fois-ci le rapport de Burndown, visible également sur la page d'accueil du projet d'équipe,

qui sera affiché et mis à jour en temps réel. L'une nouveautés clé de cette dernière version de Team Foundation Server: l'apparition d'un tableau de bord en mode Post-it. Dans des projets réalisés selon la méthode Scrum, l'équipe dispose généralement d'un tableau ou d'un mur dans lequel se fait la gestion de l'attribution et de l'avancement des tâches de chacun, à l'aide de post-it que l'on déplace. Il s'agit ni plus ni moins que de ce tableau que l'on retrouve dans l'onglet board [Fig.3].

Celui-ci est lié en temps réel aux données de TFS. Cela signifie simplement que le fait de déplacer un post-it, en glisser-déposer, de la colonne TO DO à la colonne IN PROGRESS va immédiatement répliquer cette action partout dans Team Foundation Server et pour tous les membres de l'équipe. Au quotidien, chaque membre de l'équipe pourra manipuler les Work Items qui leur ont été assignés au travers de leur outil dédié (Visual Studio, Microsoft Test Manager, etc.) et leurs actions seront directement visibles depuis ce tableau de bord. Toutes ces fonctionnalités, très orientées vers le glisser-déposer, sont utilisables en environnement tactile.

On pourra visualiser les tâches de ce tableau de bord de deux manières différentes, soit en les triant par Product Backlog Item, soit en les triant par membres du projet d'équipe à qui ont été assignées ces tâches.

L'onglet Work Item permet l'affichage des Work Items de la même façon qu'il était possible de le faire dans les versions précédentes de Team Foundation Server à l'aide de requêtes personnalisables qui offrent une grande flexibilité d'utilisation.

L'onglet Sources est ni plus ni moins qu'une version web du gestionnaire de source comme on peut le retrouver avec le Team Explorer notamment dans Visual Studio. Il permet de naviguer dans les dossiers du contrôleur de code sources, mais également de consulter les changeset liés aux check-in de l'équipe mais aussi les parties de codes mis « sur étagère » (shelveset).

Cette interface offre la navigation dans l'historique des versions des sources, mais également des fichiers générés par les builds.

L'onglet Build permet d'accéder à la liste des Builds terminées ainsi que celles en attente d'exécution. Il est possible de consulter les rapports d'exécution détaillés des Builds terminées et de suivre celles en cours en direct comme on le ferait depuis Visual Studio.

Pour rappel, le service de Build de Team Foundation Server est un outil permettant la création de Workflow d'automatisation de

compilation du code, mais également d'autres opérations personnalisables telles que la validation préalable de convention de codage, mais également l'exécution de tests automatisés, la génération de packages ou bien encore un déploiement automatisé. Tout ceci est paramétrable et personnalisable par une personne disposant des droits de planification de Builds.

> La nouvelle notion d'équipe

Cette nouvelle version de Team Foundation Server marque un tournant en intégrant toujours plus les méthodologies agiles de développement aux outils fournis aux développeurs. La notion d'équipe prend alors tout son sens dans la gestion notamment des droits dans nos projets.

À la création du projet, une équipe de projet est immédiatement créée et c'est cette équipe qui est mise en avant dès la page d'accueil du Team Web Access jusqu'à la gestion des éléments de travail.

En effet, une fois le projet créé sur ma collection, il suffit d'ajouter, par leur compte utilisateur ou avec des groupes de sécurité, les membres à l'équipe depuis Team Web Access afin qu'ils puissent immédiatement contribuer au projet et disposer des droits nécessaires à l'utilisation de tous les outils. Cela simplifie grandement l'opération en comparaison aux précédentes versions.

Une fois le projet démarré, la notion d'équipe est encore là, afin de nous aider à planifier notre charge de travail. Il est alors possible de répartir efficacement les éléments de travail entre les membres de l'équipe en prenant en compte, par exemple, des congés d'un membre ou l'autre dans la charge de travail qu'on lui attribue. La gestion du travail sur la durée du sprint est alors gérée au niveau de l'équipe et non plus du projet.

> Conclusion

Ont été présentés dans ce second article les principes de la méthodologie Agile Scrum et les outils offerts par Team Foundation Server 11 permettant sa mise en œuvre dans le cadre de la réalisation d'un projet.

Dans le prochain article seront écrites les premières lignes de code de notre application Metro Windows 8 SubwayReader.

Philippe **Didiergeorges** et Vivien **Fabing**
Consultants équipe ALM

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

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

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

Premiers pas avec Zend Framework

Face à Java EE ou .Net, PHP s'impose peu à peu dans le milieu professionnel. Certains outils lui permettent aujourd'hui de s'y faire une place. C'est le cas de frameworks tels que Symfony ou Zend Framework qui permettent de structurer et d'industrialiser les développements et apportent ainsi au langage une meilleure visibilité au sein des entreprises. Cet article propose une première approche de Zend Framework.

Zend Framework est un framework pour PHP 5 développé par Zend Technologies et sous licence BSD. Il permet de structurer les développements en PHP tout en mettant en avant les bonnes pratiques du développement orienté objet. Complet et bien documenté, il offre toutes les briques nécessaires à la réalisation d'applications professionnelles. Les différents composants du framework sont suffisamment dissociables pour ne pas nous obliger à baser toute l'architecture de l'application sur Zend Framework.

La version 2 du framework, pour laquelle la compatibilité ascendante n'est pas assurée, est en cours de développement. Une première version bêta est disponible à l'heure où nous écrivons ces lignes, mais il est préférable de continuer à utiliser la dernière version stable de la branche 1.11 pour tout projet ayant une mise en production planifiée à court ou à moyen terme, car la date de sortie de la version finale et stable de Zend Framework 2.0 n'est pas encore déterminée. Dans cet article, nous nous appuyons sur un exemple simple pour aiguiller un nouvel utilisateur dans l'apprentissage de ce framework. L'objectif est de mettre en place un environnement de développement et de se familiariser avec Zend Framework en réalisant une première page et un premier formulaire au sein d'une application MVC. Il ne nous est pas ici possible de rentrer dans les détails du fonctionnement des différents composants mis en œuvre, certaines explications sont passées sous silence par souci de concision.

PRÉSENTATION DE L'APPLICATION EXEMPLE

L'exemple présenté ici est avant tout destiné à proposer une première expérience simple avec Zend Framework. Il s'agit d'une interface permettant à différents participants de se mettre d'accord sur un lieu où aller prendre le déjeuner en utilisant la méthode du vote par assentiment : chaque utilisateur sélectionne un ou plusieurs restaurants / lieux où il souhaite déjeuner, celui remportant le plus de voix étant naturellement choisi. Il s'agit là d'une question épineuse que nous avons régulièrement à résoudre au bureau entre collègues ! Cette petite application utilise l'implémentation du patron d'architecture MVC (Modèle-Vue-Contrôleur) proposée par Zend Framework. Elle consiste en une simple page de présentation du résultat des votes du jour ainsi qu'une page de formulaire pour voter. Elle permet de présenter l'outil en ligne de commande fourni avec le framework, couteau suisse du développeur utilisant Zend Framework, ainsi que certains des principaux composants.

ENVIRONNEMENT DE DÉVELOPPEMENT

L'exemple présenté ici a été testé dans l'environnement suivant :

- Apache avec « mod_rewrite » activé et acceptant les fichiers « .htaccess »

- PHP version 5.3.6 (5.2.4 minimum pour faire fonctionner Zend Framework version 1.11.11)

- Zend Framework version 1.11.11

Côté base de données, nous utilisons ici SQLite qui permet d'éviter l'installation et la configuration d'un SGBD, mais Zend Framework peut être utilisé avec les SGBD du marché.

INSTALLATION DU FRAMEWORK

Tout d'abord, il faut récupérer l'archive « Zend Framework 1.11.11 Full » sur la page de téléchargement du site de l'éditeur : « <http://framework.zend.com/download/current/> ».

L'installation du framework ne pose pas de problème particulier. Il s'agit principalement d'une bibliothèque PHP qui doit être désarchivée et « visible » par l'application. Il faut donc l'ajouter au projet PHP une fois celui-ci créé, ou l'ajouter dans l'« include path » dans le fichier de configuration « php.ini ». Pour simplifier le développement, Zend Framework est livré avec un outil en ligne de commande, situé dans le répertoire « bin » de l'archive, permettant d'exécuter automatiquement un certain nombre de traitements. Très pratique en environnement de développement, il n'est par contre pas indispensable au fonctionnement de l'application.

L'outil en ligne de commande nécessite un peu de configuration pour être reconnu comme commande intégrée afin de ne pas devoir préciser, à chaque utilisation, le chemin complet. Sous les systèmes de type Unix (Linux ou OS X), cela peut être fait en ajoutant l'alias suivant au fichier « .bash_profile » du répertoire utilisateur :

```
alias zf=chemin/vers/ZendFramework/bin/zf.sh
```

où « chemin/vers/ZendFramework » est le répertoire où est décompressée l'archive Zend Framework. Pour vérifier que l'alias est bien effectif, il suffit de lancer la commande zf pour lui demander, par exemple, la version de Zend Framework :

```
$ zf show version
```

qui devrait retourner (au numéro de version près) :

```
Zend Framework Version: 1.11.11
```

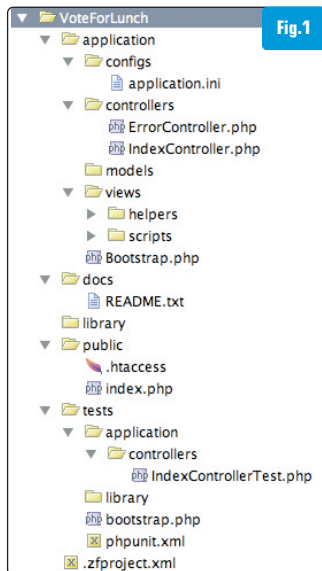
Le pendant Windows existe sous la forme du script « bin/zf.bat » dont on pourra ajouter le chemin dans la variable d'environnement PATH.

CRÉATION ET CONFIGURATION DU PROJET

Lors de la réalisation d'un nouveau projet MVC avec Zend Framework, la première étape consiste à créer l'arborescence du projet et les fichiers principaux. L'outil en ligne de commande permet de générer une arborescence de projet par défaut. Pour cela, on se place dans le répertoire désiré pour le développement de l'application et

on utilise la commande suivante qui prend en paramètre le nom du projet :

```
$ zf create project VoteForLunch
```



Arborescence par défaut d'une application MVC Zend Framework

Cette commande crée un répertoire « VoteForLunch » qui contient l'arborescence représentée en [Fig.1]. Les fichiers générés contiennent le squelette des différentes classes de base.

Toutes les lignes de commande présentées par la suite sont à exécuter depuis le répertoire racine de l'application qui vient d'être générée : « VoteForLunch/ ».

Dans ce répertoire nouvellement créé, nous pouvons noter l'existence par défaut d'un répertoire consacré à MVC, nommé « application ». Il contient lui-même un répertoire pour les contrôleurs, un répertoire pour les vues, ainsi qu'un répertoire pour les modèles. Cette application MVC fait usage notamment des compo-

sants « Zend_Application », « Zend_Controller » et « Zend_View » du framework. L'application peut se paramétrer en grande partie par son fichier de configuration « application/configs/application.ini » découpé en différentes sections correspondant aux différents environnements (« [développement] », « [test] », « [production] ») de l'application, ainsi que par sa classe d'amorçage ou « Bootstrap » (fichier « application/Bootstrap.php ») qui comme son nom l'indique contient tout ce qui est nécessaire à l'initialisation de l'application.

Remarque : le composant « Zend_Application » fait également usage d'un « autoloader » qui permet de faire automatiquement le lien entre une classe et le chemin vers le fichier dans lequel cette dernière est définie en se basant sur des conventions de nommage. Nous n'avons pas besoin d'en maîtriser ici le fonctionnement exact, car nous utilisons l'outil en ligne de commande qui respecte ces conventions. Cependant, il est important d'en connaître l'existence pour aller plus loin avec le framework.

Le répertoire « public » est le répertoire de l'application web qui doit donc être lisible par le serveur web. Il contient le fichier « index.php » qui conjointement au fichier « .htaccess », qui lui redirige toutes les requêtes, remplit le rôle de contrôleur frontal (Front Controller). Le répertoire « docs » est destiné à recueillir la documentation relative à l'application. Le répertoire « tests » contient les tests unitaires du projet mais nous n'en parlons pas dans cette prise en main. Enfin, le répertoire « library » contient, comme son nom l'indique, toutes les bibliothèques nécessaires à l'exécution de l'application. Vite par défaut, nous allons y ajouter la bibliothèque de Zend Framework précédemment désarchivée. Pour ce faire, nous créons un lien symbolique (pour les systèmes Unix) de la façon suivante :

```
$ ln -s chemin/vers/ZendFramework/library/Zend library/Zend
```

Il est également possible de simplement copier la bibliothèque dans

le répertoire « library ». Le lien symbolique permet l'utilisation de la même version de Zend Framework dans plusieurs projets distincts.

A ce stade, le projet est opérationnel et l'application par défaut peut être lancée. Il nous reste cependant à configurer le serveur web de développement. Nous supposons ici que le serveur de développement se trouve sur la machine de développement. Le site web se trouve dans « VoteForLunch/public ». Il est possible de configurer un hôte virtuel sous Apache comme spécifié dans le fichier « docs/README.txt », ou simplement de le configurer pour qu'il utilise le répertoire « public » de cette application comme répertoire web par défaut. De cette façon, l'accès au projet se fait très simplement par l'URL « <http://localhost/> ».

Remarque : une application MVC Zend Framework utilise le patron de conception (design pattern) « Front Controller ». Toutes les requêtes sont routées au travers de la même page « index.php ». Le contrôleur frontal est en charge d'aiguiller les demandes vers les bons contrôleurs d'actions. Les URL se forment de la façon suivante : « <http://localhost/controller/action/parameters> ». Lorsqu'aucun contrôleur n'est précisé, c'est l'« IndexController » qui est appelé. Lorsqu'un contrôleur est demandé sans préciser d'action, c'est l'action index du contrôleur donné qui est appelée.

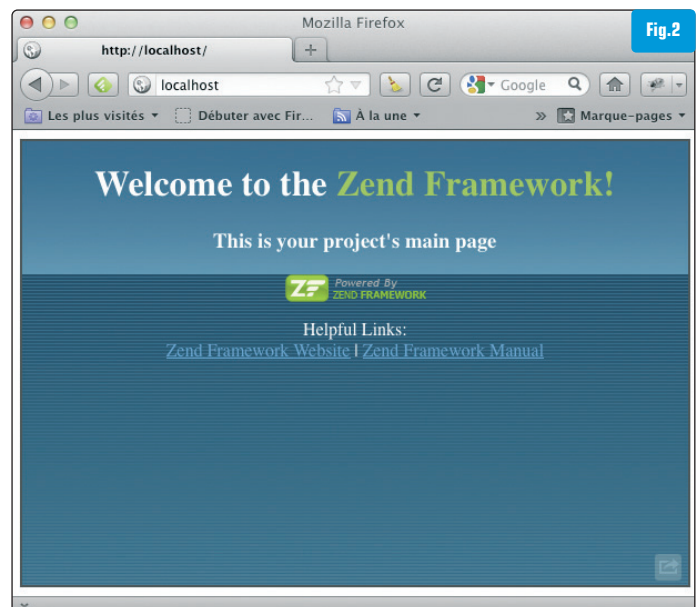
Il est important, en environnement de production particulièrement, de ne pas exposer toute l'arborescence du projet dans le serveur web. Seul le répertoire « public » doit être visible, pour des raisons évidentes de sécurité. Si le serveur est correctement configuré, en saisissant l'URL dans le navigateur, on obtient la page d'accueil par défaut présentée en [Fig.2].

On peut configurer l'environnement d'exécution de l'application (par défaut « [production] ») en renseignant la variable d'environnement « APPLICATION_ENV ». Ceci peut se faire dans le fichier « public/.htaccess » en y ajoutant cette ligne :

```
SetEnv APPLICATION_ENV development
...
```

ou directement dans la configuration du « VHost » comme indiqué dans le fichier « docs/README.txt ».

L'application est maintenant prête à recevoir ses premières pages !



Page d'accueil par défaut d'une application ZF

MODÈLES ET BASE DE DONNÉES

La première étape avant la réalisation de la première page en elle-même est la création d'une base de données avec une table simple permettant de recueillir les différents votes des utilisateurs. Pour créer la base de données (un simple fichier dans le cas de SQLite), le client en ligne de commande Sqlite3 suffit amplement. Il existe également des clients graphiques pour SQLite. Il est enfin possible de réaliser un script PHP chargé de la génération de la base de données suivant le type d'environnements (production, développement, test, etc.) en fonction de la configuration que nous allons préciser ci-dessous dans application.ini. Dans notre cas, nous créons la base de données « voteforlunch.db » dans le répertoire « data » avec l'outil en ligne de commande :

```
$ mkdir data
$ sqlite3 data/voteforlunch.db
```

Une fois l'outil lancé, nous créons la table avec la commande SQL suivante :

```
CREATE TABLE vote (
    id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    username VARCHAR(32) NOT NULL DEFAULT 'noname',
    restaurantname VARCHAR(32) NOT NULL DEFAULT 'noname',
    votedate DATE NOT NULL
);
```

Remarque : afin de simplifier l'exemple, nous ne créons pas ici les tables « restaurant » et « utilisateur » avec jointures vers ces tables depuis la table « vote ».

Et nous y insérons ces lignes de test :

```
INSERT INTO vote (username, restaurantname, votedate) VALUES
('jean', 'japonais', DATE('NOW'));
INSERT INTO vote (username, restaurantname, votedate) VALUES
('georges', 'chinois', DATE('NOW'));
INSERT INTO vote (username, restaurantname, votedate) VALUES
('brad', 'chinois', DATE('NOW'));
INSERT INTO vote (username, restaurantname, votedate) VALUES
('jean', 'chinois', DATE('NOW'));
```

La commande SQL :

```
SELECT * FROM vote;
```

doit alors retourner (à la date près) :

```
1|jean|japonais|2012-02-22
2|georges|chinois|2012-02-22
3|brad|chinois|2012-02-22
4|jean|chinois|2012-02-22
```

On peut alors quitter sqlite3 ce qui se fait avec la commande

```
.quit
```

Une fois la base de données créée, il faut indiquer à l'application comment l'utiliser. Pour cela on peut compléter le fichier de configuration « application/configs/application.ini » ou utiliser l'outil en ligne de commande qui peut le faire pour nous :

```
$ zf configure db-adapter 'adapter=PDO_SQLITE&dbname=APPLICATION_PATH «../data/voteforlunch.db»' production
```

Cette commande permet d'ajouter la configuration de la base de données à la section « production » du fichier de configuration. Par défaut, les différentes sections héritent en premier lieu de la section « production », mais il est possible de surcharger les éléments qui diffèrent. On pourrait donc configurer l'utilisation d'une base de données spécifique à l'environnement de développement avec une commande ressemblant à ceci :

```
$ zf configure db-adapter 'adapter=PDO_SQLITE&dbname=APPLICATION_PATH «../data/voteforlunch-dev.db»' development
```

Pour cet exemple, nous nous contentons d'utiliser la même base de données pour les différents environnements. Le fichier « application.ini » doit comporter deux nouvelles lignes dans la section « [production] » :

```
; application/configs/application.ini

[production]
; ...
resources.db.adapter = «PDO_SQLITE»
resources.db.params.dbname = APPLICATION_PATH «../data/voteforlunch.db»
```

Avec cette configuration, l'application est en mesure de se connecter à la base de données. Il reste alors à créer une source de données utilisant cette base de données. Nous utilisons pour cela le patron de conception Table Data Gateway dont l'implémentation proposée par le framework est le composant « Zend_Db_Table » qui fait partie du composant « Zend_Db », ainsi qu'un autre patron de conception, le Data Mapper, pour l'interconnexion entre la source de données et le modèle.

Le Table Data Gateway est généré grâce à la ligne de commande :

```
$ zf create db-table Vote vote
```

qui permet de générer un nouveau fichier dans le répertoire nouvellement créé « application/models/DbTable » : « Vote.php ». Ce fichier est très simple, et le restera :

```
<?php

class Application_Model_DbTable_Vote extends Zend_Db_Table_Abstract
{

    protected $_name = 'vote';

}
```

Très peu de paramètres sont nécessaires, « Zend_Db_Table » utilisant la définition de la table pour fonctionner (notamment la clé primaire). Il est cependant possible de configurer manuellement un certain nombre de paramètres, comme stipulé dans la documentation. Créons maintenant le Data Mapper en charge de la liaison entre « Zend_Db_Table » et le modèle.

```
$ zf create model VoteMapper
```

Le fichier « application/models/VoteMapper.php » est créé. Il contient le squelette de la classe « Application_Model_VoteMapper » que l'on complète de la façon suivante :

```
<?php
```



```

class Application_Model_VoteMapper
{
    protected $_dbTable;

    public function setDbTable($dbTable)
    {
        if (is_string($dbTable)) {
            $dbTable = new $dbTable();
        }
        if (!$dbTable instanceof Zend_Db_Table_Abstract) {
            throw new Exception('Invalid table data gateway provided');
        }
        $this->_dbTable = $dbTable;
        return $this;
    }

    public function getDbTable()
    {
        if (null === $this->_dbTable) {
            $this->setDbTable('Application_Model_DbTable_Vote');
        }
        return $this->_dbTable;
    }

    public function save(Application_Model_Vote $vote)
    {
        //Un utilisateur ne peut voter qu'une fois par jour pour
        un restaurant donné
        if (null == $this->getDbTable()->fetchRow(
            $this->getDbTable()->select()->where('votedate = ?', date(
                'Y-m-d'))
            ->where('username = ?', $vote->getUsername())
            ->where('restaurantname = ?', $vote->getRestaurantname())
        )) {

            $data = array(
                'username' => $vote->getUsername(),
                'restaurantname' => $vote->getRestaurantname(),
                'votedate' => date('Y-m-d'),
            );

            $this->getDbTable()->insert($data);
        }
    }

    public function findByDate($date)
    {
        //récupération des votes pour une date donnée
        $resultSet = $this->getDbTable()->fetchAll(
            $this->getDbTable()->select()->where('votedate = ?', $date)
            ->order('username ASC')
        );
        $entries = array();
        foreach ($resultSet as $row) {
            $entry = new Application_Model_Vote();
            $entry->setId($row->id)
                ->setUsername($row->username)
                ->setRestaurantname($row->restaurantname)

```

```

                ->setVotedate($row->votedate);
            $entries[] = $entry;
        }
        return $entries;
    }

    public function getSummaryArray($date)
    {
        //récupération d'un tableau résumant le nombre de votes
        par restaurant
        $table = $this->getDbTable();
        $select = $table->select();

        $select->from($table,array('restaurantname','COUNT(*) as count'))
            ->where('votedate = ?', $date)
            ->group('restaurantname')
            ->order('count DESC');

        $resultSet = $table->fetchAll($select);

        $entries = array();
        foreach ($resultSet as $row) {
            $entry = Array(
                'restaurantname'=>$row->restaurantname,
                'count'=>$row->count
            );
            $entries[] = $entry;
        }
        return $entries;
    }
}

```

Remarque : des fonctions classiques du « Data Mapper », telles que « find » [qui prend en paramètre un identifiant permettant de filtrer sur la clé primaire de la table], sont ici volontairement omises, par souci de clarté, car elles ne sont pas utilisées.

La classe de modèle en elle-même peut maintenant être créée :

```
$ zf create model Vote
```

Ceci génère le fichier « application/models/Vote.php ». On le complète de façon notamment à simplifier l'affectation de valeurs aux champs de l'objet « vote » à l'aide de la fonction « setOptions » qui prend un tableau en paramètre :

```

<?php

class Application_Model_Vote
{
    protected $_username;
    protected $_restaurantname;
    protected $_votedate;
    protected $_id;

    public function __construct(array $options = null)
    {
        if (is_array($options)) {
            $this->setOptions($options);
        }
    }
}

```



```

    }
}

public function __set($name, $value)
{
    $method = 'set' . $name;
    if (('mapper' == $name) || !method_exists($this, $method)) {
        throw new Exception('Invalid vote property');
    }
    $this->$method($value);
}

public function __get($name)
{
    $method = 'get' . $name;
    if (('mapper' == $name) || !method_exists($this, $method)) {
        throw new Exception('Invalid vote property');
    }
    return $this->$method();
}

public function setOptions(array $options)
{
    $methods = get_class_methods($this);
    foreach ($options as $key => $value) {
        $method = 'set' . ucfirst($key);
        if (in_array($method, $methods)) {
            $this->$method($value);
        }
    }
    return $this;
}

public function setUsername($text)
{
    $this->_username = (string)$text;
    return $this;
}

public function getUsername()
{
    return $this->_username;
}

public function setRestaurantname($text)
{
    $this->_restaurantname = (string)$text;
    return $this;
}

public function getRestaurantname()
{
    return $this->_restaurantname;
}

public function setVotedate($ts)
{
    $this->_votedate = $ts;

```

```

    return $this;
}

public function getVotedate()
{
    return $this->_votedate;
}

public function setId($id)
{
    $this->_id = (int)$id;
    return $this;
}

public function getId()
{
    return $this->_id;
}
}

```

La partie modèle est maintenant en place, ainsi que le lien avec la base de données. Nous allons maintenant nous intéresser au traitement des données et à leur restitution, tâches attribuées au contrôleur et à la vue.

CONTROLLER ET VUE

Le modèle sert de lien entre le contrôleur et la vue. Le contrôleur, rôle joué par le composant « Zend_Controller », s'occupe de traiter les requêtes du client (« GET » ou « POST ») et de préparer les données. La vue, composant « Zend_View », est en charge de la présentation des données.

Encore une fois, l'outil en ligne de commande nous permet de créer les fichiers aux bons endroits dans l'arborescence. Le contrôleur et la vue par défaut (index) sont créés en une seule commande :

```
$ zf create controller Vote
```

Cette commande crée la classe du contrôleur « VoteController » dans le fichier « application/controllers/VoteController.php », avec l'action « indexAction() » par défaut. La vue correspondant à cette action est également créée, dans le répertoire « application/views/scripts/vote/ ». Nous modifions l'action « indexAction() » de façon à ce qu'elle récupère les informations nécessaires à l'affichage du résumé et du détail des votes du jour.

```

public function indexAction()
{
    $vote = new Application_Model_VoteMapper();
    //récupération des votes du jour
    $this->view->allVotes = $vote->findByDate(date('Y-m-d'));
    $this->view->votesByRestaurant = $vote->getSummaryArray(date('Y-m-d'));
}

```

Pour présenter ces informations au client web, on complète la vue « application/views/scripts/vote/index.phtml » :

```

<p><a href="<?php echo $this->url(
    array(

```

```
'controller' => 'vote',
'action' => 'addvote'
),
'default',
true) ?>>Votez pour un restaurant</a></p>

<p>Résumé;sum&eacute;r du jour :</p>
<table>
  <tr>
    <th>Restaurant</th>
    <th>Nombre de votes</th>
  </tr>
  <?php foreach ($this->votesByRestaurant as $resum): ?>
    <tr>
      <td><?php echo $this->escape($resum['restaurantname']) ?></td>
      <td><?php echo $this->escape($resum['count']) ?></td>
    </tr>
  <?php endforeach ?>
</table>
<br/>

<p>D&eacute;tail des votes du jour : </p>
<ul>
  <?php foreach ($this->allVotes as $vote): ?>
    <li><?php echo $this->escape($vote->username) ?> :
      <?php echo $this->escape($vote->restaurantname) ?></li>
  <?php endforeach ?>
</ul>
```

Cette vue contient notamment un lien vers le formulaire de vote (que nous n'avons pas encore créé) généré à l'aide du « helper » `$this->url()`. A cette étape, nous pouvons tester la page correspondant à l'action « `indexAction()` » du contrôleur « `VoteController` ». Pour cela, rappelons que par défaut les url d'une application MVC développée avec Zend Framework sont de la forme « `contrôleur/action/paramètres` ». On peut donc tester l'action d'index du contrôleur « `VoteController` » en se rendant à l'adresse « <http://localhost/vote/index> » ou plus

simplement « <http://localhost/vote> », l'action « `index` » étant utilisée par défaut. Elle rassemble bien les enregistrements de test insérés manuellement lors de la création de la base de données, comme le montre la [Fig.3].

L'étape suivante consiste à ajouter une page de formulaire permettant aux utilisateurs de voter.

FORMULAIRE DE VOTE

Pour préparer la page qui va accueillir le formulaire de vote, nous utilisons l'outil en ligne de commande pour générer une nouvelle action et une nouvelle vue correspondante.

```
$ zf create action Addvote vote
```

Afin d'ajouter le formulaire, nous utilisons un outil proposé par Zend Framework, il s'agit des classes de formulaires « `Zend_Form` ». Nous pouvons générer un fichier de formulaire de la façon suivante :

```
$ zf create form Vote
```

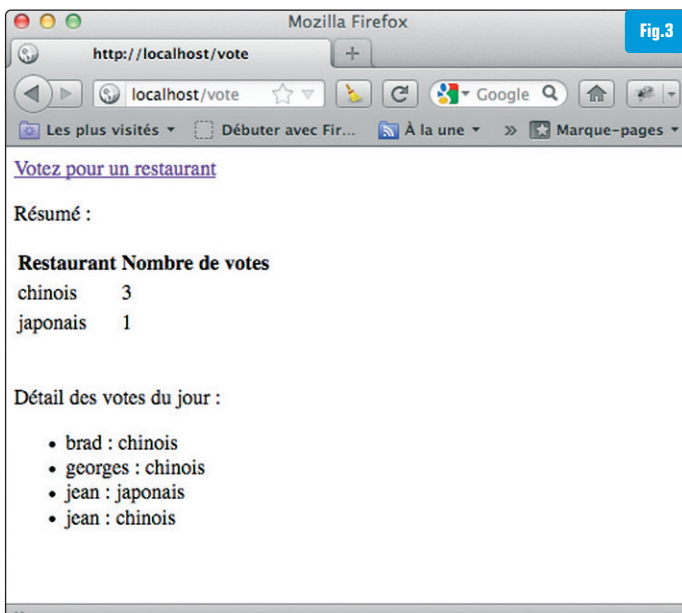
Cette commande permet de créer le fichier « `application/forms/Vote.php` ». Nous le complétons comme suit :

```
<?php
class Application_Form_Vote extends Zend_Form
{
    public function init()
    {
        // La méthode HTTP d'envoi du formulaire
        $this->setMethod('post');

        // Un élément pour le nom d'utilisateur
        $this->addElement('text', 'username', array(
            'label' => 'Votre nom :',
            'required' => true,
            'filters' => array('StringTrim'),
            'validators' => array(
                array('validator' => 'StringLength', 'options' => array(0, 20))
            )
        ));

        // Un élément pour le nom du restaurant
        $this->addElement('text', 'restaurantname', array(
            'label' => 'Votre choix de restaurant :',
            'required' => true,
            'filters' => array('StringTrim'),
            'validators' => array(
                array('validator' => 'StringLength', 'options' => array(0, 20))
            )
        ));

        // Un bouton d'envoi
        $this->addElement('submit', 'submit', array(
            'ignore' => true,
            'label' => 'Voter',
        ));
    }
}
```



Page de détail des votes du jour

Les formulaires Zend Framework sont complètement configurables, que ce soit tant au niveau du rendu graphique que des validateurs par exemple. Ils peuvent par ailleurs intégrer un certain nombre de protections, comme une protection contre les attaques CSRF ou encore contre les robots avec un Captcha.

On peut maintenant ajouter le formulaire dans la vue « application/views/scripts/vote/addvote.phtml » :

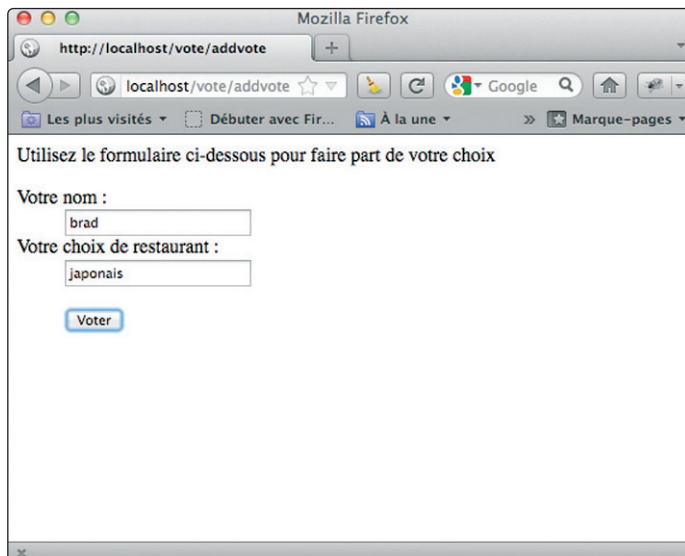
```
<p>
Utilisez le formulaire ci-dessous pour faire part de votre choix
</p>

<?php
$this->form->setAction($this->url());
echo $this->form;
```

L'action « addvoteAction() » est complétée pour y ajouter le traitement des données postées par le formulaire de vote :

```
public function addvoteAction()
{
    $request = $this->getRequest();
    $form = new Application_Form_Vote();

    if ($this->getRequest()->isPost()) {
        if ($form->isValid($request->getPost())) {
```



Formulaire d'ajout de vote

```
$vote = new Application_Model_Vote($form->getValues());
$mapper = new Application_Model_VoteMapper();
$mapper->save($vote);
return $this->_helper->redirector('index');
}

$this->view->form = $form;
}
```

Les deux pages que nous souhaitions mettre en place sont maintenant fonctionnelles. Le formulaire se trouve à l'adresse « <http://localhost/vote/addvote> » et est également accessible par le lien présent sur la page « <http://localhost/vote> ».

La page de formulaire est représentée en [Fig.4].

Attention : dans cet exemple, le système de « layout » (« zend_layout ») n'est pas activé et les pages HTML renvoyées ne sont pas valides. En effet, c'est le squelette du site qui doit contenir (entre autres) l'entête HTML et préciser l'encodage utilisé. Par conséquent, l'utilisation de caractères accentués peut poser problème, mais cela se résout facilement en créant le squelette du site ou en ajoutant temporairement l'entête HTML dans chacune des vues.

POUR ALLER PLUS LOIN

Nous avons pu appréhender quelques bases de Zend Framework. Nous avons mis en place une version minimaliste d'application MVC contenant deux pages et un formulaire. Nous avons utilisé, principalement, les composants « Zend_Db », « Zend_Application », « Zend_Controller », « Zend_View », ou encore « Zend_Form ».

Pour aller plus loin dans l'apprentissage du framework, la documentation officielle est toute indiquée. Elle présente indépendamment les différents composants parmi lesquels on peut mentionner « Zend_Layout » pour la réalisation du squelette du site, « Zend_Cache » pour améliorer les performances de l'application, ou encore « Zend_Session », ... et des dizaines d'autres !

De plus, comme Zend Framework se veut modulaire, il est relativement aisé d'interchanger les composants pour utiliser des composants externes comme, par exemple, un ORM (ex : Doctrine) ou encore un moteur de Template (ex : Twig, bien que son intégration requière un peu plus de travail). De quoi répondre à tous les besoins !

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

Le mois des cadeaux (prolongation jusqu'au 30 juin 2012)

Abonnez-vous jusqu'à -50%

et pour un 1€ de + , recevez un "livre numérique" des Éditions ENI

Coupon d'abonnement page 43 et sur www.programmez.com/abonnement.php



Prise en main de Drupal Commerce

Drupal Commerce est une suite de e-commerce complète pour Drupal 7. Un outil qui, s'il est encore jeune, est souple et puissant. Prise en main.

Drupal 7 est une éclatante démonstration de la qualité des logiciels open source. Alors que Drupal 6 avait fait ses preuves comme CMS de référence pour les grands comptes, la communauté a repris son travail pour fournir quelque chose de très supérieur. Ainsi avec Drupal 7 viennent de meilleures performances et l'intégration en natif de quelques-uns des meilleurs modules d'extensions à Drupal 6, les modules Images et Content Construction Kit notamment, ce dernier permettant d'ajouter des champs de divers types à un type de contenu. Drupal Commerce, est une nouvelle suite de e-commerce, qui existe en parallèle de Ubercart qui a de son côté été porté sur Drupal 7. Drupal Commerce s'appuie tout naturellement sur les nouvelles fonctionnalités de Drupal 7. Avec lui, mettre en place un site de e-commerce s'avère très aisé, dès que l'on sait pallier une documentation pas vraiment à la hauteur, au moment de la rédaction de cet article. Un phénomène assez courant dans le monde de l'open source, personne n'étant parfait :) Mais quelle que soit la qualité de Drupal Commerce, le moment viendra où il sera nécessaire de lui ajouter quelques fonctionnalités. On écrira alors un module d'extension. Cet article se propose de montrer les manipulations de base pour ces deux opérations. Le seul prérequis à la lecture de cet article est d'être familiarisé avec l'environnement Drupal 6 ou 7.

1 NOTRE BUT

Nous allons donc mettre en place un site de e-commerce qui vend des compilateurs et des livres. On veut que le site présente un catalogue au visiteur. Bien entendu, comme pour tout site de e-commerce, une fonctionnalité de panier doit être en place. Nous supposons que le lecteur qui voudrait essayer les manipulations de cet article dispose d'un Drupal 7 installé sur sa machine. Nous supposons également que Drupal Commerce est installé. A la base, Drupal Commerce vient sous la forme d'un module Drupal comme un autre, et son installation

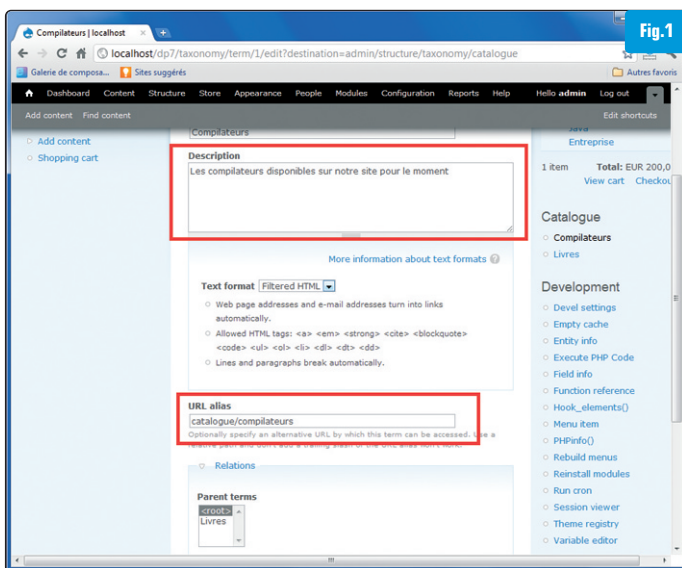
ne présente aucune difficulté. Il est recommandé d'installer également le module 'Development', accessible à drupal.org, car celui-ci propose un bloc pour vider le cache du système ou pour réinstaller un module en un clic de souris. Ce module se révèle très vite comme étant absolument indispensable pendant la phase de développement.

2 AU CENTRE, LA TAXINOMIE

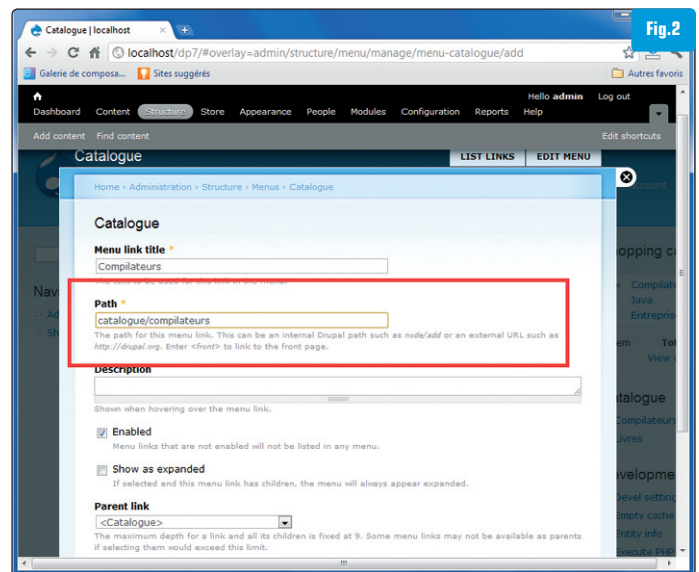
En fait notre premier but, la mise en place du site, ne présente qu'une seule difficulté, la carence de documentation. Celle-ci est écrite par des personnes qui connaissent Drupal dans les moindres recoins, et pour qui beaucoup (trop) de choses sont des évidences. Une de ces évidences est qu'une fonctionnalité clé de Drupal est la taxinomie (de l'anglais taxonomy, généralement traduit en taxonomie dans l'univers Drupal). Or si nous voulons que notre site présente un catalogue, nous devons penser qu'un catalogue n'est rien d'autre qu'un classement de contenu, donc une taxinomie. La première chose à faire est donc de créer un vocabulaire. Pour cela rendez-vous dans l'interface d'administration, puis cliquez sur structure, puis sur Taxinomie, puis sur Add vocabulary. Baptisez votre nouveau vocabulaire avec pertinence. Choisissez Catalogue, par exemple :) A présent, créez des termes pour classer les produits que vous voulez vendre sur votre site. Pour nous, les termes 'Compilateurs' et 'Livres' s'imposent d'eux-mêmes :) Le point clé est maintenant de renseigner correctement les termes. Soyez attentifs au contenu du champ description, car celui-ci sera automatiquement affiché lorsque le visiteur cliquera sur une catégorie du catalogue. Mais surtout, soignez l'alias d'URL pour chacun des termes du catalogue. Pour notre exemple vous donnerez :

catalogue/compilateurs
catalogue/livres

Et laissez les relations à leurs valeurs par défaut. [Fig.1].



Un vocabulaire est à la base de tout catalogue dans Drupal Commerce



Attribuez aux entrées de menu les URL des termes de vocabulaire, et vous obtiendrez automatiquement un catalogue fonctionnel.

3 UN CATALOGUE AU MENU

Une fois le vocabulaire défini, nous avons besoin de quelque chose qui le matérialise, c'est-à-dire quelque chose qui en affiche les termes pour le visiteur. Sous Drupal, ce sera un menu. Pour créer celui-ci, dans l'interface d'administration, cliquez sur 'Structure', puis sur 'Menus'. Cliquez maintenant sur 'Add menu' pour créer le nouveau menu. Baptisez-le tout simplement 'Catalogue'. Vous devez maintenant ajouter des entrées de menus.

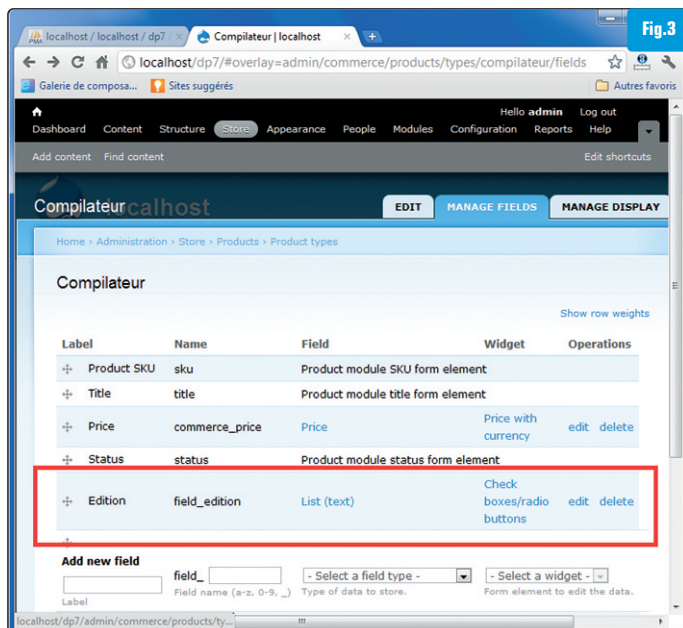
Pour cela cliquez sur 'Add Link'. Créez une entrée de menu par terme du vocabulaire précédemment créé. Donc pour notre exemple, nous aurons deux entrées de menus : Compilateurs et Livres. Et à chaque entrée de menu donnez l'URL (path) qui va bien. Le tour de magie avec Drupal consiste à donner la même URL que pour les termes du vocabulaire. Par exemple, pour 'Compilateurs', vous donnerez catalogue/compilateurs comme illustré [Fig.2]. Validez. Puis éditez à nouveau l'entrée de menu, et appréciez l'intelligence de Drupal. Celui-ci aura, de lui-même, remplacé l'URL par quelque chose comme :

```
taxonomy/term/1
```

C'est-à-dire qu'il aura automatiquement associé l'entrée de menu au terme de vocabulaire correspondant.

Ce point fort de Drupal, n'est malheureusement pas suffisamment mis en avant dans la documentation. Notre catalogue est prêt et ce n'était pas plus difficile que cela. Il nous reste à le mettre sous les yeux de l'utilisateur. Pour cela, dans l'interface d'administration, cliquez sur 'Structure' puis 'Blocks' pour arriver à l'interface d'administration des blocs. Vous constaterez que Drupal 7 a de lui-même créé un bloc de Catalogue.

Ce bloc affichera donc notre catalogue. Vous n'avez plus qu'à choisir, depuis cette page d'administration, à quel endroit de vos pages le bloc doit être affiché. Le catalogue est maintenant totalement terminé. Mais, pendant que vous êtes dans l'interface des blocs, faites également afficher le panier de Drupal Commerce, qui n'est rien d'autre qu'un bloc. Il ne nous manque plus que des produits pour remplir notre catalogue :-)



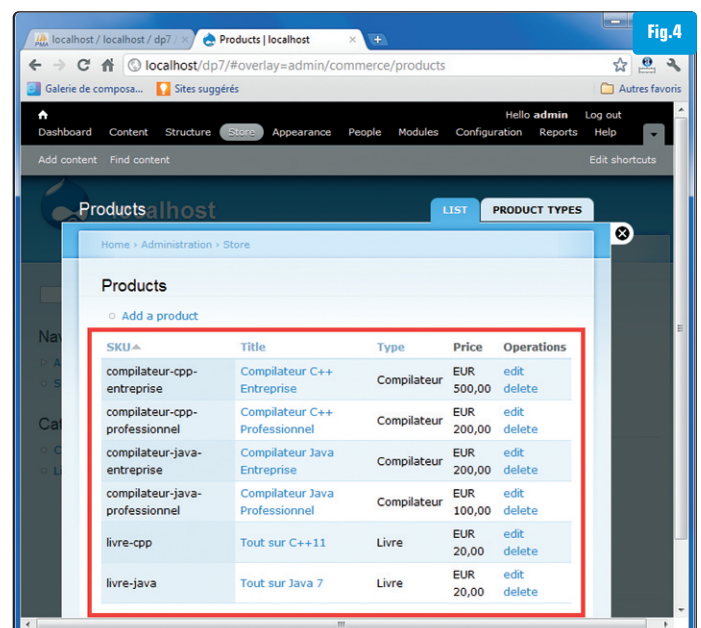
Créez des types de produits (entités) pour chaque catégorie de produits, en ajoutant au besoin des champs personnalisés.

4 ENTITÉ ET NOEUDS

Avec Drupal 7 est apparue l'entité (entity) comme abstraction de base. Sous Drupal 6 l'abstraction de base était le nœud (node). Autrement dit tout était un node... sauf ce qui n'en était pas, comme un utilisateur ou un commentaire. On pouvait donc reprocher à Drupal 6 un certain manque d'homogénéité. En outre un node pouvait s'avérer trop limité pour rendre certains contenus. Pour pallier cela est arrivé un module génial, CCK, qui permet d'ajouter des quantités de champs divers à un node Drupal 6. Mais pour ajouter un champ à un utilisateur (qui n'est pas un node), par exemple, il fallait écrire un module rien que pour cela, et dépendant de CCK. Au final, l'ensemble est assez complexe et ardu. Drupal 7 résout tout cela de façon très élégante, en amenant une abstraction en dessous du node, l'entité. Pour donner une idée simple des choses, l'entité est une structure de données, et c'est l'entité qui se trouve à la charnière du stockage des données, et non plus le node. Et quoi de plus naturel que de vouloir ajouter des champs à une structure de données ? C'est pour cela que les concepteurs de Drupal 7 lui ont intégré nativement le module CCK, qui devient la Field API. Ainsi nous pouvons créer, sans aucun souci, des structures de données aussi complexes que nous pouvons le souhaiter. Et sous Drupal 7 tout est entité cette fois. Un utilisateur, un vocabulaire, un menu, sont des entités, et nous avons cette fois l'homogénéité qui manquait à Drupal 6. Qu'est devenu le node dans Drupal 7 ? C'est tout une abstraction de niveau supérieur à l'entité, une entité spécialisée, et dédiée au stockage des contenus, ainsi qu'à leur rendu sur le site.

5 LES PRODUITS SOUS DRUPAL COMMERCE

Drupal Commerce embrasse pleinement la philosophie de Drupal 7. Les produits seront donc des entités, indépendantes de la représentation qu'en donneront des nodes. Nous devons donc commencer par créer ces entités. Cliquez sur 'Store' puis sur 'Products'. Cliquez alors sur l'onglet 'Product Types', expression consacrée pour entité produits. Créez un type pour les livres. Acceptez tous les champs par défaut. Puis créez un type pour les compilateurs. Enfin, pour ce



Créez tous les produits à vendre dans votre boutique, à partir des types précédemment définis.

type, ajoutez aux champs par défaut, par exemple, un champ 'Edition' qui vous permettra de différencier des compilateurs en édition professionnelle ou entreprise. (Merci à l'API Field :-). [Fig.3]. Ceci fait, revenez à la page produits (Products) en cliquant sur l'onglet 'List'. Créez alors un certain nombre de produits à vendre, des compilateurs C++, Java, des livres, etc. comme illustré. [Fig.4]. Quand vous aurez créé tous les produits que vous souhaitez vendre, vous verrez que ceux-ci ne s'affichent, pour l'instant, nulle part en dehors de l'interface d'administration. Vous verrez qu'il n'existe pas encore de lien entre le catalogue précédemment créé et nos produits. Ce comportement peut surprendre le lecteur qui viendrait d'Ubercart. Mais ce comportement est normal, dans le sens où il continue de suivre la philosophie Drupal 7. Autrement dit, les produits que vous avez créés ne sont que des entités. Nous touchons là un point très fort de Drupal Commerce, le produit, quel qu'il soit, 'existe' dans la boutique, indépendamment de toute représentation. Ceci signifie que pour un même produit, nous pouvons créer plusieurs représentations, comme par exemple un affichage standard, et un affichage qui fera état d'une promotion, ou un affichage auquel on lierait l'affichage d'un autre produit offert en cadeau, etc.

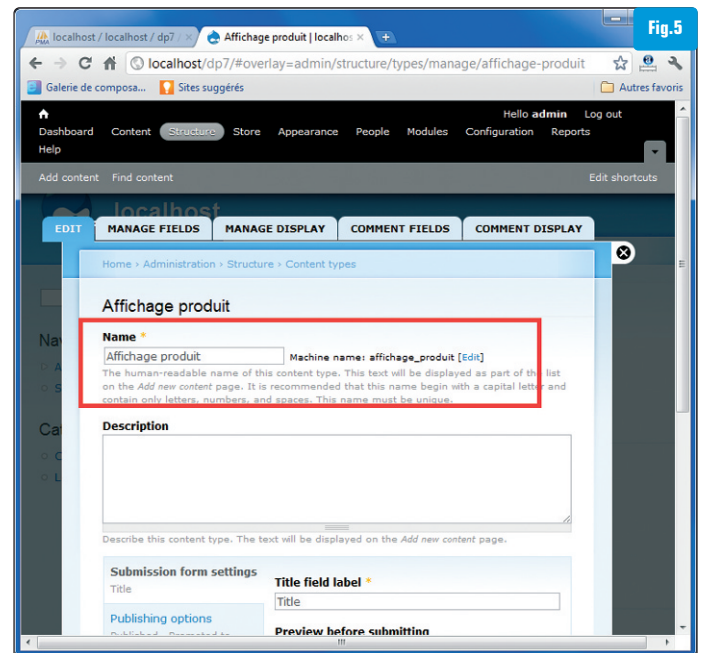
6 PRÉPARER LES AFFICHAGES DES PRODUITS

C'est ici que nous allons retrouver nos nodes et que nous allons profiter encore mieux de l'API Field intégrée. Dans l'interface d'administration, cliquez sur 'Structure', puis sur 'Content types', car nous allons ajouter un type de contenu. Telle est la dénomination sous Drupal 7, mais il est bon de savoir qu'un type de contenu est une entité :-). Cliquez sur 'Add content type'. Donnez par exemple le nom 'Affichage produit' à ce type de contenu, comme illustré. [Fig.5] Nous créons un seul type de contenu, qui conviendra pour afficher tous nos produits. Ce petit miracle est rendu possible par l'API Field. Dès que votre type de contenu est créé, ajoutez lui un champ de type 'Product reference' comme illustré. [Fig.6] Ce type de champ a été créé par Drupal Commerce lors de l'installation du module et c'est grâce à ce champ que vous pourrez associer un produit à un node,

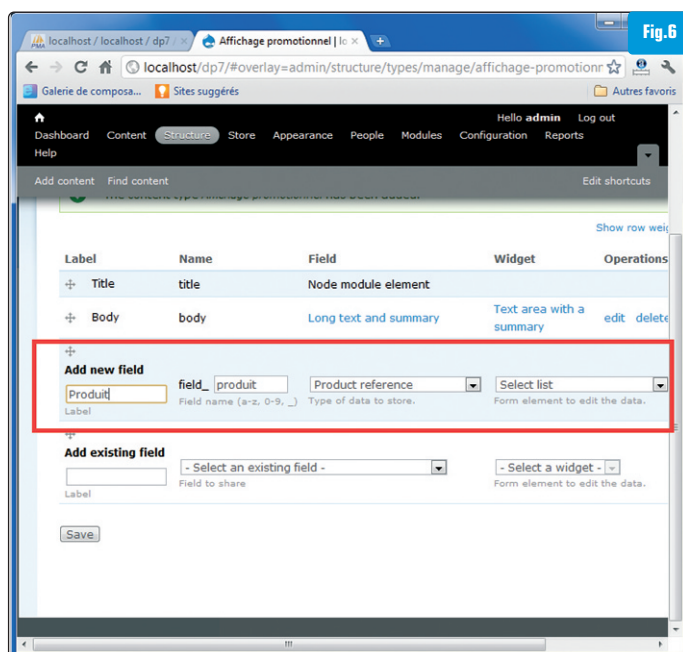
lors de la création de ce dernier. Pour que cette association puisse être faite, il vous reste à éditer le champ et à cocher les produits qui pourront être référencés, comme illustré [Fig.7]. Il ne reste plus qu'à lier notre type de contenu à notre catalogue, c'est-à-dire, fondamentalement, à notre taxonomie. Pour cela ajoutez un champ de type 'Term reference' comme illustré [Fig.8]. Lorsque vous validerez la création de ce champ, il vous sera demandé quel vocabulaire vous souhaitez lui affecter. Sélectionnez le vocabulaire Catalogue et validez, c'est terminé.

7 CRÉER DES AFFICHAGES DE PRODUITS

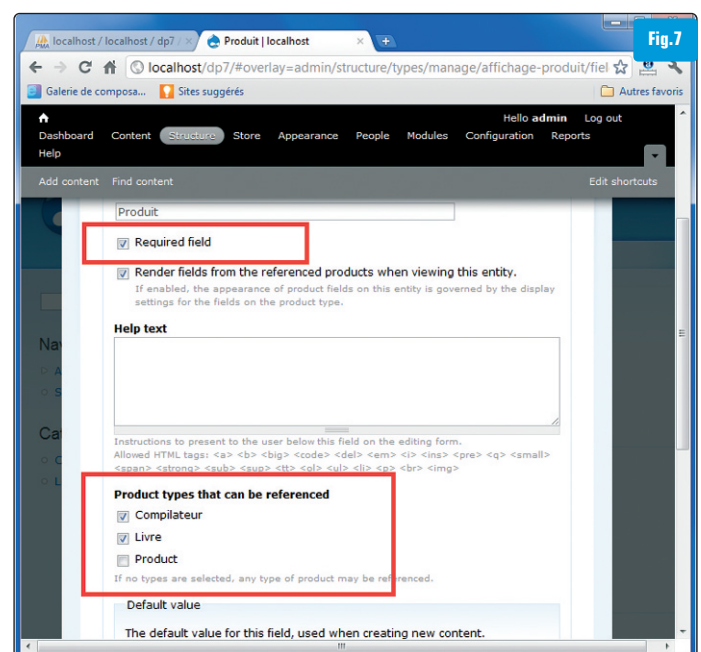
Tout étant maintenant prêt, créer des affichages de produits revient à créer des contenus (ou node) de type 'Affichage produit'. Lors de



Un même type de contenu conviendra pour afficher tous nos produits.



Ajoutez un champ de type 'Référence produit' (Product reference) à votre type de contenu d'affichage produit.



Sélectionnez les produits qui pourront être référencés par le champ 'Reference products'.

cette création, le formulaire vous demandera de spécifier quel est le produit sous-jacent, et à quelle catégorie du catalogue il appartient. Dès que vous aurez créé quelques affichages, vous aurez la satisfaction de constater que vous disposez d'un site complètement fonctionnel, avec catalogue et panier.

Il ne vous reste qu'à installer un module pour la fonctionnalité de paiement. Il en existe de nombreux disponibles à drupal.org. De nombreux autres modules d'extensions à Drupal Commerce, pour de nombreux usages, sont également disponibles sur drupal.org. Toutefois, vous éprouverez peut-être le besoin d'écrire vous-même un module pour mettre en place une fonctionnalité particulière. Voyons brièvement comment procéder.

8 ÉCRIRE UN MODULE D'EXTENSION À DRUPAL COMMERCE

Un module d'extension à Drupal Commerce est avant tout un module d'extension à Drupal. On déposera un tel module dans le sous-répertoire modules de Drupal Commerce.

Il est recommandé de respecter certaines conventions de codage et de nommage. Celles-ci sont documentées à www.drupalcommerce.org.

A titre d'exemple le lecteur trouvera sur notre site un module qui apporte quelques petites fonctionnalités.

Ce module permet d'afficher la devise à gauche d'un prix, il permet de définir une nouvelle monnaie dans la boutique, et il permet d'afficher une information complémentaire dans le champ du prix d'un produit. Le fichier .info de notre module est simple :

```
name = Commerce Programmez
description = Démonstration de module pour Drupal Commerce
core = 7.x
dependencies[] = commerce
package = Commerce (Contrib)
version = "7.x-1.0"
```

Nous indiquons que notre module dépend de Drupal Commerce, et nous le plaçons dans un package, dont le nom répond aux conven-

tions citées plus haut. Sous Drupal, les modules sont construits autour de hooks, qui sont des fonctions de rappel invoquées aux moments opportuns par le système.

Un module sous Drupal Commerce est construit de même. L'API Drupal Commerce est constituée d'un jeu de hooks. Lorsque le système invoque les hooks de Drupal Commerce, celui-ci se charge d'invoquer les hooks des modules placés sous lui tout simplement. Voici le code qui agit sur les devises :

```
/*
 * Implémentation de hook_commerce_currency_info()
 */
function commerce_programmez_commerce_currency_info() {
  return array (
    'ZZZ' => array(
      'code' => 'ZZZ',
      'numeric_code' => '999',
      'symbol' => 'zzz',
      'name' => t('Fausse monnaie'),
      'symbol_placement' => 'before',
      'code_placement' => 'before',
      'minor_unit' => t('Monnaie'),
      'major_unit' => t('Fausse'),
      'rounding_step' => '0.05',
    ),
  );
}

/*
 * Implémentation de hook_commerce_currency_info_alter(&$currencies,
 $langcode)
 */
function commerce_programmez_commerce_currency_info_alter(&$currencies, $langcode) {
  $currencies['EUR']['code_placement'] = 'before';
  $currencies['EUR']['symbol_placement'] = 'hidden';
  $currencies['EUR']['symbol'] = 'EUR';
}
```

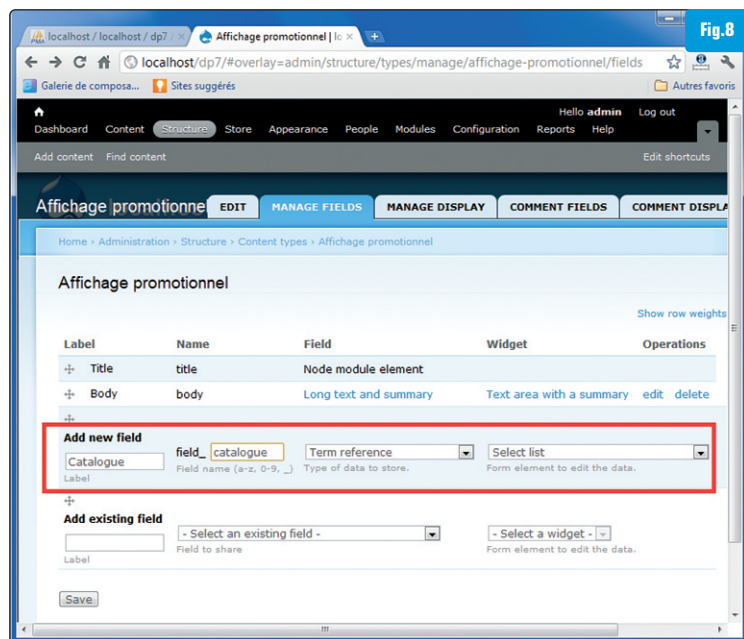
Rien d'extraordinaire dans ce code. Le premier hook définit une nouvelle devise qui devient disponible dans l'interface d'administration et si cette devise est sélectionnée par l'administrateur, elle sera affichée près du prix. Le second hook définit le positionnement de l'affichage de la devise courante.

Le lecteur qui consultera la documentation de Drupal Commerce s'étonnera inmanquablement de la faible taille de celle-ci, en regard des possibilités de l'outil.

L'explication se trouve dans le fait que Drupal Commerce s'appuie énormément sur la géniale API Field de Drupal. Ceci implique que pour écrire un module complexe à Drupal Commerce, il sera nécessaire de maîtriser cette API.

Notre module d'exemple, qui est loin d'être complexe, passe lui aussi par l'API Field pour afficher l'information complémentaire au prix d'un produit.

Nous étudierons en détail l'API Field dans un prochain article.



Ajoutez un champ de type 'Term reference' pour lier catalogue et type de contenu d'affichage produit.

Frédéric Mazué - fmazue@programmez.com

Programmation **SOLID** : une autre approche

SOLID est une notion s'articulant sur un ensemble de cinq principes de base de la programmation orientée objet. Il ne s'agit pas à proprement parler de règles de programmation, qui auraient pu être incorporées dans les langages, mais de principes permettant d'écrire du code plus facilement extensible et maintenable.

Les cinq principes sont :

- Responsabilité unique (le S de Single Responsibility Principle) : un objet doit ne s'occuper que d'une seule pratique correspondant au métier.
- Dualité ouvert / fermé (le O de Open / Closed Principle) : un objet doit être ouvert à l'extension, mais fermé à la modification.
- Substitution de Liskov (le L de Liskov Substitution Principle) : un objet doit pouvoir être remplacé par un autre qui le spécialise, sans que cela n'empêche le fonctionnement correct.
- Séparation des interfaces (le I de Interface Segregation Principle) : les fonctionnalités ne doivent être regroupées sur une interface que si elles ont un sens métier commun.
- Inversion de dépendance (le D de Dependency Inversion Principle) : un objet qui a besoin d'un service ne doit pas dépendre de ce service, mais à l'inverse le service doit dépendre du contrat que demande l'objet.

Ce qui est intéressant dans ces principes est qu'ils donnent une justification issue de l'expérience à l'utilisation de notions plus académiques comme l'encapsulation, le polymorphisme ou la programmation par interfaces contractuelles. Si vous développez dans un langage orienté objet et que vous ne maîtrisez pas encore ces notions, l'auteur est convaincu que passer quelques heures à bien les comprendre et savoir les appliquer est un des meilleurs investissements que vous puissiez réaliser, avec un retour sur la pratique meilleur que celui issu d'une formation traditionnelle à la POO.

Le but de cet article est de vous présenter des cas de code concrets où ces principes ont permis de supprimer des adhérences trop fortes dans le code, d'augmenter la maintenabilité ou de découper des classes trop grosses dont la maintenance devenait problématique.

DESCRIPTION DE L'APPLICATION CIBLE

Les lecteurs ont probablement déjà lu une description rapide des principes SOLID, dans un article de blog ou sur toute autre source. Malheureusement, les exemples accompagnant ces descriptions sont souvent basés sur des cas purement académiques. Bien que suffisants pour la compréhension, l'auteur pense que ceux-ci ne permettent souvent pas l'acquisition réelle de ces outils intellectuels. Cet article cherche à améliorer la prise en main des principes SOLID en les montrant dans le contexte d'un cas concret, issu de l'activité industrielle d'un éditeur de logiciel.

Pour cela, il nous faut contextualiser les exemples en expliquant un concept d'ETL XML, que nous appellerons pipeline pour le reste de l'article. Ce pipeline de données XML est écrit en POO, avec un objet pour chacune des actions d'extraction ou de transformation, et un objet Pipeline permettant la manipulation globale [Fig.1].

La première action se charge de récupérer des données en provenance de flux quelconques, puis les actions suivantes transforment au fur et à mesure une charge utile au format XML, jusqu'à livraison du contenu final à l'appelant.

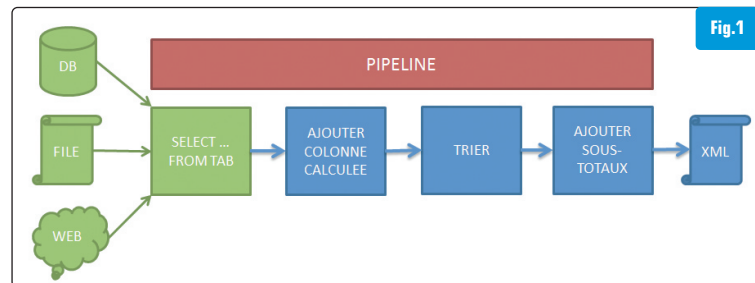


Fig.1

A (très) gros traits, le code de la classe Pipeline est comme suit :

```

public class Pipeline
{
    private List<IAction> _Actions;

    public void AjouterAction(IAction action)
    {
    }

    public XmlDocument Executer()
    {
        XmlDocument Dom = new XmlDocument();
        foreach (IAction action in _Actions)
            action.Executer(Dom);
    }
}
  
```

Quant aux actions, elles implémentent toutes l'interface IAction :

```

interface IAction
{
    void Executer(XmlDocument Dom);
}
  
```

Le contexte étant posé, nous allons voir comment les principes SOLID ont permis d'améliorer le code orienté objet du pipeline.

Note : pour des raisons de place, tous les exemples dans cet article sont plus proches du pseudo-code que d'un code réellement fonctionnel. Par contre, les définitions d'interfaces, de classes, et globalement tout ce qui est important pour la compréhension, a été repris tel quel du code en production.

PRINCIPE DE RESPONSABILITÉ UNIQUE

Il s'agit peut-être du plus important de ces cinq principes, ou en tout cas un de ceux qui sert le plus souvent lorsqu'on doit prendre une décision sur un remaniement de classes. Pour montrer un mode courant d'utilisation de ce principe, il nous faut expliquer que le pipeline étant en grande majorité utilisé avec des bases de données, une notion d'optimisation SQL a progressivement été mise en place, permettant de redescendre certaines activités dans la première action.

La requête SQL initiale est alors modifiée, et les actions correspondantes désactivées temporairement : [Fig.2].

Au passage, certaines actions comme le tri ne sont pas toujours plus rapides en SQL qu'en XSL/T, et un niveau d'adaptation au contexte (taille des données, présence d'index, etc.) est donc nécessaire, même si cela ne nous concerne pas dans le cas présent.

Pour mettre en place cette optimisation, le code de la classe Pipeline a été modifié pour lancer un traitement d'optimisation juste avant l'exécution. Comme le pipeline devait se rappeler, lors d'une seconde exécution, de ne pas relancer le processus d'optimisation à moins que d'autres actions aient été ajoutées, le code avait été placé sous sa responsabilité :

```
public class Pipeline
{
    // ...

    public XmlDocument Executer()
    {
        Optimiser();
        XmlDocument Dom = new XmlDocument();
        foreach (IAction action in _Actions)
            action.Executer(Dom);
    }

    private void Optimiser()
    {
        if (!OptimisationNecessaire) return;
        foreach (IAction action in _Actions)
        {
            if (action is ActionTri)
                _Actions[0].Requete += « ORDER BY ...»;
            else if (action is ActionCalcul)
                // ...
        }
    }
}
```

Après quelques mois durant lesquels de plus en plus d'actions ont été intégrées dans le processus d'optimisation, ce qui devait arriver arriva, et NDepend remonta une alerte sur la taille de la fonction Optimiser. Le mode de refactorisation apparut de manière claire grâce au principe de responsabilité unique. Une responsabilité est définie comme « une raison pour changer ». Si nous considérons qu'une raison de modifier ce code pouvait être la prise en compte d'un paramètre supplémentaire sur une action, il est logique qu'il ne fasse pas partie de l'objet Pipeline, mais de celui correspondant à l'action. La responsabilité devait être unique pour chaque objet :

- Le pipeline a la responsabilité de ne lancer le processus d'optimisation que lorsque c'est nécessaire (première exécution ou modification des actions), et de n'exécuter que les actions qui n'ont pas été remplacées par un enrichissement de la requête SQL.
- Chaque action a la responsabilité de la correspondance entre son

traitement d'exécution en mémoire et la portion de code SQL qu'elle émet lors de l'optimisation.

Le remaniement a donc consisté en tout premier en la création d'une interface contractuelle pour les actions optimisables en SQL :

```
interface IOptimisable
{
    bool ModifierSQL(ArborescenceSQL Requete);
}
```

Une notion d'arborescence SQL a été apportée pour faciliter la modification à chaque action, qui n'a ainsi pas à systématiquement décomposer le contenu pour savoir où porter son ajout.

Une action comme celle en charge du tri implémente alors, en plus d'IAction, cette nouvelle interface, et elle est seule responsable de la correspondance exacte entre les deux traitements :

```
class ActionTri : IAction, IOptimisable
{
    public void Executer(XmlDocument Dom)
    {
        // ...
    }

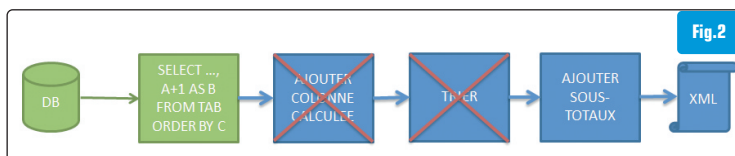
    public bool ModifierSQL(ArborescenceSQL Requete)
    {
        Requete.Order += «, » + _Variable + «ASC»;
    }
}
```

Quant au code du pipeline, il peut désormais se contenter de ses responsabilités propres, à savoir le test de l'utilité de l'optimisation SQL et, le cas échéant, le dispatch de la demande sur chacune des actions qui la supporte :

```
public class Pipeline
{
    // ...

    public XmlDocument Executer()
    {
        if (OptimisationNecessaire) Optimiser();
        // ...
    }

    private void Optimiser()
    {
        foreach (IAction action in _Actions)
        {
            IOptimisable optimisable = action as IOptimisable;
            if (optimisable != null)
                if (optimisable.ModifierSQL(_ArborescenceRequete))
                    action.Desactiver();
        }
    }
}
```



En appliquant le principe de responsabilité unique et en cherchant la responsabilité de chaque objet, nous avons obtenu un remaniement créant une dizaine de fonctions de quelques dizaines de lignes chacune, au lieu d'un énorme bloc de plusieurs centaines de lignes,

extrêmement difficile à maintenir et à faire évoluer. Ce remaniement s'est d'ailleurs avéré crucial pour l'évolution du composant dans les quelques mois qui ont suivi.

PRINCIPE OUVERT / FERMÉ

L'apparent paradoxe porté par ce principe s'explique par le fait que deux demandes contradictoires sont faites au code orienté objet : il doit être évolutif, mais également robuste. La robustesse plaide pour l'encapsulation, en fermant soigneusement tout objet aux modifications qui lui seraient portées depuis l'extérieur (à l'extrême, les objets immutables offrent la garantie qu'ils ne peuvent être modifiés une fois créés, ce qui limite sérieusement le nombre de bogues). Mais le besoin d'évolutivité fait que ces mêmes classes d'objet doivent pouvoir voir leur comportement spécialisé, et en un mot, qu'elles doivent être ouvertes. Il faut pouvoir étendre leur fonctionnement sans avoir à les recompiler ou, encore pire, compiler leur code dupliqué dans d'autres classes. Le principe Ouvert / Fermé a le mérite de poser le problème en termes de domaines d'application, de façon à résoudre le paradoxe : une classe doit être ouverte à l'extension, mais fermée à la modification. Ainsi, la fermeture doit être de mise pour tous les champs et les propriétés en écriture. A l'inverse, les méthodes doivent être ouvertes. L'encapsulation est une notion bien connue de la POO, et nous ne montrerons ici qu'un exemple réaliste de son utilité. Supposons qu'une action `ActionAgregation` s'optimise en SQL en transformant les champs en `SUM(champ)`, et que nous lui laissons l'accès à la formule dans `ActionCalcul`. Rien ne l'empêcherait de modifier la formule en ajoutant `SUM()` autour de celle-ci. Or, comme ce n'est pas sa responsabilité, elle ne saurait pas qu'il y a une différence entre `SUM(a/b)` et `SUM(a)/SUM(b)`, ni dans quel cas il faut utiliser l'une ou l'autre formulation. Il est donc essentiel d'encapsuler le champ `Formule` et de ne pas laisser un accès en écriture être réalisé sur son contenu. L'ouverture à l'extension nécessite de prendre un exemple différent, à savoir la spécialisation d'une action qui s'occupe de réorganiser les colonnes en une action qui permet d'en supprimer une. Cette spécialisation se fait très souvent par héritage, même si d'autres moyens sont possibles. Ceci nécessite quelques conditions sur le code :

- L'action mère ne doit pas être marquée *sealed*.
- Elle doit être marquée *public* si l'héritage peut être externe.
- Les méthodes doivent être marquées *virtual*.
- Le champ contenant la liste des colonnes restantes doit être *protected*.

Au final, le principe ouvert / fermé amène au code ci-dessous pour la classe-mère :

```
public class ActionOrdreColonnes : IAction
{
    protected List<string> _Colonnes;

    public virtual void Executer(XmlDocument Dom)
    {
        // ...
    }
}
```

Celle-ci peut alors être simplement héritée pour spécialiser son comportement :

```
class ActionSupprimerColonne : ActionOrdreColonnes, IAction
{
```

```
    private string _ColonneASupprimer;

    public override void Executer(XmlDocument Dom)
    {
        _Colonnes = _Colonnes.Remove(_ColonneASupprimer);
        base.Executer(Dom);
    }
}
```

Bien entendu, dans des conditions où la spécialisation peut poser problème, le curseur ouvert / fermé peut se déplacer et amener par exemple à déclarer certaines classes *sealed*, de façon à empêcher leur héritage.

PRINCIPE DE SUBSTITUTION DE LISKOV

Continuons sur cet exemple d'héritage pour aborder le principe de substitution de Liskov. Si une classe quelconque nécessite une instance d'`ActionOrdreColonnes`, nous pouvons bien sûr lui passer une instance d'`ActionSupprimerColonne`, car la seconde hérite de la première et donc la spécialise.

Ceci est assuré par le compilateur C#, qui valide la cohérence des types. Depuis .NET 4.0, la covariance et la contravariance sont d'ailleurs complètes, avec une prise en charge y compris dans les délégués. Mais ceci n'est que la face émergée de l'iceberg, et le principe de substitution va plus loin. En effet, supposons maintenant qu'une règle interne de `ActionOrdreColonnes` empêchait qu'on n'ait aucune colonne à traiter, bref que la liste `_Colonnes` ne puisse pas être vide. La classe qui utilisait `ActionOrdreColonnes` savait qu'elle pouvait en toute sécurité utiliser la première colonne de données :

```
class Utilisateur
{
    public void Traiter(ActionOrdreColonnes ordonnanceur)
    {
        XmlDocument Dom;
        ordonnanceur.Executer(Dom);
        var t = Dom.SelectSingleNode("TABLE0").SelectSingleNode("*").InnerText;
    }
}
```

Que se passe-t-il si `ActionSupprimerColonne` ne respecte pas cette règle, et permet de supprimer la totalité des colonnes présentes ? La classe utilisatrice émettra une exception non prévue, ce qui est bien évidemment un problème. Afin d'éviter ce genre de souci, le principe de substitution de Liskov prône que tous les comportements d'une classe héritée se retrouvent dans ceux de la classe qui les hérite. Ceci nécessite en particulier que les invariants, comme la règle de notre exemple, doivent être préservés.

Mais le principe va plus loin, et doit être appliqué à toutes les règles tacites de la classe. Par exemple, si on utilise une programmation par contrat et que des pré-conditions sont mises en place sur les paramètres entrants, celles-ci ne peuvent pas être renforcées par l'héritage. Supposons qu'une pré-condition sur l'appel d'une fonction exige un entier entre 1 et 10. L'héritage ne peut pas se permettre de n'accepter que des nombres entre 6 et 10, sinon il y aura rupture du contrat tacite de fonctionnement. La règle s'applique bien sûr également sur les post-conditions, mais en sens inverse, à savoir que celles-ci ne peuvent être affaiblies par la spécialisation.

PRINCIPE DE SÉPARATION DES INTERFACES

Ce principe pose qu'il est préférable de mettre en place plusieurs interfaces plutôt qu'une seule sauf lorsque les méthodes sont effectivement liées de manière indéniable par le métier qu'elle expose. On peut par exemple imaginer l'interface ci-dessous pour interroger une base de données sur sa structure :

```
interface InfoBase
{
    ICollection<string> GetListeTables();

    ICollection<string> GetListeChampsPourTable(string Table);
}
```

Il est alors irréfutable que les deux fonctions doivent travailler ensemble, car le retour de la première donne les valeurs possibles pour le paramètre d'entrée de la seconde. Mais dans une grande majorité des cas, les méthodes peuvent, et doivent, être séparées. Revenons sur l'exemple d'optimisation des actions du pipeline. A ce moment du code, nous avons introduit une interface IOptimisable :

```
interface IOptimisable
{
    bool ModifierSQL(ArborescenceSQL Requete);
}
```

Il aurait bien sûr été envisageable d'inclure la méthode ModifierSQL dans IAction elle-même. Nous aurions tout à fait pu partir du principe que toute action était optimisable a priori, et qu'il suffisait de ne rien faire dans l'implémentation si ce n'était pas le cas. Nous avons d'ailleurs même à notre disposition un retour booléen qui avait été spécialement pensé pour prévenir en retour le pipeline que l'action pouvait être outrepassée à l'exécution ou pas. Dans le cas où l'action n'était pas optimisable, nous aurions simplement écrit :

```
class ActionNonOptimisable : IAction
{
    public void Executer(XmlDocument Dom)
    {
        // ...
    }

    public bool ModifierSQL(ArborescenceSQL Requete)
    {
        return false;
    }
}
```

Toutefois, le principe de séparation des interfaces agit comme le principe de responsabilité simple sur celles-ci, et nous commande dans ce cas de réaliser deux interfaces séparées, car leur responsabilité du point de vue du métier n'est tout simplement pas la même. Pour bien comprendre le problème de maintenance posé par l'interface multiple, il nous faut préciser qu'avant l'utilisation de la classe ArborescenceSQL, la requête à manipuler était passée sous forme de chaîne :

```
interface IOptimisable
{
    bool ModifierSQL(string Requete);
}
```

Supposons que nous ayons ouvert l'interface suivante à des usages externes :

```
public interface IAction
{
    void Executer(XmlDocument Dom);

    bool ModifierSQL(string Requete);
}
```

Après tout, il est tout à fait possible d'envisager des actions créées par d'autres éditeurs de façon à être intégrées dans le pipeline à l'exécution. Surtout, tous les types dans les signatures étaient simples et connus (string, XmlDocument, bool). Le problème serait alors advenu lorsque nous aurions introduit le changement utilisant ArborescenceSQL au lieu de string pour la manipulation de la requête : toutes les implémentations extérieures auraient été cassées, alors que presque toutes ne se servaient vraiment que de la première méthode, et renvoyaient simplement false pour la seconde. Il y a donc bien une différence d'utilisation des méthodes, qui justifie que deux interfaces contractuelles soient réalisées. Ce n'est d'ailleurs pas pour rien que la première, IAction, est marquée *public*, alors que la seconde, IOptimisable, a finalement été positionnée en *internal*.

Note : nous pourrions aussi faire hériter IOptimisable de IAction, mais ceci est encore un sujet en soi, et nous ne le traiterons pas ici.

PRINCIPE D'INVERSION DE DÉPENDANCE

L'exemple du pipeline, issu d'un cas d'utilisation réel, va nous permettre d'aborder la notion d'inversion de dépendance par un biais différent de la plupart des autres explications.

En règle générale, la notion d'inversion de dépendance est expliquée par un raisonnement contractuel : une classe n'a pas besoin d'une implémentation donnée d'un service, mais bien de la simple possibilité de réaliser ce service. C'est le principe même du contrat : supposons qu'une action de pipeline envoie des mails en masse : ce qui est important est qu'elle dispose d'une fonction prenant une adresse, un sujet, un contenu et réalisant l'envoi. Ce n'est pas de savoir directement que telle ou telle API en est capable.

Pourtant, en pratique, l'inversion de dépendance apparaît souvent comme la conséquence d'une mise en commun de fonctionnalité. Supposons en effet que plusieurs actions aient besoin d'envoyer des mails. Après avoir constaté que deux actions développées par des équipes séparées ont divergé, l'une d'entre elles utilisant l'API Mail et l'autre l'API Outlook, une mise en commun sera opérée par refactorisation, et quoi de plus simple que de créer une interface :

```
interface IGestionMail
{
    void EnvoyerMail(string Message, string Destinataire, string Objet);
}
```

Une des implémentations pourra être celle-ci :

```
class GestionMailOutlook : IGestionMail
{
    public void EnvoyerMail(string Message, string Destinataire, string Objet)
    {
        Microsoft.Outlook.Mail.Send();
    }
}
```

```
}
}
```

Le sens des dépendances étaient auparavant comme suit : [Fig.3].

Une fois l'interface mise en place, on constate l'inversion du sens de la dépendance des modules de gestion vers la couche de gestion des pipelines et des actions. Ils ont alors en commun une dépendance à IGestionMail, définie dans l'API des pipelines : [Fig.4].

Bien sûr, il faut alors pousser jusqu'à la mise en commun de l'instance de gestionnaire de mail dans le pipeline, qui distribuera cette instance aux actions pour un besoin éventuel :

```
public class Pipeline
{
    // ...

    private IGestionMail _GestionMail = null;

    public Pipeline(IGestionMail GestionMail)
    {
        _GestionMail = GestionMail;
    }

    public XmlDocument Executer()
    {
        // ...
        foreach (IAction action in _Actions)
        {
            action.GestionMail = _GestionMail;
            action.Executer(Dom);
        }
    }
    // ...
}
```

Et c'est bien un appelant externe, qui aura à la fois la dépendance sur les pipelines et la dépendance sur les gestionnaires de mail (d'ailleurs partageables avec d'autres utilisations, du coup), qui écrira :

```
class Utilisateur
{
    public void Lancer()
    {
        Pipeline p = new Pipeline(new GestionMailOutlook());
    }
}
```

Fig.3

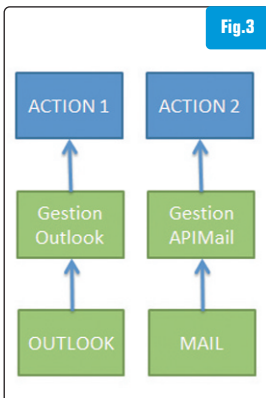
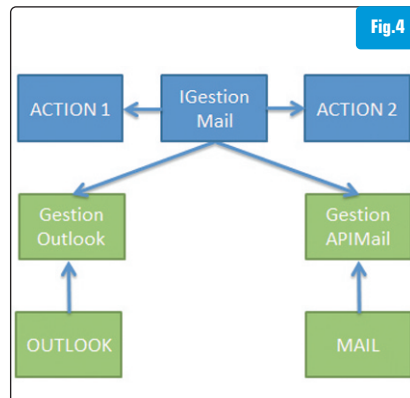


Fig.4



```
p.Executer();
}
}
```

L'inversion de dépendances est le principe le plus « technique » des cinq, dans le sens où il peut être pris simplement comme l'application d'une méthode permettant d'obtenir le résultat voulu, alors que les autres procèdent d'une approche moins tranchée. Cette relative facilité de mise en place compense le fait que c'est un principe qui contrevient à l'ordonnancement « instinctif » des références (d'où son nom de principe d'« inversion »). Ce genre de code appelle bien sûr à mettre en place un véritable système d'injection, qui permet d'affecter une instance d'un gestionnaire de mail particulier à une propriété d'une classe donnée, et ce en passant simplement par un fichier XML contenant des règles d'affectation. Ce type d'outils permet par exemple de garder une instance unique pour une affectation multiple, ou d'affecter une instance de tel ou tel type en fonction de la valeur d'une propriété, ou du type de la cible, etc. Il découle tout naturellement de la mise en place de l'inversion de dépendance. Nous renvoyons le lecteur à la documentation de Managed Extensibility Framework de Microsoft pour plus de détails sur l'injection en .NET 4.0.

CONCLUSION

Plus encore que des principes de codage, la compréhension des cinq principes SOLID est un atout pour la gestion du refactoring : lorsqu'on remanie du code, il est important de prendre des décisions averties sur la façon de découper ou de regrouper des ensembles de code : les principes SOLID sont l'outil numéro un du développeur dans ces situations.

Nous avons vu quelques exemples d'utilisation des principes SOLID sur un cas réel, mais nous pourrions les multiplier à l'infini. Que faire si on doit envoyer des mails en masse, et plus seulement sur une seule adresse ? Le principe de séparation des interfaces nous propose de ne pas compléter IGestionMail, mais plutôt de rajouter une IGestionMailMasse, car le besoin métier est fondamentalement différent, et pourra éventuellement être rempli par deux instances injectées depuis des types différents.

Le principe de responsabilité unique commandera également de mettre en place une interface pour les actions ayant besoin d'un gestionnaire de mail. Le principe de Liskov nous mettra en garde contre une spécialisation d'un gestionnaire vérifiant les adresses mails alors qu'un autre ne le faisait pas.

Avec un peu d'entraînement, un développeur POO jongle à longueur de temps avec ces principes, qui lui donnent des réponses applicables rapidement à ses questionnements. C'est l'idée même des principes : au moment où on les applique, on ne sait pas nécessairement précisément en quoi ils nous aident. On se rend compte lorsqu'on oublie de les appliquer... et qu'on passe plus du temps à remanier le code qu'à lui apporter des fonctionnalités nouvelles !

Références

<http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod>
<http://msdn.microsoft.com/en-us/magazine/cc546578.aspx>
<http://c2.com/cgi/wiki>



Jean-Philippe Gouigoux

(MVP, MCPD.NET)

Responsable du pôle Architecture / Formation / Innovation de la société MGDIS. Il est auteur sur la programmation .NET efficace pour les éditions ENI et Simple Talk – <http://gouigoux.com/blog-fr/>

Créer une liste téléphonique avec SQL Server Reporting Services

1^{re} partie

Le moteur de rapport de SQL Server 2008 R2 est capable de fournir de très nombreux services utilisables au sein d'une entreprise. Un exemple de base d'utilisation est la fameuse liste téléphonique utilisée par de très nombreuses personnes telles que les assistantes de direction ou les chargés d'accueil téléphonique.

Nous partons d'un exemple simple basé sur une liste de type Contact stockée dans SharePoint 2010 [Fig.1]. Nous verrons comment charger les informations de base utiles dans une liste téléphonique (Nom, Prénom et Téléphone du bureau). Il faut que ces informations soient chargées dans la liste source.

> Connexion à la source de données

La première étape est de connecter l'outil de développement de rapport « SQL Server Business Intelligence Development Studio ». Il faut donc créer un nouveau projet de type « Report Server Project » [Fig.2].

Nous devons maintenant créer une nouvelle chaîne de connexion, pour ceci, nous devons cliquer avec le bouton droit sur « Shared Data Sources » et choisir « Add New Data Source » [Fig.3]. Il faut choisir le type Microsoft SharePoint List, spécifier l'URL du site SharePoint où la liste est stockée [Fig.4].

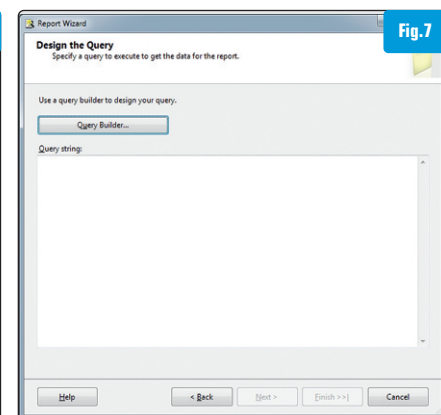
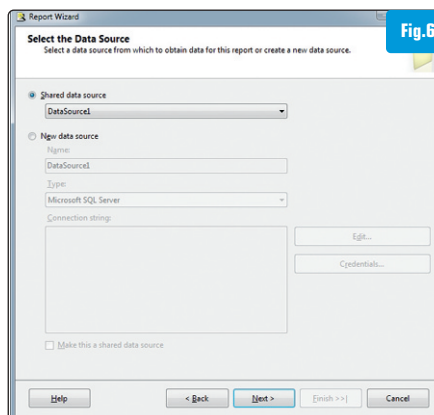
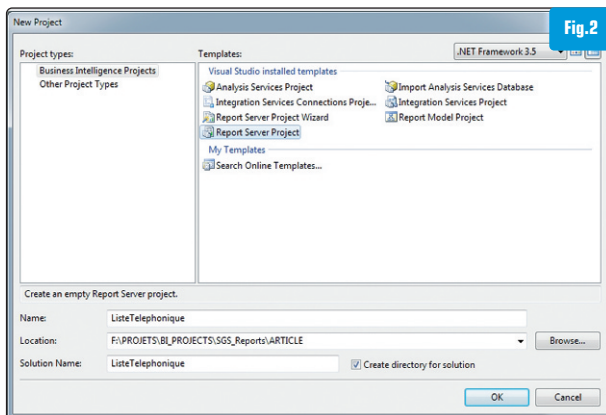
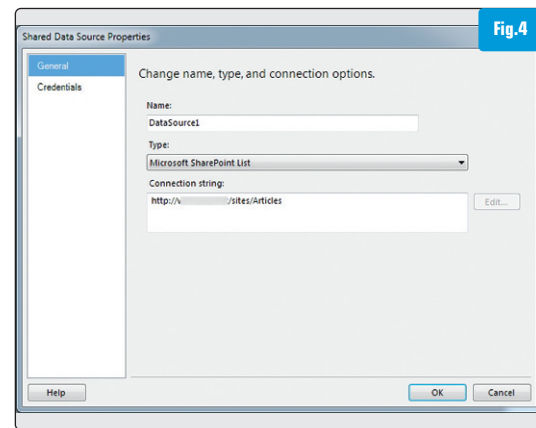
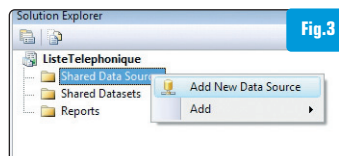
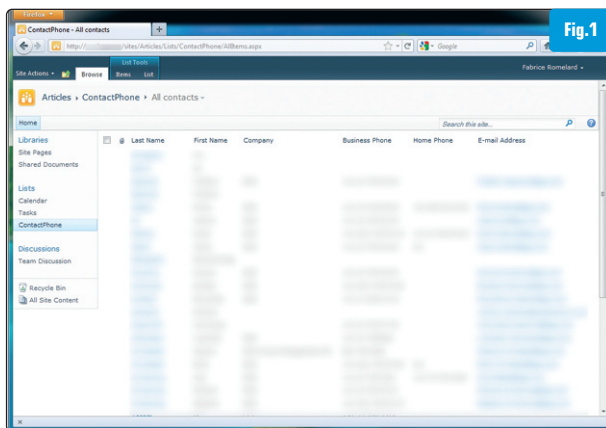
Il faut cocher « Use Windows Authentication (integrated security) » dans la fenêtre d'options « Credentials ». Notre chaîne de connexion est utilisable dans un rapport.

> Création du rapport

Pour créer ce rapport, il faut cliquer avec le bouton droit sur « Reports » et choisir « Add New Report » [Fig.5]. Un assistant de création apparaît, il faut cliquer sur Next et choisir la source de données créée précédemment « DataSource1 » [Fig.6], puis cliquer sur « Next ». Il faut cliquer sur « Query Builder » pour créer la requête CAML adaptée pour notre besoin [Fig.7].

L'assistant nous propose l'ensemble des listes disponibles. Il faut cocher la liste qui nous intéresse et les colonnes de cette liste, il est possible de tester cette sélection en cliquant sur « Run Query » [Fig.8], puis il faut cliquer sur OK pour valider. L'assistant nous fournit la requête exécutée :

```
<RSSharePointList xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ListName>ContactPhone</ListName>
  <ViewFields>
    <FieldRef Name="Title" />
    <FieldRef Name="FirstName" />
  </ViewFields>
</RSSharePointList>
```



```
<FieldRef Name="WorkPhone" />
</ViewFields>
</RSSharePointList>
```

Il faut cliquer sur « Next » pour passer à l'étape suivante, afin de choisir « Tabular » et valider par « Next ». A ce stade, il faut ajouter les trois champs (Nom, Prénom et Business Phone) dans la partie détails [Fig.9], puis valider par « Next ». Il faut choisir le Style du tableau « Corporate » et valider par « Next ». Enfin, il faut donner un nom à ce rapport « Report_ListeTelephonique » et valider par « Finish » pour obtenir notre rapport [Fig.10].

> Modification du rapport

La première modification à effectuer est d'ajouter un filtrage sur les numéros vides et un classement sur la requête envoyée au serveur SharePoint, ceci par l'option <Query>

```
<RSSharePointList xmlns:xsi="http://www.w3.org/2001/XMLSchema
-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ListName>ContactPhone</ListName>
  <ViewFields>
```

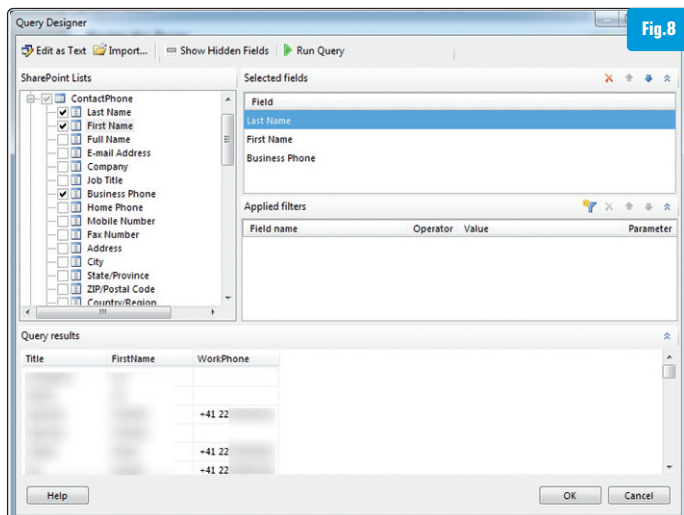


Fig.8

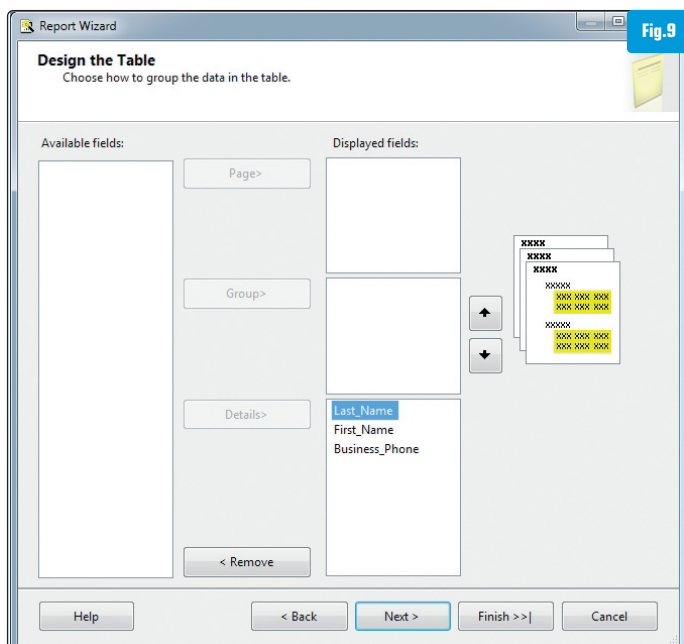


Fig.9

```
<FieldRef Name="Title" />
<FieldRef Name="FirstName" />
<FieldRef Name="WorkPhone" />
</ViewFields>
<Query>
  <Where>
    <And>
      <IsNotNull>
        <FieldRef Name="WorkPhone" />
      </IsNotNull>
      <Neg>
        <FieldRef Name="WorkPhone" />
        <Value Type="Text" />
      </Neg>
    </And>
  </Where>
  <OrderBy>
    <FieldRef Name="Title" Ascending="true" />
    <FieldRef Name="FirstName" Ascending="true" />
  </OrderBy>
</Query>
</RSSharePointList>
```

La seconde modification est d'ajouter un champ calculé pour la première lettre du nom de famille. Ceci se fait en cliquant avec le bouton droit sur le DataSet et choisir « Add Calculated Field » [Fig.11]. Ce champ contiendra la formule « =Left[Fields!Last_Name.Value,1] ».

Fabrice Romelard [MVP]

Intranet - Extranet Chief Technical Officer - SGS

<http://blogs.developpeur.org/fabrice69>

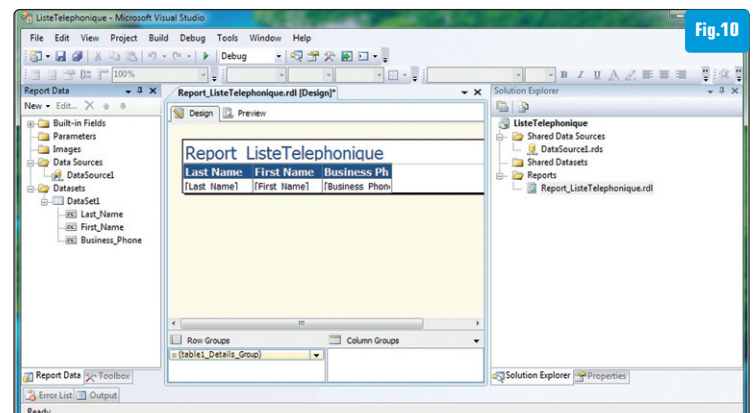


Fig.10

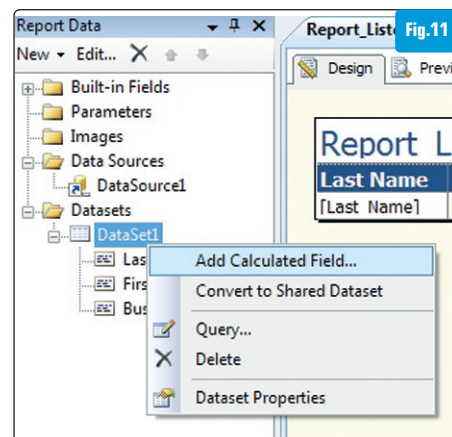


Fig.11

Guava : la boîte à outils Java by Google

Quel que soit le projet de développement logiciel, on retrouve un grand nombre de portions de code récurrentes répondant à des cas d'usages allant de la manipulation des chaînes à la gestion des collections. De fait, les développeurs sont souvent amenés à réécrire du code identique, ce qui s'avère coûteux et peut nuire à la qualité de certaines implémentations. Pour adresser cette problématique, Google met à disposition Guava, sa propre boîte à outils Java, qui apporte des réponses simples et pratiques à ces cas d'usages. Cet article propose un tour d'horizon de ses fonctionnalités les plus marquantes.

Présente au cœur de l'ensemble des projets Java version 5 ou supérieure de Google, Guava est librement mise à disposition depuis fin 2009. La question de sa pérennité ne se pose donc même pas. Souvent comparée aux boîtes à outils Commons de la fondation Apache, Guava se démarque par le fait qu'elle cible Java 5 comme version minimum afin de bénéficier au mieux des nouveautés apportées par cette mouture et plus particulièrement des génériques. En outre, elle s'appuie sur les meilleures pratiques du domaine en termes de design, avec notamment la contribution remarquable d'un certain Joshua Bloch, bien connu de la communauté Java. Concrètement, Guava fournit de nombreuses fonctionnalités qui idéalement auraient dû se trouver en standard au sein du JDK. Malheureusement, elles n'y sont pas à l'heure actuelle et au rythme où s'effectuent les montées de versions du JDK sur les projets d'entreprises, il faudra sans doute attendre plusieurs années avant de bénéficier de celles qui seront ajoutées.

Partant de ce constat, l'approche prônée par les équipes de Guava se révèle pragmatique. Elle consiste à conseiller d'utiliser en priorité les fonctionnalités présentes en standard dans le JDK et ensuite seulement de passer par Guava pour les besoins non couverts. Guava est donc bien à considérer comme un complément aux manques du JDK.

La librairie fournit un ensemble de classes et de méthodes utilitaires répondant à des usages très fréquents au sein des applications, allant de la manipulation des chaînes de caractères à la gestion des collections en passant par des opérations récurrentes sur les beans. Ainsi, Guava vise avant tout à simplifier la vie du développeur Java au quotidien.

Certains s'interrogeront peut-être sur la pertinence d'intégrer une dépendance supplémentaire au sein de leur projet pour des classes qu'il est possible de développer soi-même. Alors pourquoi utiliser plutôt Guava ? Tout simplement pour gagner du temps et bénéficier des fonctionnalités avancées d'une librairie éprouvée par la communauté et suivant les meilleures pratiques du monde Java, ce qui assure une qualité optimale.

La dernière limitation pointée du doigt concerne la taille de la librairie puisque sa dernière version, la 11, sortie fin 2011 tourne aux alentours de 1,6 Mo et se présente sous la forme d'un bloc monolithique constitué d'un unique JAR. Un découpage plus modulaire, à l'image de ce que propose Spring, en différents artefacts correspondant aux grands domaines adressés par la librairie, aurait sûrement été plus judicieux, avec la possibilité de récupérer le bloc complet comme cela est fait actuellement.

> Manipulation des chaînes de caractères

Guava se compose d'une douzaine de packages rassemblés sous `com.google.common`. Cœur de Guava, le package base rassemble tout ce qui a trait au package `java.lang` du JDK. On retrouve ainsi tout ce qui facilite la manipulation des chaînes de caractères. En ce sens, `Joiner` propose une API fluide facilitant la transformation d'un ensemble d'objets en une chaîne de caractères, ce qui s'avère fort utile pour la génération de données au format CSV. La transformation en `String` d'une liste de `String` contenant également des `nulls` s'effectue comme suit :

```
Joiner joiner = Joiner.on(";").skipNulls();
List<String> names = Arrays.asList("Sylvain", null, "Rachida", null);
out.println(joiner.join(names)); // affiche Sylvain;Rachida
```

Le séparateur associé au `Joiner` est défini à la construction via la méthode `factory on` qui renvoie un objet immuable que nous configurons avec `skipNulls` pour spécifier que les valeurs nulles doivent être omises. La méthode `useForNull` est à retenir car elle permet de définir une `String` qui remplacera les valeurs nulles lors du join. En outre, `Joiner` propose un `MapJoiner` opérationnel sur des maps. L'instanciation est identique mais la configuration requiert l'appel à `withKeyValueSeparator` :

```
Joiner.MapJoiner mapJoiner = Joiner.on(", ").withKeyValueSeparator("=");
out.println(mapJoiner.join(map)); // affiche key1=val1, key2=val2 ...
```

À l'opposé, la classe `Splitter` permet le découpage d'une chaîne pour en extraire une collection de valeurs, ce qui facilite le parsing de données CSV. `Splitter` utilise soit un délimiteur fixe de type `String`, soit un pattern de type regex. Sa configuration permet ensuite de supprimer les blancs autour des valeurs via `trimResults` ou d'ignorer les valeurs vides via `omitEmptyStrings` :

```
Splitter splitter = Splitter.on(";").trimResults().omitEmptyStrings();
Iterable<String> names = splitter.split("Sylvain;Rachida; ;Claude");
```

Autre classe indispensable, `CharMatcher` permet la mise en place de solutions de pattern matching lisibles là où l'utilisation de `Regex` s'avère complexe. Offrant un large éventail de possibilités, `CharMatcher` peut être utilisé pour récupérer les caractères respectant des contraintes au sein d'une chaîne :


```
CharMatcher matcher = CharMatcher.DIGIT.or(CharMatcher.is('/'));
String output = matcher.retainFrom("Date : 02/07/1922"); //
contient 02/07/1922
```

L'expressivité du CharMatcher peut être mise à profit dans la validation d'un login utilisateur ne devant pas contenir de caractères spéciaux par exemple :

```
CharMatcher noSpecialChars = CharMatcher.noneOf("!@?({}%").
negate();
boolean isValid = noSpecialChars.matchesNoneOf(login);
```

Les méthodes *or*, *and* et *negate* du CharMatcher permettent de réaliser des opérations booléennes sur la contrainte définie.

> Méthodes utilitaires pour POJO

La classe Objects fournit 3 méthodes utiles au quotidien pour les classes du modèle objet d'une application. Désormais intégrées au sein de Java 7, les méthodes *equal* et *hashCode* demeurent utiles pour les versions antérieures de Java. En revanche, la méthode Objects.toStringHelper, qui facilite l'implémentation de la méthode toString d'un POJO, n'a pas d'équivalent et se révèle indispensable durant le debug. Sur un bean User, l'implémentation de ces méthodes est réalisée comme suit :

```
public class User {
    private long id;
    private String firstName;
    private String lastName;
    private String login;
    private String email;
    private Date birth;

    // constructeur, getters, setters ...

    public int hashCode() {
        return Objects.hashCode(id, firstName, lastName, login, birth,
email);
    }

    public boolean equals(Object o) {
        if (o == null) {
            return false;
        }

        if (o instanceof User) {
            User other = (User) o;
            return Objects.equal(other.getFirstName(), firstName) &&
                Objects.equal(other.getLastName(), lastName) &&
                Objects.equal(other.getEmail(), email) &&
                Objects.equal(other.getLogin(), login) &&
                Objects.equal(other.getBirth(), birth);
        }

        return false;
    }

    public String toString() {
```

```
        return Objects.toStringHelper(this).add("Id", id).
            add("First name", firstName).
            add("Last name", lastName).
            add("Login", login).
            add("Email", Strings.nullToEmpty(email)).
            toString();
    }
}
```

Outre ces méthodes, Guava facilite la comparaison d'objets via ComparisonChain qui permet de s'abstraire de la mécanique liée à l'interface Comparable ou au Comparator du JDK. La création de ComparisonChain se fait via la méthode de classe start et les champs à comparer sont ensuite ajoutés via appel à compare. La comparaison est lancée grâce à result et est paresseuse puisqu'à la première différence entre 2 champs elle est interrompue et un résultat différent de 0 est renvoyé.

> Programmation par contrat

À l'instar des Apache Commons, Guava possède des méthodes statiques permettant de spécifier des pré-conditions à l'exécution. Regroupées au sein de Preconditions du package base, chacune d'entre elles compte 3 variantes. La première sans argument autre que la condition à vérifier, la seconde rajoutant un message affiché en cas de non satisfaction de la condition et enfin une dernière proposant un remplacement des paramètres au sein du message d'erreur défini. La méthode checkNotNull est une des plus utiles. Elle vérifie si l'argument passé en entrée est non nul et lance une NullPointerException le cas échéant. La méthode checkArgument valide quant à elle un paramètre via une expression booléenne et lance un IllegalArgumentException si besoin. Appliquée à notre bean User, la validation des paramètres du constructeur et des setters est réalisée comme suit :

```
public User(int id, String firstName, String lastName, String login) {
    checkArgument(id >= 0, "Id must be positive : %s", id);
    this.id = id;
    this.firstName = checkNotNull(firstName, "First name must not
be null");
    this.lastName = checkNotNull(lastName);
    this.login = checkNotNull(login);
}

public void setFirstName(String firstName) {
    this.firstName = checkNotNull(firstName);
}
```

Les méthodes de Guava permettant de mettre en place une sorte de programmation par contrat ne sont pas révolutionnaires, mais elles ont l'avantage d'avoir un nom clair avec la possibilité de passer des paramètres dans les messages d'erreur associés. En outre, checkNotNull renvoie l'argument passé en entrée s'il est valide, ce qui réduit sensiblement la quantité de code à écrire et améliore la lisibilité.

> Collections

Mergée au sein de Guava et regroupée dans le package collect, l'ancienne API Google-Collections constitue une part importante de l'API. Guava tire parti de l'extensibilité de l'API Collections du JDK, rendue

possible par un design remarquable, pour adresser des problématiques bas-niveau non couvertes par l'API standard en y apportant des réponses fiables. A l'instar de tout ce qui est fait dans Guava, les collections sont instanciables via des factories statiques réduisant de fait la verbosité inhérente aux génériques. La création d'une map étant simplifiée de la sorte :

```
Map<String, Map<String, Integer>> map = Maps.newHashMap();
```

L'inférence de type réalisée sur la méthode `newHashMap` affranchit le développeur de la déclaration trop verbeuse à l'instanciation de la map. Le JDK 7 a réduit cette verbosité en introduisant l'opérateur diamond `<>`.

Outre ces factories de construction, Guava introduit de nouvelles collections répondant à des besoins fréquents mais non couverts par le JDK. Au premier rang desquelles on trouve `Multimap` qui modélise une map où plusieurs valeurs sont associées à une même clé. Elle remplace avantageusement les constructions de la forme `Map<K, Set<V>>` toujours difficiles à gérer. A la première insertion d'un couple clé-valeur, le `Set` sous-jacent est créé automatiquement. Les opérations d'ajout et de retrait s'effectuant comme avec une map classique. Les implémentations de `Multimap` ciblent soit un `Set` soit une `List` comme nous pouvons le voir ici :

```
ListMultimap<String, String> ingredients = ArrayListMultimap.create();
ingredients.put("Lasagnes", "Tomate");
ingredients.put("Lasagnes", "Viande");
List<String> lasagnes = ingredients.get("Lasagnes"); // contient
Tomate, Viande
```

Au rayon nouvelles collections, citons également le `MultiSet`, une collection non ordonnée autorisant les doublons, qui a la particularité de compter le nombre d'occurrences de chaque élément inséré. Son utilisation est recommandée lorsqu'il est nécessaire de collecter des statistiques sur la fréquence d'apparition d'éléments au sein d'un ensemble, comme pour compter le nombre d'apparitions d'un mot dans un texte. Semblable à une map classique, la `BiMap` garantit l'unicité des valeurs autorisant un parcours bidirectionnel de son contenu. Enfin, un mot sur la classe `Table` qui modélise une table de données sur laquelle il est possible de travailler en colonne ou en ligne de manière aisée. Guava permet de définir des contraintes sur les collections tant au niveau de listes et des sets qu'au niveau des maps respectivement via les classes `Constraints` et `MapConstraints` qui appliquent une contrainte sur une collection donnée. Les contraintes sont respectivement définies en implémentant les interfaces `Constraint` et `MapConstraint`. Etant donnée une map représentant un annuaire, nous définissons une contrainte d'ajout basée sur la validité du numéro de téléphone associé au contact afin de lancer une exception et de rejeter l'ajout si le numéro est incorrect [Fig.1] :

```
ListMultimap<String, String> annuaire = ArrayListMultimap.create();
annuaire = MapConstraints.constrainedListMultimap(annuaire, new
MapConstraint<String, String>() {
    public void checkKeyValue(String k, String v) {
        Preconditions.checkNotNull(v);
        String regex = "(\\d{2}){5}";
```

```
        if (!v.matches(regex)) {
            throw new IllegalArgumentException("Phone number invalid : " + v);
        }
    }
});

annuaire.put("Sylvain", "0604030201"); // OK
annuaire.put("Sylvain", "09"); // KO
```

D'autre part, on peut souligner l'approche poussée par Guava consistant à favoriser l'utilisation de collections immuables dès que possible afin de préserver le caractère thread-safe des programmes. Plutôt que de passer par `Collections.unmodifiableXXX` du JDK, Guava propose une implémentation de type `ImmutableXXX` pour la plupart des collections. Les constructions se faisant par des méthodes statiques telles que `copyOf` et `of` ou via un builder spécifique. Enfin, un mot sur la quantité importante de méthodes utilitaires indispensables aux collections proposées par Guava. Que ce soit au niveau des listes, des sets ou des maps, Guava regorge de méthodes faisant gagner un temps précieux au cours du développement et il est fortement conseillé aux développeurs de parcourir l'excellente documentation de l'API pour en tirer pleinement parti.

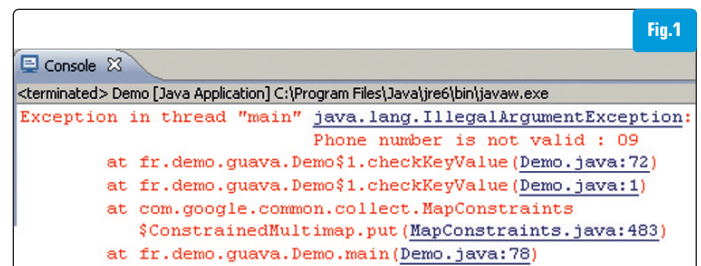
> Programmation fonctionnelle

Guava introduit certaines notions de programmation fonctionnelle. Si certains concepts de ce paradigme peuvent être à peu près reproduits en Java, ils conduisent bien souvent à un code plutôt verbeux, saupoudré de classes anonymes. De fait, leur utilisation via Guava doit être faite de manière pragmatique et seulement dans le cas où cela conduit à une meilleure lisibilité et à un gain en quantité de code. Guava propose 2 notions fondamentales du monde fonctionnel :

- Les fonctions modélisées par l'interface `Function<F,T>` qui prennent un argument de type `F` pour rendre un résultat de type `T` au sein d'une méthode `apply` à implémenter.
- Les prédicats modélisés par l'interface `Predicate<F>`, cas particulier des fonctions retournant uniquement un booléen.

Combinée avec les collections, l'utilisation de ces 2 types de fonctions permet d'effectuer de puissants traitements de transformation ou de filtrage rendant des vues immuables des collections correspondantes. En ajoutant une liste de rôles au bean `User`, récupérer la liste des admins devient chose aisée avec un prédicat :

```
Collection<User> admins = Collections2.filter(users, new Predicate
<User>() {
    public boolean apply(User user) {
        return user.hasRight("ADMIN");
    }
});
```



Erreur suite à l'ajout d'un numéro invalide

Ici, le prédicat est défini via une classe anonyme mais il présente l'avantage de pouvoir être créé séparément afin d'avoir une base de prédicats et de fonctions réutilisables. La classe *Iterables* offre des méthodes tirant parti des prédicats pour récupérer par exemple les éléments d'une collection en vérifiant un.

Les fonctions sont utilisées pour des transformations rapides comme par exemple transformer des objets en DTO équivalents avant de les passer aux couches supérieures. Toujours sur le bean User, obtenir une liste de String représentant les utilisateurs sous un format affichable est assez simple :

```
List<String> usersToDisplay = Lists.transform(users, new Function<User, String>() {
    public String apply(User user) {
        return user.getFirstName() + " " + user.getLastName() + " - " + user.getEmail();
    }
});
```

Malheureusement, Guava n'implémente pas pour le moment de fonctionnalité de type *reduce* qui permettrait d'obtenir un résultat de calcul à partir d'une fonction appliquée à une collection d'objets, ce

qui pourrait s'avérer pertinent par exemple pour calculer la somme des revenus d'un ensemble d'utilisateurs.

> Divers

Difficile de faire un tour exhaustif de Guava sur un seul article tant les fonctionnalités couvertes sont grandes. Outre les plus importantes décrites jusqu'ici, on signalera le package *primitives* contenant un grand nombre de méthodes utilitaires faisant gagner un temps précieux pour le travail avec les types primitifs de Java. Le package *util.concurrent* propose un ensemble de wrappers autour de *Future* du JDK qui se révèlent utiles dans des cas d'utilisation poussés de la concurrence. De même, la notion de *Service* est intéressante. Le package *io* demeure pertinent pour des environnements Java inférieurs au JDK 7. Pour les autres, l'API *Path* et les blocs *ARM* améliorent nettement la situation et ne nécessitent plus d'avoir recours à Guava. Enfin, un mot sur les dernières nouveautés de Guava avec un package *eventbus* apportant une gestion de communication de type souscription-publication entre composants et le package *cache* qui est une réponse légère et puissante aux besoins en cache mémoire des applications.

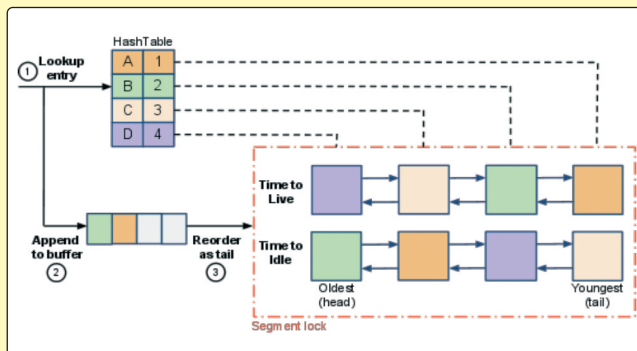
> Conclusion

Ce tour d'horizon aura mis en avant les fonctionnalités majeures de Guava dont beaucoup mériteraient d'être intégrées en standard au sein du JDK. Il y a d'ailleurs fort à parier que certaines d'entre elles fassent leur apparition dans le futur comme cela a été le cas avec la classe *Objects* du JDK 7. En introduisant également des notions de programmation fonctionnelle, Guava prépare le développeur à l'arrivée des lambdas expressions tout en facilitant la création de composants réutilisables. En outre, Guava met à disposition de tous un code robuste et éprouvé au sein duquel tout le savoir-faire et les bonnes pratiques de Google transparaissent avec la mise en exergue de l'immuabilité, du design pattern builder, des factories ou une utilisation optimisée des génériques. Ainsi, Guava permet au développeur de se concentrer pleinement sur la valeur ajoutée d'un programme en apportant des réponses simples et pratiques à des cas d'usages fréquents.

Sylvain Saurel – Ingénieur d'Etudes Java / JEE
sylvain.saurel@gmail.com

Prise en compte dans Maven

```
<dependency>
  <groupId>com.google.guava</groupId>
  <artifactId>guava</artifactId>
  <version>10.0.1</version>
</dependency>
```



API Differences

Between Guava 10.0.1 and Guava 11.0.1

Removed Packages

com.google.common.base.
internal

Added Packages

com.google.common.hash
com.google.common.math

Changed Packages

com.google.common.base
com.google.common.cache
com.google.common.collect
com.google.common.io
com.google.common.net
com.google.common.primitives
com.google.common.util.
concurrent

Implémentation du timing le caching concurrent (Google).

Prochain numéro
mai 2012 – n°153
Parution 31 mai

Spécial Open Source

- Créer son projet open source de A à Z
- Open cloud : l'open source pose les fondations du cloud
- Faire carrière dans l'open source : les filières, les jobs, les formations
- Microsoft fait sa « révolution » open source

Linux et hébergement :
quoi de neuf ?

Les dépendances en Javascript

Le temps où le Javascript n'était qu'un simple langage permettant de modifier quelques pages web est révolu. Maintenant utilisé comme un langage de script à part entière, il est le vecteur qu'utilisent de nombreux développeurs pour créer leurs outils.

De simples utilitaires jusqu'aux serveurs applicatifs, l'utilisation du Javascript dans les projets professionnels se démocratise, comme peuvent en attester les offres d'emploi postées chaque jour dans ce secteur. Pour autant, le Javascript en lui-même reste ce qu'il était avant tout ce remue-ménage; seule la perception qu'en ont les développeurs a changé. Or, bien que ce langage repose sur des concepts particulièrement intéressants, il n'a pourtant pas été originellement conçu pour l'emploi que nous lui trouvons aujourd'hui. L'exemple le plus flagrant de cet écart entre son objectif originel et les attentes que nous avons aujourd'hui de lui est son mode de gestion des dépendances, tout simplement inexistant. Là où tous les langages interprétés permettent à leurs scripts d'utiliser des modules externes, le Javascript reste l'ovni ne disposant d'aucun moyen natif permettant de réaliser cette tâche.

Heureusement, il dispose d'un avantage de taille : la force de sa communauté, probablement l'une des plus actives à ce jour. Le problème des dépendances a ainsi été traité par plusieurs comités, et selon plusieurs points de vue. Aujourd'hui, il existe donc un certain nombre de possibilités viables pour combler le manque originel.

Cet article a pour but de vous initier aux deux façons les plus répandues de résoudre ce problème : via les normes CommonJS et AMD. Enfin, nous passerons très rapidement sur la future norme Harmony et sur l'une de ses évolutions.

COMMONJS

En marge du comité officiel chargé de standardiser le Javascript, un nouveau groupuscule s'est créé, nommé CommonJS. Son but est d'instaurer des normes communes à l'ensemble des programmes écrits en Javascript, en vue d'une plus grande uniformité du langage. Parmi les travaux menés par CommonJS, le plus célèbre reste le système de dépendances, utilisé dans la plupart des outils système actuels. La syntaxe est particulièrement simple, d'où une adoption rapide. Elle repose sur l'emploi conjoint de deux composants :

- Une variable locale nommée `module` : Cette variable contient les informations d'un module. En CommonJS, chaque fichier est un module. Lorsqu'un module souhaite exporter des valeurs (les rendre accessibles aux autres modules), il doit le faire en modifiant l'objet `module.exports`.
- Une fonction prototypée `Module require(String identifiant)` : Lors de son appel, l'interpréteur va vérifier si le module indiqué par la chaîne `identifiant` est chargé. Si ce n'est pas le cas, il va récupérer le script le définissant (l'identifiant représentant généralement un chemin sur le système de fichier), et l'évaluer. Ainsi, lors du retour de la fonction, le module aura forcément été chargé (une exception étant lancée si le chargement échoue). La valeur de retour de la fonction est donc garantie être l'objet `module.exports` du module demandé.

moduleTest.js

```
var FOO = 42;

module.exports.getFoo = function () { return FOO; };
```

programme.js

```
var ModuleTest = require('./moduleTest');

console.log(ModuleTest.getFoo());
```

Comme vous pouvez voir, la syntaxe est légère et ne gêne pas la lecture du code. Cependant, cette simplicité la rend sujette à un défaut pas forcément visible au premier coup d'oeil, et pourtant extrêmement important compte tenu du but originel du Javascript ... elle ne fonctionne qu'en mode synchrone. Cela signifie que, lors de l'appel à la fonction `require`, le module indiqué doit impérativement être chargé avant le retour de la fonction. Pire encore, cette syntaxe implique également qu'un appel à `require` peut se trouver n'importe où dans le code, rendant impossible le listage des dépendances pour les charger simultanément avant l'exécution du code. Et quand bien même cela serait envisageable (par une analyse statique du code avant son exécution), le fait que ce soit une fonction implique pour finir que le paramètre `identifier` puisse également être une variable, rendant définitivement impossible la résolution de certaines dépendances avant exécution. Dans un environnement système, ces défauts sont globalement négligeables. En effet, les interpréteurs ont alors accès au système de fichier. Ces derniers étant rapides pour un usage ponctuel, le surcoût requis lors de tels appels synchrones n'est généralement pas visible pour l'utilisateur. Cependant, dans un contexte distant, la situation est différente. Un tel mécanisme implique que le navigateur devra lancer les requêtes au fur et à mesure de l'exécution, et une seule à la fois. En supposant un temps de chargement de 100 ms par fichier, l'exécution d'un fichier contenant dix dépendances prendra donc plus d'une seconde.

Petite note sur la portée des variables : la norme CommonJS prévoit que chaque module dispose de son propre scope d'exécution. En clair, cela signifie que la variable `FOO` déclarée plus haut ne sera pas accessible dans le programme principal. Ce comportement ne peut cependant pas être émulé à l'heure actuelle dans un navigateur. Une garde resterait donc nécessaire afin d'éviter de polluer l'espace de nom, donnant quelque chose comme :

moduleTest2.js

```
(function (module) {
    var FOO = 42;
    module.exports.getFoo = function () { return FOO; };
})(module);
```

AMD

La norme CommonJS est intéressante mais, comme nous venons de le voir, ne supporte pas de chargement parallèle des dépendances, la rendant sous-optimale dans un environnement distant. Pour résoudre ce problème, une nouvelle norme a été imaginée, nommée AMD (pour Asynchronous Module Definition). Elle repose sur les fondements de la norme CommonJS, mais y rajoute quelques éléments et règles :

- Une fonction prototypée `Undefined define([String name], [Array dependencies], Mixed module)`
Cette fonction permet de définir un nouveau module nommé `name`. Si le paramètre est absent, la bibliothèque implémentant `define` aura la charge de lui donner un nom, généralement basé sur l'emplacement du fichier dans l'arborescence (de tels modules sont

alors dits anonymes). Les dépendances du module sont indiquées dans un tableau, dont chaque entrée correspond à un nom de module requis. Lesdits modules sont alors eux-mêmes chargés (en parallèle, l'interpréteur ayant alors connaissance de la totalité des dépendances requises), puis passés en paramètre à la fonction `module` (s'il s'agit d'une fonction).

En effet, le paramètre `module` peut être, au choix, un objet ou une fonction. S'il s'agit d'un objet, il sera considéré comme étant le module. S'il s'agit d'une fonction, elle sera considérée comme étant la fonction de construction du module. Lorsque les dépendances se seront toutes chargées, cette fonction sera alors appelée avec les dépendances en paramètre, et son retour sera utilisé comme valeur pour le module courant.

- Une fonction prototypée `Undefined require([Object configuration], [Array dependencies], Function block)` Cette fonction permet simplement de définir un bloc de code (la fonction `block`) exécuté lorsque toutes les dépendances mentionnées dans `dependencies` sont chargées.

Comme pour la fonction `define`, les modules sont passés en paramètre à la fonction `block`.

Le paramètre `configuration` ne sera pas abordé ici, il sert à modifier la configuration du loader. Cela permet notamment de définir l'emplacement de certaines bibliothèques.

moduleTest.js

```
define(function () {  
  var FOO = 42;  
  return { getFoo: function () { return FOO; } };  
});
```

programme.js

```
require(['./moduleTest'], function (ModuleTest) {  
  console.log(ModuleTest.getFoo());  
});
```

Cet exemple montre la force de `requirejs` : ce code fonctionnera sans modification majeure entre la version Node et la version web. En effet, la seule différence résidera dans la façon de définir les fonctions `define` et `require`, qui dans la plupart des cas se résument à inclure préalablement une bibliothèque externe. AMD est principalement représenté dans le monde du Javascript par la bibliothèque `RequireJS`, qui intègre à la perfection toutes ces fonctionnalités. Elle est utilisable dans un contexte web autant que dans des applications système, respectant ainsi l'agnosticisme de la spécification originelle. En dépit de ses qualités, AMD souffre néanmoins de certaines critiques visant plus particulièrement sa syntaxe, relativement verbeuse par rapport à `CommonJS`. En effet, les blocs de code doivent impérativement être englobés dans des fonctions, et les dépendances doivent être écrites à la fois dans un tableau et dans les paramètres.

HARMONY

Javascript est un langage dérivant de la spécification `Ecmascript`. Bien que le Javascript dispose d'un certain nombre de mises à jour, c'est son aïeul qui fait le plus parler de lui ces derniers temps. En effet, le comité de standardisation en charge de l'`Ecmascript` travaille en ce moment même activement avec la communauté en vue de préparer les nouvelles caractéristiques de la prochaine spécification, nommée `Harmony`.

Parmi ses caractéristiques, une en particulier nous intéresse aujourd'hui :

la gestion des modules et chargeurs. En effet, conscients du vide laissé par l'absence d'un moyen natif de gérer les dépendances, l'un des principaux sujets de discussion est la spécification `Module`. Cette dernière propose l'utilisation des mots-clefs natifs `import`, `export`, `from`, `at` et `module` afin de spécifier des dépendances.

moduleTest.js

```
var FOO = 42;  
export function getFoo() { return FOO; }
```

programme.js

```
module ModuleTest at 'moduleTest';  
console.log(ModuleTest.getFoo());
```

L'avantage de cette syntaxe, pas forcément visible au premier coup d'oeil, réside dans le fait qu'il ne s'agit pas d'un appel de fonction (ce n'est qu'une association de mots-clefs). Par conséquent, cela implique que les standardisateurs sont libres de forcer un élément important : le paramètre de `from` et `at` ne peut en effet pas être une variable. Ou, du moins, s'il s'agit d'une variable, cette variable est considérée comme étant le module d'où extraire les fonctions, et non pas le chemin du module dans l'arborescence de code (pour `from` uniquement, `at` interdisant tout simplement autre chose qu'une chaîne de caractère). Cette différence par rapport à `CommonJS` comble la faille que nous avons pu observer. En effet, ceci implique que les dépendances du programme peuvent être résolues dès la phase de parsing du fichier source et non plus différées jusqu'à l'exécution effective du programme. Dans un cadre web, le navigateur peut alors paralléliser le chargement de ces fichiers sources. En dépit de son attrait, `Harmony` reste cependant à l'heure actuelle une douce espérance. En effet, la standardisation définitive de tous les éléments n'est pas prévue avant un certain temps, et les implémentations actuelles de Javascript ne disposent aujourd'hui pas des fonctionnalités futures. Parier sur ce système de chargement de dépendances n'est donc pas possible à l'heure actuelle.

Signalons tout de même le projet `Traceur`, qui vise à convertir un programme écrit en `Harmony` vers le même programme n'utilisant que des fonctionnalités de l'`Ecmascript 5`. Les modules peuvent donc dans une certaine mesure être utilisés par ce biais, mais une utilisation en environnement de production reste déconseillée.

EN CONCLUSION

Ce document est loin d'être exhaustif. De nombreux moyens alternatifs permettent de résoudre ces problèmes, généralement par le biais d'outils permettant de compresser plusieurs fichiers sources en un seul. Les énumérer serait impossible, mais citons néanmoins le compilateur `Closure`, la bibliothèque générique `JS.Class` ou encore `Browserify`.

Compte tenu de la diversité des solutions proposées actuellement à ce problème, le développeur reste donc confronté à un choix épineux, que seule l'expérience peut aider à résoudre efficacement. Les débutants seront peut-être tentés par la simplicité d'un `Browserify` simulant le fonctionnement de `CommonJS` pour les navigateurs, et les aînés seront peut-être plus tentés par un `RequireJS` gérant `AMD` à la perfection, mais disposant d'une syntaxe plus stricte.

Maël Nison

Étudiant à Epitech. Spécialisé dans les technologies Web il évolue notamment sur les nouvelles API JavaScript. Son projet de fin d'année est de créer un portage de Minecraft en WebGL.

Prise en main de WordPress

A l'origine simple outil pour faire du blog, WordPress est devenu au fil du temps un CMS complet. Il est open source, ouvert à la programmation et sympathique. Prise en main.

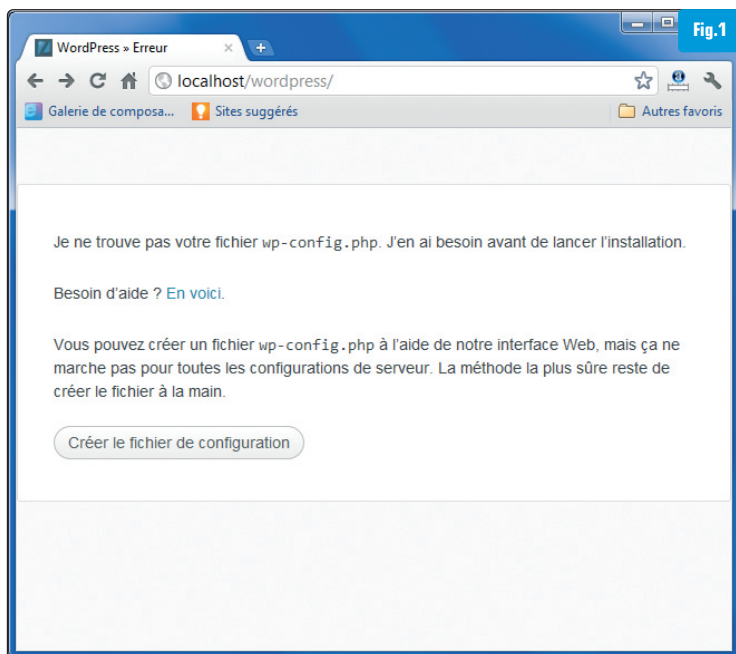
WordPress est né en 2001, du travail de Michel Valdrighi. Michel n'était pas satisfait de Blogger qu'il utilisait à l'époque et faisait le constat que trop rares étaient les outils de blog open source. Il a alors créé b2. Le projet a séduit la communauté. Et si Michel a délaissé plus tard b2 pour des raisons personnelles, son travail a été une source d'inspiration pour de nombreux développeurs. Plusieurs projets en ont émergé, dont celui fondé par Matt Mullenweg : WordPress. Aujourd'hui WordPress c'est gigantesque ! Cela comprend wordpress.com, le site de blogging gratuit, bien sûr le logiciel open source WordPress, et aussi une vaste communauté, dont une branche est francophone. Vous pouvez la retrouver à www.wordpress-fr.org.

1 VUE D'ENSEMBLE DE WORDPRESS

WordPress est et reste un outil qui va bien pour faire du blog, mais il représente aujourd'hui bien plus que cela. Au fil du temps, il est devenu un CMS à part entière, même s'il reste plus léger qu'un CMS comme Drupal. En fait, cette légèreté est un de ses atouts majeurs. Avec elle, vient la facilité d'installation et d'administration qui autorise à travailler avec un hébergement bas de gamme. Ici, administration est à comprendre au niveau du serveur qui héberge WordPress. En ce qui concerne l'administration du site au quotidien, WordPress présente une interface complète qui n'a rien à envier aux autres CMS. Comme eux, il est thématable, ce qui vous autorise à choisir l'apparence de votre blog, ou, mieux, pour le lecteur de Programmez ! à la définir vous-même. Enfin, il est extensible, rendant ainsi possible de programmer des Widgets et des modules d'extensions.

2 INSTALLER WORDPRESS

Pour installer WordPress, il est nécessaire de disposer d'un hébergement « bas de gamme » mais qui malgré tout vous donne accès à MySQL et PHP. Un simple accès FTP suffit pour installer WordPress. Vous n'avez pas besoin de louer un serveur dédié, même si cette dernière solution reste le nec plus ultra pour le geek. La procédure d'installation de WordPress est des plus simples. On télécharge la dernière version du logiciel qui vient sous la forme d'une simple archive, on décompresse celle-ci puis on upload les fichiers via le service FTP de l'hébergement. Ensuite (ou auparavant), on crée une base de données MySQL, puis on peut lancer la procédure d'installation/autoconfiguration de WordPress. Pour cela, il suffit de donner l'URL théorique de votre site/blog dans le navigateur. Il vous sera alors présentée une page vous informant de l'absence d'un fichier de nom wp-config.php [Fig.1]. La page vous informe qu'il est toujours possible de créer ce fichier manuellement en cas de problème. Mais a priori tout ira bien avec la procédure automatique. Vous pouvez donc cliquer sur le bouton 'Créer le fichier de configuration'. Continuez jusqu'à une page que vous devrez renseigner avec le nom de votre base de données, le nom d'utilisateur MySQL ainsi que son mot de passe [Fig.2]. Continuez la procédure jusqu'à arriver à un formulaire qui vous invite à renseigner votre compte administrateur [Fig.3]. La case 'Autoriser mon site à apparaître dans les moteurs de recherche comme Google...' mérite votre attention. Elle est normalement cochée par défaut. Ne la décochez pas si vous voulez que votre site soit référencé dans les moteurs de recherche. Vous pourrez éventuellement revenir ultérieurement sur ce choix dans l'interface d'administration de WordPress. Nous verrons plus



Début de la procédure d'installation : génération du fichier wp-config.php



Dites ensuite à WordPress quelle base de données vous souhaitez utiliser.

loin comment celui-ci procède pour gérer ce point. Poursuivez la procédure d'installation qui va se terminer très rapidement. Vous serez alors invité à vous connecter au site sous votre compte utilisateur, ce qui vous amène à l'interface d'administration de Wordpress [Fig.4]. C'est terminé, vous avez un nouveau site, propulsé par Wordpress :) Vous pouvez commencer à publier quelques articles.

3 QUELQUES RÉGLAGES POST-INSTALLATION

Pour commencer, il est bon de savoir tenir votre Wordpress à jour. Vous trouverez pour cela une procédure automatisée sous le menu 'Tableau de bord'. Il est également possible de mettre à jour manuellement. Dans tous les cas, il est recommandé de faire au préalable une sauvegarde de la base de données, et il est également recom-



Au cours de la procédure, vous devrez renseigner votre compte administrateur.

mandé de désactiver les extensions. Des extensions, il en existe de nombreuses, et Wordpress a l'excellente idée de vous en présenter la liste depuis l'interface d'administration [Fig.5]. Vous pourrez également installer, activer, désactiver ces extensions directement depuis l'interface. Installez les extensions qui vous séduisent. Je vous suggère aussi de visiter la page 'Widgets' sous le menu 'Apparence'. Enrichissez la barre latérale de votre site avec quelques widgets. Pour la suite de cet article, il est pertinent d'ajouter au moins le widget 'catégories'. Rendez-vous ensuite à 'Général' sous le menu 'Réglages', et réglez le fuseau horaire, ainsi que le format de date que vous souhaitez voir utilisé par Wordpress. Allez ensuite à 'Vie privée', puis cliquez sur le bouton radio 'Demander aux moteurs de recherche de ne pas indexer ce site'. Validez, puis examinez le source d'une page quelconque hors interface d'administration. La page d'accueil de votre site par exemple. Vous constaterez que Wordpress ajoute cette balise dans l'entête de chaque page :

```
<meta name='robots' content='noindex,nofollow' />
```

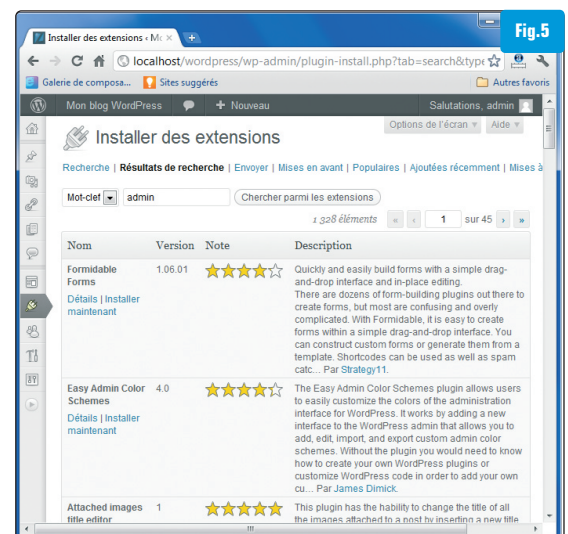
Cette balise demande aux robots des moteurs de recherche de ne pas indexer les pages (noindex), ni de suivre les liens qu'ils pourraient découvrir dans la page (nofollow). Tel est le moyen tout ou rien de Wordpress, là où d'autres CMS vous recommanderons de donner un fichier robots.txt à la racine de votre site, afin de pouvoir spécifier précisément quelles parties devront être indexées, et quelles autres ne le devront pas. Revenez à l'interface et cliquez sur 'Demander aux moteurs de recherche d'indexer ce site'. Bien sûr vous avez toujours la possibilité de créer votre fichier robots.txt. Dans ce cas, vous aurez besoin de savoir travailler les mécanismes de réécriture d'URL que propose Wordpress.

4 RÉÉCRITURE D'URL

Nous allons maintenant aborder un point qui se doit d'être soigné si l'on souhaite que le site soit bien référencé dans les moteurs de recherche : la réécriture d'URL. Commencez par créer une catégorie (taxinomie) 'Langages' dans l'interface d'administration. Puis créez deux articles. Par exemple un article parlant de Java et un article parlant de C++, et rangez ces deux articles sous la catégorie Langages. Ceci fait, rendez-vous à la page d'un de ces articles.



Votre première connexion vous amènera à l'interface d'administration de Wordpress.



Directement depuis l'interface d'administration vous pourrez connaître les extensions disponibles et les installer.

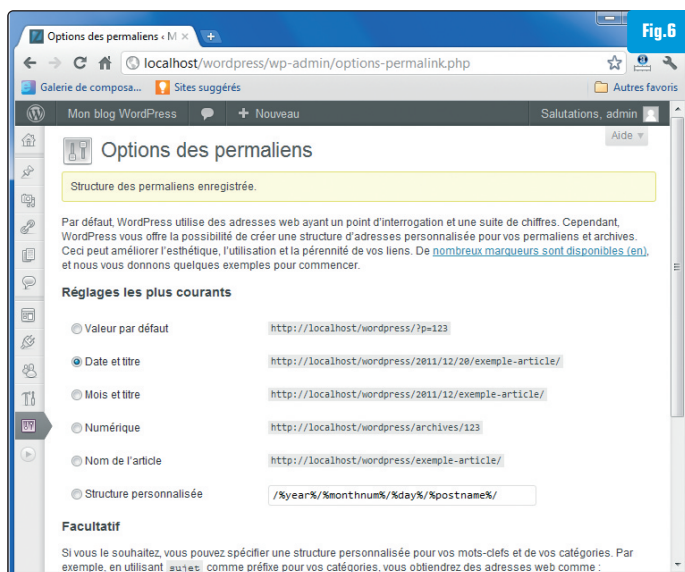
Regardez l'URL affichée dans la barre d'adresse du navigateur. Vous voyez quelque chose comme : <http://localhost/wordpress/?p=5> Cette URL vous indique que le site affiche le 5^e contenu publié (p=5), mais ceci ne renseigne que vous, et encore. Et surtout, cela ne renseigne pas les moteurs de recherche qui adorent trouver des termes intelligibles dans les URL. En outre, les moteurs ne tiennent en général pas compte de tous les paramètres (?p=5 est un paramètre) d'une URL. Par exemple, Google ne va pas au-delà du 2^e paramètre. Nous préférierions donc afficher une URL de cette forme :

<http://localhost/wordpress/admin/langages/tout-sur-java>

C'est-à-dire une URL qui contienne le nom de l'auteur de l'article, la catégorie sous laquelle est classé l'article et le titre de l'article. Cette opération s'appelle une réécriture d'URL. Il y a plusieurs solutions pour y parvenir. Ainsi, si le serveur Web de notre site est Apache, nous pouvons définir des réécritures d'URL via des expressions régulières dans un fichier .htaccess. Cette approche est parfaitement déplaisante. En outre, le serveur Web de notre hébergement bas de gamme n'est peut-être pas Apache... WordPress, qui veut être opérationnel dans les hébergements les plus frustrés propose deux possibilités pour réécrire les URL. La première se situe dans l'interface d'administration, la seconde consiste en l'appel d'API lorsque l'on programme un module d'extensions, mais là vous retrouverez les expressions régulières ;-). Rendez-vous dans l'interface d'administration sous le menu 'Réglages/Permalien'. Cliquez par exemple sur le bouton radio 'Date et titre', [Fig.6] puis retournez à la page de votre article. La page apparaît tout à fait normalement, mais son URL a changé. Elle ressemble désormais à quelque chose comme : <http://localhost/wordpress/2011/12/20/tout-sur-java/> Revenez maintenant à la page d'administration, et observez que WordPress a automatiquement rempli le champ de saisie à côté du bouton radio 'Structure personnalisée'. Vous voyez :

```
/year%/monthnum%/day%/postname%/
```

Il s'agit d'une succession de marqueurs à partir desquels WordPress réécrit les URL. La page d'administration vous présente un lien sur une page en anglais qui énumère tous les marqueurs reconnus par WordPress. Muni de ces renseignements, il vous suffit de cliquer le bouton 'Structure personnalisée' et de renseigner son champ comme ceci :



L'interface d'administration de WordPress permet de définir des réécritures d'URL.

```
/author%/category%/postname%
```

Et vous obtenez l'URL sous la forme souhaitée. Il est intéressant de savoir que vous n'êtes pas limité à des marqueurs dans ce champ. Ainsi si vous donnez :

```
/monsuperblog/author%/category%/postname%
```

L'URL réécrite sera :

<http://localhost/wordpress/monsuperblog/admin/langages/tout-sur-java>

Cette dernière possibilité peut s'avérer cruciale dans le cas, malgré tout assez peu probable, où votre hébergeur ne vous offre pas un serveur Web qui permet la réécriture d'URL, et que vous n'avez aucune possibilité de le configurer. Il existe alors une astuce pour contourner le problème. Il suffit de donner en guise de structure personnalisée :

```
/index.php/author%/category%/postname%
```

Comme nous l'avons dit, WordPress peut tourner dans les environnements les plus spartiates. Il saura traiter ce cas. Vos URL auront alors la forme :

<http://localhost/wordpress/index.php/admin/langages/tout-sur-java>

Ce qui n'est certes pas idéal, mais ça a le mérite d'exister et de fonctionner, ainsi que d'aider au référencement de votre site.

5 LE CRON

Cron est un programme UNIX qui permet de lancer des tâches à des moments précis et planifiés, ce qui en fait un outil d'administration précieux. On l'utilise par exemple pour faire des sauvegardes régulières de base de données, ou pour expédier des mailings aux heures de basse fréquentation d'un site. Cron existe bien évidemment sous Linux, qui est le système d'exploitation le plus répandu sur les serveurs susceptibles d'héberger WordPress. On le trouve également sous Mac OS X et il a son équivalent sous Windows. Le problème avec Cron est qu'il nécessite une connexion SSH pour être configuré. Il est fort à craindre que si votre hébergement est mutualisé, vous ne pourrez y accéder. Là encore WordPress, qui est conçu pour tourner dans ce genre d'environnement, propose une solution alternative. Cette solution consiste en la programmation d'API, pour planifier les tâches à exécuter. Ces API permettent notamment de définir des tâches à exécution unique et des tâches à exécution multiples. Voici un exemple :

```
function tache_unique() {
    $fp = fopen("log.txt","a+");
    fputs($fp, "Execution tache unique « . time() . «\n\r»");
    fclose($fp);
}

wp_schedule_single_event(time()+10, 'tache_unique');
```

La tâche à exécuter est dans le corps de la fonction tache_unique. Nous nous contentons d'écrire un petit message dans un fichier qui est baptisé fichier de log pour la circonstance. La planification est placée par l'appel à wp_schedule_single_event qui demande un timbre à date au format UNIX en premier argument. Nous fixons le moment d'exécution à 10 secondes après l'appel à wp_schedule_single_event. Nous devons maintenant répondre à une question existentielle : où diable déposer ce code ? La bonne réponse

se est : dans un module d'extension. Nous apprendrons à écrire un module d'extension à WordPress dans un prochain article. En attendant nous allons utiliser une des particularités de WordPress : les fichiers `functions.php`. Ces fichiers permettent de programmer rapidement des 'extensions'. On trouve un tel fichier, général au système, dans le répertoire `wp-includes`. On en trouve également dans les thèmes. Un thème est l'ensemble des fichiers PHP qui servent à rendre les pages. Ainsi sur votre WordPress fraîchement installé, qui utilise le thème par défaut `twentyeleven`, vous trouverez un fichier `functions.php` dans le répertoire `wp-content/themes/twentyeleven`. Comme tous les fichiers php constitutifs de WordPress, les fichiers `functions.php` sont chargés à chaque fois que le système doit afficher une page, et le code qu'ils contiennent est exécuté. Vous ne pouvez pas déposer notre code d'exemple dans le fichier `functions.php` général, car au moment où celui-ci est exécuté, la fonction `wp_schedule_single_event` n'est pas encore disponible. Par contre, vous pouvez utiliser sans problème le fichier `functions.php` du thème `twentyeleven`. Il est important d'insister sur le fait que des 'extensions' ainsi créées, ne sont en fait pas des extensions. Ce ne sont que des hacks qui permettent de faire des essais rapides et rien de plus. En outre, chaque mise à jour d'un thème ou du système est susceptible d'écraser le code contenu par ces fichiers. Vous trouverez le fichier `log.txt` dans le répertoire `wp-admin`.

6 BIEN COMPRENDRE LE CRON WORDPRESS

Il est tout d'abord important de bien réaliser que ce Cron n'en est pas tout à fait un. Car, ainsi que vous l'avez compris, son exécution est dépendante du chargement du système, donc des visites effectuées sur le site. Si dans notre exemple, nous avons planifié une tâche pour 10 secondes après le moment de la planification, l'exécution réelle n'aura lieu que lors de la prochaine visite se produisant 10 secondes après. Si une visite se produit avant, la tâche ne sera pas exécutée, ce qui est logique. Mais s'il n'y a aucune visite après les 10 secondes, la tâche n'est jamais exécutée. S'il y a une visite seulement une heure après, alors la tâche est exécutée une heure après. Enfin notre code donne l'impression de planifier la tâche en boucle. Ceci est en partie vrai. En fait `wp_schedule_single_event` ne planifie que toutes les 10 minutes. Ses appels restent sans effet dans ces intervalles. Si nous supposons que notre site reçoit deux visiteurs par minute, notre tâche unique ne sera malgré tout lancée que

toutes les 10 minutes. En revanche, l'API de planification de tâche périodique, `wp_schedule_event`, ne connaît pas ce délai de latence de 10 minutes. Il vous incombe de vérifier qu'une tâche n'est pas déjà planifiée, par exemple comme ceci :

```
if(!wp_next_scheduled('tache_unique')) { //etc...
```

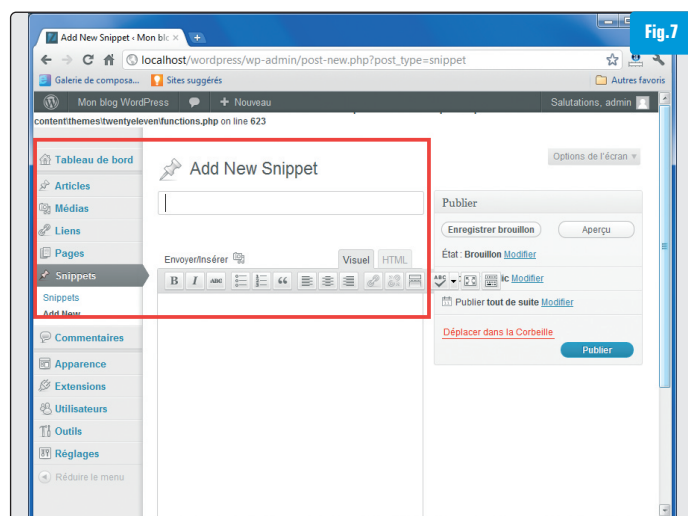
7 LES TYPES DE CONTENU PERSONNALISÉS

Au départ WordPress ne permettait de ne publier que des articles, alias billet de blog. Puis un nouveau type de contenu est apparu : les pages statiques. Il est désormais possible de définir des types de contenu supplémentaires, qui vont trouver leur utilité si vous faites plus que bloguer avec WordPress, et si vous l'utilisez plus comme un gestionnaire de contenu généraliste. Supposons que vous vouliez partager des petits morceaux de code (snippets) avec vos visiteurs. Vous ne mettez pas ces contenus dans des articles, car vous souhaitez que ce contenu soit statique. Mais vous ne les mettez pas non plus dans les pages statiques car vous réservez celles-ci à vous présenter par exemple. Il vous faut donc un autre type de contenu. Et voici comment le créer :

```
register_post_type(
    'snippet',
    array(
        'label' => __('snippet'),
        'labels' => array(
            'name' => __('Snippets'),
            'singular_name' => __('Snippet'),
            'add_new' => __('Add New'),
            'add_new_item' => __('Add New Snippet'),
            'edit_item' => __('Edit Snippet'),
            'view_item' => __('View Snippet'),
            'search_items' => __('Search Snippets'),
            'not_found' => __('No snippets found'),
            'not_found_in_trash' => __('No snippets found in trash'),
        ),

        'public' => true,
        'hierarchical' => true,
        'menu_position' => 20,
        'supports' => array('title', 'editor', 'command'),
    )
);
```

Moyennant quoi la création d'un nouveau contenu de type Snippet est disponible dans l'interface d'administration [Fig.7]. Nous renvoyons le lecteur à la documentation officielle disponible sur <http://codex.wordpress.org/>. La fonction `__()` sert à la traduction automatique. Vous devrez procéder comme dans cet exemple si vous voulez que votre code soit internationalisé. Un autre point important est la position dans le menu. WordPress numérote les positions dans le menu de 5 en 5 à partir de 0. Donner 20 nous positionne ainsi en fin de liste, sous l'entrée 'Pages'. Dans un article à venir, nous examinerons comment construire un module d'extension complet et propre sur lui :-)



Nous pouvons désormais créer un contenu personnalisé depuis l'interface d'administration.

Frédéric Mazué - fmazue@programmez.com

Bug « élément feed fermant manquant dans un flux OData »

Ce bug n'en est pas vraiment un. Il s'agit d'un comportement voulu, mais tellement étrange que la désignation de bug reste justifiable. Imaginons une solution Visual Studio 2010 de type web, dans laquelle nous aurions rajouté un service de données WCF. Pour que cet élément soit disponible, il suffit d'installer le SDK « WCF Data Services ». Ce type de service permet d'exposer une source de données IQueryable selon le protocole OData. Ce protocole REST/Atom est particulièrement adapté à la mise à disposition de données publiques, léger et sans schéma fixé à l'avance. Pour que ce service fonctionne, il convient de remplir le code-behind avec un code correspondant globalement à ceci :

```
using System.Collections.Generic;
using System.Data.Services;
using System.Data.Services.Common;
using System.Linq;

namespace ODataFeedManquant
{
    [EntityPropertyMapping("Activite", SyndicationItemProperty.Summary, SyndicationTextContentKind.Plaintext, true)]
    [DataServiceKey("ID")]
    public class Unite
    {
        public string ID { get; set; }
        public string Activite { get; set; }
    }

    public class Ensemble
    {
        private static List<Unite> _Liste = new List<Unite>();

        static Ensemble()
        {
            for (int i = 0; i < 20; i++)
                _Liste.Add(new Unite() {
                    ID = i.ToString(),
                    Activite = (new string[] { «Boulot», «Lego», «Dodo» })[i % 3],
                });
        }

        public IQueryable<Unite> Unites
        {
            get { return _Liste.AsQueryable(); }
        }
    }

    public class Historique : DataService<Ensemble>
    {
        public static void InitializeService(DataServiceConfiguration config)
        {
            config.SetEntitySetAccessRule(«*», EntitySetRights.AllRead);
            config.DataServiceBehavior.MaxProtocolVersion = DataServiceProtocolVersion.V2;
            config.MaxResultsPerCollection = 10;
        }
    }
}
```

```
config.SetEntitySetAccessRule(«*», EntitySetRights.AllRead);
config.DataServiceBehavior.MaxProtocolVersion = DataServiceProtocolVersion.V2;
config.MaxResultsPerCollection = 10;
}
```

Cette implémentation expose une vingtaine de données dans une collection nommée « Unites », chacune des lignes comprenant un code identifiant qui est la clé ainsi qu'une chaîne « Activite » qui sera utilisée comme description résumée des entrées ATOM sortantes.

PROBLÈME

Si nous appelons cette collection « Unites » du service depuis un navigateur Firefox, en entrant <http://localhost:7194/Historique.svc/Unites> (7194 est le port choisi arbitrairement par Cassini), tout se passe apparemment très bien. Nous ne voyons que 10 des 20 lignes de données, mais ce comportement est dû à la ligne de code suivante : `config.MaxResultsPerCollection = 10;` Pourtant, si nous utilisons un navigateur Internet Explorer ou un client plus évolué comme PowerPivot pour Excel pour consommer ce service OData, nous obtenons une erreur. Après avoir fini par regarder le flux dans Fiddler, il apparaît que le flux est en réalité tronqué : l'élément fermant `</feed>` (qui est pourtant l'élément racine du flux XML sortant) est tout simplement manquant ! Firefox s'en accommode, ce qui est limite dangereux, mais l'éditeur de code source montre bien ce problème. Le flux XML commence bien par un `<feed>`, contenant les `<entry>` correspondant aux lignes de données. Mais à la fin du fichier, pas de `</feed>` fermant. Toutefois, il se trouve un code d'erreur qui nous donne un début d'explication : la réponse dépasse le nombre maximal de résultats par collection (10, en l'occurrence). Mais pourquoi le fichier XML n'est-il pas bien formé ?

EXPLICATION

La réponse est fournie par Microsoft. Et accrochez-vous bien : c'est prévu comme ça ! Microsoft renvoie volontairement un XML mal formé pour forcer le client à planter. Microsoft a considéré que le fait de renvoyer un élément `<error>` ainsi qu'un code HTTP risquait de ne pas suffire, et qu'il fallait prendre encore plus de sécurité sur le fait que le client planterait et ne risquerait pas d'avaler l'erreur. Peine perdue car Firefox la masque de toute façon, et c'était d'ailleurs le comportement d'Internet Explorer également jusqu'à la version 8.

SOLUTION

La solution, par défaut, est de ne simplement pas utiliser cette fonctionnalité de blocage de la taille de remontée `MaxResultsPerCollection`, et de compenser en mettant en place une limite plus souple avec une ligne comme `config.SetEntitySetPageSize(«Unites», 10);`

Jean-Philippe Gougoux
Responsable Pôle Architecture / Formation / Innovation – MGDIS
<http://www.gougoux.com/blog-fr/>

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 / Événements | Newsletter | Vidéos



www.solutions-logiciels.com

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

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

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

10 numéros par an.

☐ M. ☐ Mme ☐ Mlle Société

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

NOM Prénom

N° rue

Complément

Code postal : [] [] [] [] [] Ville

Adresse mail

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

DÉVELOPPEZ 10 FOIS PLUS VITE

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.

**VERSION
EXPRESS
GRATUITE**
Téléchargez-la !

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

Eu «Langage le plus
productif du marché»

WINDEV, WEBDEV
et WINDEV Mobile
sont compatibles



917

NOUVEAUTÉS



Nouveau:
créez des applications
iOS (**iPhone,**
iPad)

**EXIGEZ WINDEV 17
POUR LE DÉVELOPPEMENT
DE VOS APPLICATIONS**

Fournisseur Officiel de la
Préparation Olympique



► **DEMANDEZ VOTRE DOSSIER GRATUIT**

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

www.pcsoft.fr