

PROgrammez !

www.programmez.com

mensuel n°118 - avril 2009

Tout ce qu'il faut savoir pour

Développer Linux

sur

- Maîtriser les standards :
LSB, AppChecker
- Développer vos
premières applications
- Les développeurs
en parlent
- Faire carrière dans
l'Open Source
- Mon code source
est-il légal ?

UML

Le futur de la modélisation

Ce que prépare
Microsoft avec OSLO



Plongée
au cœur
des Mozilla
Labs

mozilla LABS

Web 2.0

Comment choisir
son framework Ajax ?

Outil

WinDev : l'atelier
de développement
à tout faire

© John Schreiner/illustrationOnLine.com

Serveur

Terracotta : simplifiez-vous le cluster

+ GPU

Maîtriser la puissance de Cuda

Coding4fun

Faites parler vos applications avec MS Speech !

Open Source

Les fondamentaux de PHP

M 04319 - 118 - F: 5,95 €



Développez 10 fois plus vite

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

WINDEV®

RÉUSSISSEZ TOUS VOS PROJETS
AVEC L'OUTIL DE DÉVELOPPEMENT
ÉLU **LE PLUS PRODUCTIF***



VERSION
EXPRESS
GRATUITE
Téléchargez-la !

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

WINDEV 14 est livré complet: éditeur d'analyses (UML,...), **RAD**, patterns, **lien avec toutes les bases de données (ODBC, OLE DB)**, Oracle, SQL Server, AS/400, Informix, DB2..., lien natif MySQL, PostgreSQL, **base de données Client/Serveur HyperFileSQL gratuite incluse**, Générateur d'états PDF, Codesbarres, **Accès natif SAP R/3, Lotus**

Notes, Gestion de planning, Gestion des Exigences, **L5G**, SNMP, **Bluetooth**, TAPI, OPC, FTP, HTTP, Socket, Twain, API, DLL, **domotique**, liaisons série et USB, **débogage à distance**, profiler, refactoring, **génére des applications JAVA à la demande**, multilingue automatique, **Gestionnaire de versions**, Installateur 1-clic, etc...

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

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



UN CODE
MULTI-PLATEFORME :
Windows, .Net, Java,
PHP, J2EE, XML,
Internet, Ajax, Pocket
PC, SmartPhone,
Client riche ...

Logiciel professionnel

DEMANDEZ LE DOSSIER GRATUIT

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

Tél: **04.67.032.032** ou **01.48.01.48.88**
info@pcsoft.fr



Fournisseur Officiel de la Préparation Olympique

www.pcsoft.fr



Sommaire

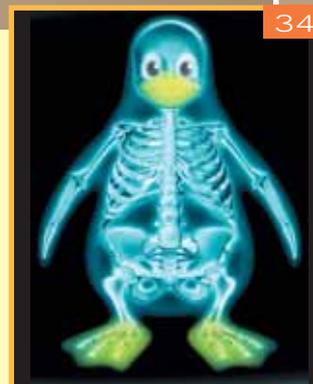
\\ actus	
Agenda	6
L'actualité en bref	8
\\ architecture	
Modélisation : marché et tendances	12
Oslo , la future plate-forme de modélisation de Microsoft	14
\\ SGDB	
Hibernate survival guide (2e partie)	18
\\ gros plan	
Choisir son framework AJAX	22
\\ outils	
WinDev : l'atelier de développement tout en un	28
\\ dossier :	
Tout ce qu'il faut savoir pour développer sur Linux	34
La parole aux développeurs !	34
Portrait de développeur	36
Bien débiter son développement sous Linux	37
Développement rapide avec Gnome : GTK+ et Python	42
Validez vos applications Linux avec AppChecker	48
Pourquoi et comment utiliser Linux Standard Base ?	51
Mon code est-il légal ?	53
Métier : développeur Open Source	54
Au cœur du Mozilla Labs	57
\\ code	
Générer vos applications à partir des modèles UML2 avec BLU AGE 2009	59
Terracotta : une approche différente du clustering	62
Maîtriser les fondamentaux de PHP ! (2e partie)	66
Découvrir et mettre en oeuvre CUDA (2e partie)	68
La reconnaissance vocale sous Windows	72
Développez en Ruby on Rails avec 3rdRail (2e partie)	76
Plus de cœurs, plus vite, plus scalable : du multi-core au " many-core "	78
\\ temps libre	
Créer un jeu vidéo de A à Z avec XNA (3e partie)	80
Les livres du mois	82



22



28



34



57



78

Contenu du CD-ROM

- Eclipse IDE 3.4 (Ganymede) en version complète,
- Blu Age Build en version d'évaluation 30 jours,
- Guide d'évaluation,
- Exemple d'un modèle PIM exécutable : modélisation UML 2 d'une application de prise en main simple : la gestion d'un annuaire téléphonique.

Blu Age Build - Fonctionnalités clefs :

- intégration native dans Eclipse (plugins),
- élaboration assistée de modèles UML 2 indépendants de la plateforme et exécutables (PIM),
- débogage des modèles,
- intégration automatique des maquettes XHTML des IHMs avec le PIM,
- transformation instantanée et intégrale du PIM en PSM, et du PSM en code Java,
- déploiement de l'application générée dans l'environnement Eclipse.

Configuration requise :

- PC sous Windows XP SP2,
- 2Go minimum de mémoire (idéalement 4Go),
- Disque dur avec 1Go d'espace libre,
- JAVA JDK 1.6,
- Tomcat 5.5.

Prise en main,
voir **page 59**



AGILE MODEL TRANSFORMATION
2009 EDITION - BUILD

Modélisez et générez votre première application Java EE !

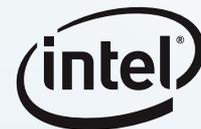
Eclipsez vos problèmes, Sublimez vos modèles,
Rationalisez vos développements!

Blu Age Build est un environnement intégré pour la mise au point de modèles PIM conformes à l'UML2, proposant notamment des fonctions de débogage de modèles, ainsi qu'une transformation instantanée des modèles en applications JEE prêtes à être déployées.



Eclipse, Built on Eclipse and Eclipse Ready, MIT, Higgins are trademarks of Eclipse Foundation, Inc.

PROgrammez! Supplément éditorial n°118. Échantillon gratuit • Ne peut être vendu séparément



DANS LA LUNE. SUR LA LUNE.



FAITES EVOLUER VOTRE CODE.

L'innovation en développement parallèle.

Découvrez le vrai potentiel de vos applications Windows* avec Intel® Parallel Studio. Toute la panoplie d'outils pour simplifier le développement multi-core, de l'analyse à la compilation, en passant par le débogage et l'optimisation. Fait pour les applications sérielles d'aujourd'hui et les innovations parallèles de demain.

Pour en savoir plus et tester la version beta: www.intel.com/software/parallelstudio





Retour aux sources

En préparant notre grand dossier Open Source / Linux, nous avons entendu la plupart des acteurs français du Libre répéter combien le secteur tirait bénéfice de la crise. L'Open Source, arme anti crise ? Avec les nouvelles contraintes financières, il contribue à réduire le coût de fonctionnement de l'informatique. L'alternative en tout cas devient de plus en plus sérieuse pour les entreprises encore hésitantes à sauter le pas. Des responsables encore frileux pouvaient bloquer le déploiement massif de l'Open Source dans leur entreprise. Les succès des serveurs d'application libre, dans la bureautique, le navigateur, prouvent aujourd'hui la pertinence de l'offre. Cependant, sur le poste de travail, Linux peine encore à trouver sa place et cela n'encourage pas les éditeurs à porter les solutions sur Linux... pas encore du moins.

Année après année, les modèles de développement se stabilisent et s'améliorent. Les éditeurs de distribution savent qu'une des clés du succès est d'attirer les développeurs. La fondation Linux tente de les séduire, les draguer : " venez chez nous, mon système est plus mieux que celui d'en face ". Mais la crise modifie également (l'attitude des éditeurs propriétaires) qui n'hésitent plus à casser les prix pour continuer à vendre, à renouveler les licences. Et s'il y avait un effet anti-Vista, l'arrivée dans quelques mois de Windows 7 réduit l'argument...

L'un des succès de l'Open Source vient de la simplicité. Rappelons-nous du douloureux passage entre VB6 et VB.Net. De nombreux VBistes ont refusé, et refusent encore, de passer à .Net, car le saut fonctionnel et technique reste important. Cette mauvaise gestion de la transition pèse lourd. Si PHP, Ruby, Python, pour ne citer que ceux-là, connaissent un énorme succès, c'est d'abord grâce à la simplicité du langage !

Aujourd'hui les langages ont trop tendance à grossir, à se voir rajouter sans cesse de nouvelles fonctions. Et cette profusion de nouveautés fait peur à ceux qui voudraient coder une petite application, ou même leur site web avec quelques fonctions riches. On peut également regretter les changements constants de ces outils et langages. Et si ces améliorations nuisaient en fait aux développeurs et aux codes ? Le débat est lancé...

PHP a séduit le développeur web par sa facilité d'apprentissage. Java tente de corriger le tir depuis Java 5 même si cela reste compliqué. VB.Net et C# doivent désormais faire leur coming out et simplifier drastiquement le langage. L'exemple de Smallbasic montre que Microsoft sait le faire. Il s'agit là sans doute également de la raison du succès des ateliers tout-en-un, tel WinDev.

Si nous reconnaissons toute la puissance d'un .Net, d'un Java, ces plates-formes, apparaissent élitistes, coupées de la majorité des développeurs hobbystes et même d'une frange des professionnels.

La question est : comment réconcilier les développeurs avec leur langage ? Voilà le véritable défi, le véritable buzzword que les éditeurs, les communautés doivent considérer.

■ François Tonic

Rédaction : redaction@programmez.com
Directeur de la Rédaction : Jean Kaminsky
Rédacteur en Chef : François Tonic - ftonic@programmez.com
Ont collaboré : F. Mazué, C. Remy, L. Guillois, D. Neary.
Experts : P. Guernonprez, F. Pedro, A. Chen Kuang Piao, O. Lance, H. Darmet, O. Dasini, S. Haim-Lubczanski, T. Marendic, W.Lawson, D. Vandeveld, A. Ageorges, F. Lepied, J-L. Recoussine.
Crédits photo : © John Schreiner // IllustrationOnline.com
Publicité : Régie publicitaire, K-Now sarl
Pour la publicité uniquement :
Tél. : 01 41 77 16 03 - diff@programmez.com
Editeur : Go-02 sarl, 6 rue Bezout
75014 Paris - diff@programmez.com
Dépôt légal : à parution - Commission paritaire : 0712K78366 ISSN : 1627-0908
Imprimeur : ETC - 76198 Yvetot
Directeur de la publication : J-C Vaudecrane
Ce numéro comporte un CD rom
Un encart Component Source sur les abonnés et une partie du tirage.

Abonnement : Programmez 22, rue René Boulanger, 75472 Paris Cedex 10 - abonnements : programmez@groupe-gli.com Tél. : 01 55 56 70 55 - Fax : 01 55 56 70 20 - 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 en ligne.

L'INFO PERMANENTE
WWW.PROGRAMMEZ.COM



PROCHAIN NUMÉRO

N°119 mai 2009,
parution 30 avril

✓ **Cloud computing**
révolution ou évolution,
le cloud change radicalement
la manière de penser et
de coder son application.
Un dossier à ne pas manquer !

✓ **Java**
Tout savoir sur Eclipse 4.0

✓ **Coding4fun**
Quand les développeurs
se lâchent !
Android Wii Remote, Tag...

■ **Microsoft** propose depuis début mars ses premiers services Online : Exchange, SharePoint, Office Communications, Office Live Meeting, disponibles mondialement. Microsoft espère rapidement développer les partenaires (qui pourront vendre les services) et surtout proposer de nouveaux services. C'est la concrétisation de l'approche Software + Services de l'éditeur.

■ **Aladdin** a dévoilé durant le dernier Cebit de nombreuses nouveautés : eToken Pro Anywhere (authentification forte, Audit USB Instantané (outil mobile de test pour vérifier le niveau de protection), HASP SRM 3.6 (gestion de DRM, support 64-bits), HASP SRM SaaS Pass (version SaaS pour gérer l'authentification des utilisateurs et les contrôler).

■ **BlackBerry** mise aussi sur un App Store à la Apple pour les applications. Pour préparer les applications françaises, le constructeur propose une liste de 10 questions pour savoir si votre logiciel est prêt pour la boutique en ligne. Une bonne checklist.

■ **InterSystems** complète sa solution d'intégration Ensemble avec de nouvelles fonctions pour les architectures orientées services : support de WS-Security 1.1, surveillance accrue des applications composites, gestion du HL7 (échange des données médicales), support de ebXML.

■ La section développeur de Microsoft, **MSDN** ouvre une zone développeur Java ! Avec les nécessités d'interopérabilité, de la disponibilité des SDK non .Net pour Azure et d'autres technologies, il fallait bien que cela arrive ! On trouve des ressources purement Java, comment intégrer Java et .Net... Site : <http://msdn.microsoft.com/fr-fr/architecture/aa718825.aspx>

Salon

Linux et l'Open Source s'exposent à Paris

Deux mots d'ordre pour cette nouvelle édition du salon Solutions Linux : crise et cloud computing. Ce sont les thèmes des keynotes du salon. Et preuve que Linux est un marché important en entreprise, Oracle assure une des deux keynotes...

Le salon fête dignement sa 10e édition avec de nouvelles dates (du 31 mars au 2 avril) et un nouveau lieu : Porte de Versailles. Ce sont plus de 160 exposants, 60 associations et de nombreux experts qui vont occuper les allées.

Un beau succès. Deux événements ont lieu durant l'exposition, la conférence annuelle d'OW2 qui se fait bien discrète depuis quelques mois et une grande conférence de presse d'April sur le pacte logiciel libre. Sun tient aussi un rôle de vedette avec son aquarium, dédié cette année à Glassfish et Java en général. Occasion parfaite pour découvrir la v3 du serveur d'applications prévu pour cet été. Impossible de ne pas parler Cloud Computing : outre une keynote dédiée à ce

sujet, des ateliers y sont consacrés comme celui de Plant Work, avec Sun (31 mars à 15h45), ou encore la table ronde autour du Cloud Computing et le rôle de l'Open Source. Le DBA y trouvera une session particulièrement intéressante sur Oracle et les langages scripts open source. Côté développeur, les conférences ciblant le développement sont assez peu nombreuses. Le principal atelier est " Développeur " (mercredi 1 avril, deux sessions).

Serveur

VMware veut imposer le " Cloud privé " !

Comment mieux faire passer le message du *Cloud computing* auprès des entreprises, générer de l'activité pour les partenaires et continuer à vendre des licences ? VMware a donné une réponse en mettant en avant le " Cloud privé ", bref, interne à l'entreprise... L'éditeur répond à cette problématique en trois points :

- une virtualisation complète sur le data-center avec Virtual Datacenter Operating System (disponible courant 2009)
- les extensions propres à l'offre "cloud privé"
- l'administration des couches du cloud

Le " Cloud privé " ou " Private cloud " se définit tout d'abord par une idée simple : disposer d'un environnement sécurisé avec les capacités du *Cloud*, que celui-ci soit interne ou externe. Il doit être flexible, et capable d'assurer la disponibilité de toutes

les applications nécessaires et de le déléguer sur le nuage informatique. VMware ne voit que des avantages à ce nouveau nuage : améliorer la consommation des serveurs, faire du " on demand " directement dans l'entreprise, être plus flexible. VMware mise bien entendu sur son initiative vCloud pour fournir les briques à son " Private cloud ". La bonne exécution des applications et l'interopérabilité sont des éléments vitaux et l'éditeur cherche à rassurer. C'est pour cela que l'on dispose de bibliothèques de développement vCloud. Cependant, on peut tout de même se poser des questions sur la pertinence du " Cloud privé ", surtout si on le compare à l'intranet actuel. N'est-ce pas retomber dans les vieux démons du réseau local ? Est-ce toujours du *Cloud* ?

agenda \

MARS

Le 31 Mars, IBM Forum Paris - La Défense, **VUME 2009** Ilog, une société du groupe IBM, organise le " Visualization User Meeting " <http://www.ilog.com/vume09/>

AVRIL

Du 31 mars au 2 avril, Paris Expo Porte de Versailles - Hall 2.2 : **Solutions Linux Open Source**, 10e édition de l'événement européen de référence sur le marché Linux et du Logiciel Libre <http://www.solutionslinux.fr/main.php>

• Hall 8 : RTS Embedded Systems 2009

Le salon des solutions informatiques temps-réel et des systèmes embarqués. <http://2009.salon-rtts.fr/>

• Le 2 Avril, Aix en Provence

Salon les Jeudis.com dédié au recrutement des ingénieurs et des informaticiens [Http://www.lesjeudis.com](http://www.lesjeudis.com)

• 07 avril 2009, Paris : Maîtriser les projets par la gestion des exigences

Spirula et Objecteering Software vous convient, de 9h00 à 12h30, à un séminaire gratuit sur le thème :

Le CMMI appliqué au MDA.

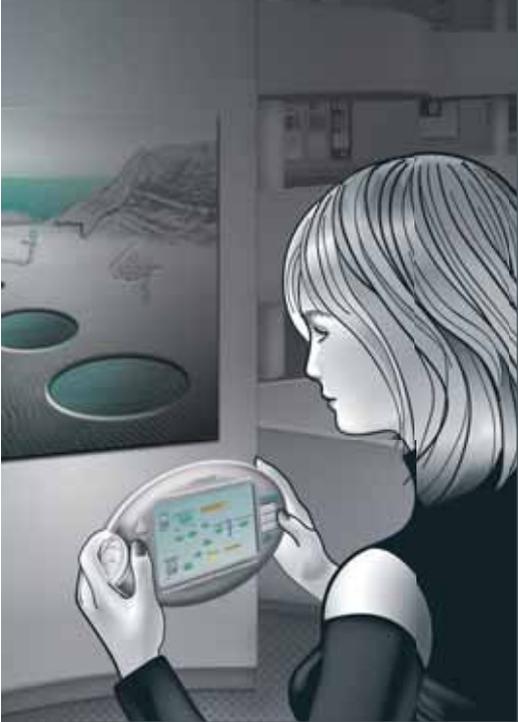
http://www.objecteering.fr/sign_up.php?sem=spirula

• Du 24 mars au 28 avril, **TDF Tech 2009** PC Soft organise son tour de France technique avec 21 thèmes et 69 sujets. www.pcsoft.fr

MAI

Le 13 mai 2009, CNIT, Paris La Défense

- Virtual forum
- Solutions IT On Demand
- Management de Projet www.virtualforum.fr



Visualisez le fonctionnement interne des logiciels grâce aux différents outils Altova®



Altova MissionKit® est une suite intégrée d'outils UML, XML et d'intégration de données pour les créateurs de logiciels modernes.



Altova MissionKit comprend plusieurs outils pour les créateurs de logiciels:

UModel® – l'outil UML pour la modélisation de logiciels

- Prise en charge de tous les types de diagrammes UML, plus BPMN
- Ingénierie inverse et génération de code en Java, C#, VB.NET

XMLSpy® – l'éditeur et environnement de développement XML

- Prise en charge de toutes les technologies de type XML
- Royalty-free Java, C#, C++ code generation

MapForce® – l'outil graphique de mappage de données

- Mappage entre XML, bases de données, EDI, fichiers plats, XBRL, Services Web
- Génération de code Java, C# et C++ libre de droits

Et jusqu'à cinq outils supplémentaires...

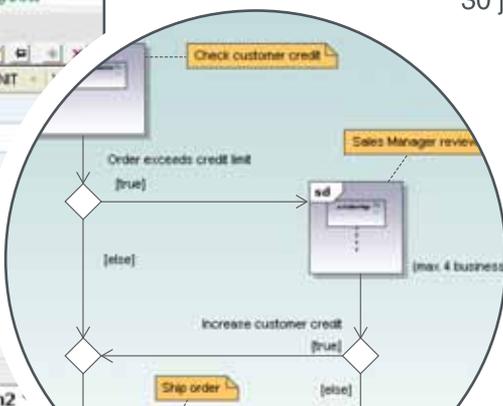
Nouveautés dans la version 2009:

- Génération de diagrammes de séquences UML
- Prise en charge de la validation, de l'édition de la taxonomie et du mappage XBRL
- Prise en charge du mappage HL7



 Téléchargement gratuit!

Essayez avant d'acheter avec une version d'essai gratuite et entièrement fonctionnelle de 30 jours disponible sur www.altova.com.



■ **Soyatec** vient de dévoiler la version 1.0 du plug-in eclipse4sl. Il s'agit d'un environnement de développement sous Eclipse pour Silverlight. Cette version complète les fonctions déjà disponibles dans les versions bêta. L'interopérabilité avec Java est améliorée.

■ **Serena** a annoncé plus d'un million de téléchargements pour son outil de gestion de projet open source et gratuit : OpenProj. Il s'agit de l'outil le plus populaire après Microsoft Project. Les utilisateurs américains sont largement majoritaires. Le succès rencontré auprès des utilisateurs d'OpenProj est dû à la fois à son téléchargement très rapide et à un accès immédiat à tous les dossiers Microsoft Project existants, sans interrompre son travail. La migration est immédiate et n'entraîne aucun bouleversement.

■ **Red Hat** confirme son positionnement sur la virtualisation depuis le rachat de KVM. L'éditeur annonce Red Hat Enterprise Virtualization. Cette gamme s'appuie sur 4 offres : Red Hat Enterprise Linux supportera KVM mais les utilisateurs Xen resteront supportés avec Red Hat Enterprise Linux 5. Virtualization Manager for Servers se concentre sur l'administration côté serveur. Virtualization Manager for Desktop concerne le poste de travail. Virtualization Hypervisor est l'hyperviseur Red Hat. La gamme se complètera dans les 18 prochains mois.

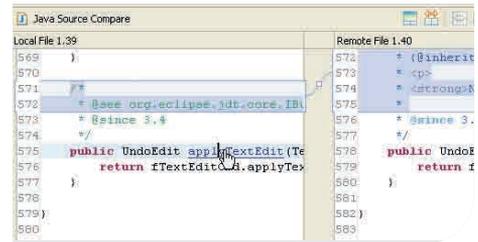
■ **NetBeans 6.7** arrivera chez les développeurs le 22 juin prochain. Cette version apportera de nouvelles modifications sur les éditeurs, le module de debug, le support des langages dynamiques, la mobilité. Sur ce point, un effort important sera fourni avec un nouvel éditeur, un support SVG étendu. A noter que la 6.7 était anciennement nommée 7.0. En attendant c'est la v6.5.1 qui a été livrée mi-mars avec quelques corrections de bugs et l'ajout par défaut de patches.

IDE

Eclipse e4 : un IDE 100 % renouvelé ?

Que devient le projet Eclipse e4 ? Depuis quelques semaines, les premières build sont disponibles, même s'il faut toujours télécharger les composants propres à e4 et les intégrer à la main. La version m1 est sortie début février. Elle est basée sur Eclipse 3.5, version elle aussi en développement. Un des chantiers les plus importants est l'interface déclarative via le projet XWT.

Le changement profond dans les interfaces et les bibliothèques d'interfaces impactera les développements actuels. C'est pour cela qu'une couche de compatibilité sera disponible afin d'assurer la migration plus en douceur des projets Eclipse 3.x vers e4. E4 doit aussi mieux prendre en compte le développement web et notamment sur le côté RCP (Rich Client Platform). Mais e4 doit surtout apporter une unification de la plate-forme : sur les projets et les API. Il s'agira aussi de pouvoir utiliser Eclipse directement dans un navigateur et plus uniquement localement. SWT connaîtra une profonde révision avec des widgets pouvant utiliser les technologies actuelles comme Ajax, Flex, Silverlight. Vous l'aurez compris, les responsables e4 veulent rendre Eclipse plus convivial, plus simple, plus



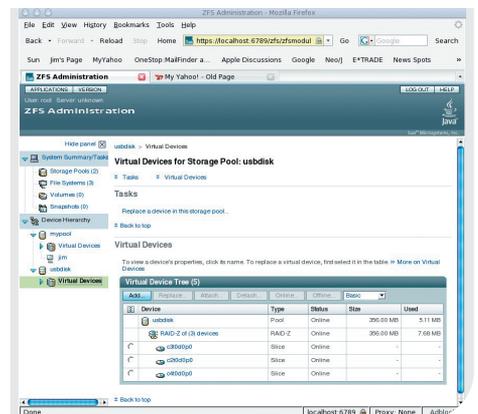
ergonomique. Car l'empilement des projets et des versions a alourdi l'IDE. Il s'agit de rendre de nouveau attractif Eclipse à tous les développeurs. A noter que le français Soyatec travaille avec la fondation sur XWT. Côté planning, la première version technique utilisable devrait sortir cet été. La version finale n'est pas prévue avant mi-2010.

En attendant, le développement de la v3.5 touche à sa fin. SWT continue d'évoluer, notamment sur MacOS X qui s'appuie sur un portage Cocoa (idem pour OpenGL). Ensuite, pas mal de petites modifications sur l'interface, les fonctions (texte, recherche en particulier). Le debug subit aussi quelques modifications avec la vue "breadcrumb". On dispose de nouveautés sur Equinox (OSGi) et lié à cela, de nouvelles API ont été implémentées dans le core Framework.

Stockage

Que devient ZFS chez Sun ?

Aujourd'hui, tout le monde sait que la solidité du système de fichiers est vitale. Nous n'avons pas toujours le réflexe ZFS. Graham Lovell (directeur exécutif Storage Servers Product Line chez Sun) parle de l'avenir de ZFS sur le marché. " Avec des budgets sous pression, ZFS est un système extrêmement attractif pour remplacer les solutions Symantec / Veritas sur les datacenter. La réduction des coûts de licences est la première raison mise en avant pour les administrateurs systèmes quand ils s'intéressent à ZFS. Il est gratuit. ZFS est le système de fichier par défaut de Solaris et c'est une pièce de la gestion de données. Il est pris en compte par les grandes marques de machines x86, Dell, IBM, HP, Sun et Fujitsu. Tous livrent et supportent Solaris. ZFS est aussi disponible sur d'autres systèmes comme MacOS X et BSD. ZFS est un thème de plus en plus populaire dans la communauté OpenSolaris et il est livré sur les Unified Storage 7000 de Sun. Pen-



dant que d'autres systèmes de fichiers, comme ReiserFS sur Linux, sombrent dans l'obscurité. Le choix d'un fichier de système prend du temps car il touche directement aux parties sensibles de l'entreprise : les données. D'autre par ZFS est le seul système de fichier pouvant traiter les données de demain, typiquement les challenges de l'informatique HPC (ou High Performance Computing, ndr)."



Microsoft® Visual Studio® 2008

Montez en gamme à tarif réduit!

jusqu'à
30%
de remise



Vous possédez un outil de développement de la gamme Microsoft Visual Studio ?

4 promotions exceptionnelles

Si vous possédez Visual Studio Professionnel, Visual Studio avec MSDN ou encore Visual Studio Team System, vous avez la possibilité d'accéder au produit supérieur de la gamme à un prix préférentiel.



Visual Studio Professionnel



Visual Studio avec MSDN



Visual Studio Team System



Visual Studio Team Suite

Vous profitez ainsi à prix réduit (selon votre produit) :

- Des serveurs Microsoft
- Des systèmes d'exploitation Microsoft
- De logiciels de productivité : Microsoft Office, Project, Visio...
- D'outils de test : Montée en charge, fonctionnels, manuel...
- Logiciels de modélisation : diagrammes d'applications, de développement...
- Jusqu'à 4 incidents de support technique,
- Ressources techniques
- Toutes les mises à jour pendant votre abonnement : Visual Studio 10 et Windows 7 dès leur disponibilité

Exemple : Visual Studio 2008 avec MSDN Premium : 1137 € HT / an / par dev**

*Valable jusqu'au 26 juin 2009 - Tous les détails des promotions sur notre site internet.
**(abonnement 2 ans obligatoire - licence Open Business)

au lieu de ~~1395 € HT~~

COMSOFT direct

Bechtel's Software Specialist

SOS DEVELOPERS

Microsoft GOLD CERTIFIED

Partner

1^{er} revendeur en France en outils de développement Microsoft.

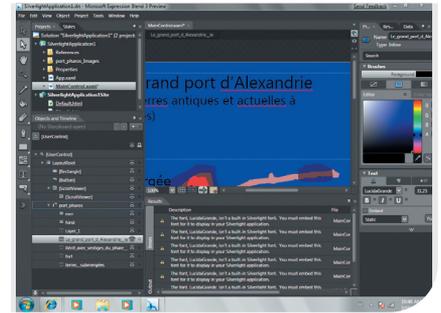
Contactez nos spécialistes MSDN au 04 97 21 58 65

msdn@comsoft.fr – www.sosdevelopers.com/msdn3.htm

Mix 2009

Microsoft dégage Silverlight 3, Blend 3 et Azure !

Du 18 au 20 mars dernier à Las Vegas, Microsoft organisait sa grand' messe des technologies web. L'éditeur a dévoilé la pré-version de Silverlight 3.0 disponible sur Windows et MacOS X. Sans oublier Internet Explorer 8 en version finale. Pour Microsoft, le design devient crucial ! Revenons sur les principales annonces.



Comme attendu, Silverlight 3.0 fait son apparition en bêta, à peine 6 mois après la sortie de la version finale de Silverlight 2. Les grosses nouveautés de cette future version sont :

- meilleure gestion de l'audio et vidéo, surtout en haute définition
- support de nouveaux codecs particulièrement attendus : H.264, Mpeg-4, AAC
- nombreuses améliorations fonctionnelles (caching, deepzoom, installation simplifiée), en tout l'éditeur annonce une cinquantaine de nouveautés et d'améliorations.
- Mécanismes de recherche dans le contenu
- Intégration possible de Virtual Earth dans une application Silverlight 3
- Développement multitouch (basé sur l'API de Windows 7)
- Support étendu et facilité des données (c'était une des lacunes jusqu'à présent de la technologie).
- La possibilité de créer des interfaces 3D
- Accélération matérielle par la GPU
- Autre nouveauté (un peu) surprise : Windows Live Messenger Web Toolkit. Grâce à ce kit de développement, il sera possible d'intégrer Messenger dans un site !

Silverlight sort du navigateur !

Surtout, cette mouture va gérer le mode déconnecté. Si à l'heure où nous écrivons nous n'avons pas pu avoir plus de détails, il s'agit de pouvoir utiliser une application Silverlight directement sur son bureau. Il s'agira d'une application Windows comme une autre, sans l'accès aux fonctions systèmes, au contenu des disques durs, ce qui la distingue d'Adobe Air. Dans l'esprit, cela se rapproche d'un Mozilla Prism. On disposera, cependant, localement d'un runtime Silverlight pour exécuter les applications. Restent à voir les interactions possibles avec le système et les modes de synchronisations en déconnecté / connecté. Autre grosse nouveauté, Soyatec qui crée des outils de développement Silverlight pour Eclipse, a dévoilé la version MacOS X de son projet Eclipse4SL ! C'est une grosse avancée pour tenter de rattraper Adobe qui est disponible depuis bien longtemps sur Mac ! Nous avons rencontré quelques instabilités sur les vues XAML, mais le projet est plus que prometteur !

Il s'agit de pouvoir récupérer directement dans Blend les dessins maquettes d'un projet web, d'une application. Dans une seule et même application, le designer pourra réaliser son maquetage en partant de rien, dessiner ses esquisses, les mettre en forme, les simuler, leur donner vie pour enfin les déployer...Sketchflow fait partie d'Expression Blend. Impressionnant ! On notera aussi le support de l'intellisense dans XAML, C# et VB. Autre nouveauté : Expression Web Superpreview. Cette nouvelle édition de l'IDE de développement web permet de tester directement dans l'outil, l'affichage de sa page, de ses sites, sur les différents navigateurs du marché (IE, Chrome, Safari, Firefox). Enfin une vraie fonction debug multi-navigateur dans Expression Web !

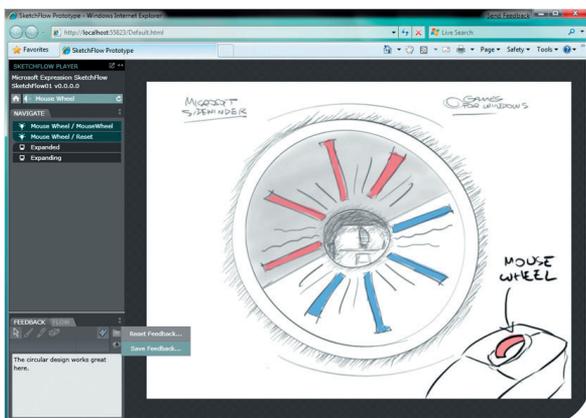
Windows Azure : 2009 !

Si Azure n'a pas tenu la vedette durant la première journée, plusieurs annonces importantes :

- l'utilisation commerciale d'Azure sera disponible courant 2009 : c'est une réelle surprise.
- Disponibilité de la CTP 2 d'Azure dès maintenant.

Le développeur peut désormais coder des applications avec un niveau élevé de sécurité (full trust) et le développement natif est possible. On dispose du support de FastCGI et de la géolocalisation permettant de choisir le Data-center de stockage. Compte-rendu et analyse complets dans le prochain numéro. Et sur www.programmez.com

■ François Tonic



Expression 3.0 : toujours plus pour les designers

Tout naturellement, Microsoft dévoile une partie de la v3 d'Expression. Blend 3 va encore améliorer le flux de travail entre le développeur et le designer. Cela passe par le support natif des formats Photoshop et Illustrator ! C'est bluffant à l'usage. On récupère très bien les calques et les styles des fichiers PSD !

Autre grosse nouveauté le *Sketchflow*.

■ **MLState** et **Epitech** viennent de signer un partenariat. Le but est que l'école en informatique puisse exploiter OPA, un nouveau langage. OPA doit rendre plus simple la conception d'applications SaaS en fusionnant serveur web, SGDB, langage applicatif. Des projets seront développés autour d'OPA dans les prochains mois.

■ Le langage **Vala** a son IDE : Val(a) IDE. Créé par Nicolas Joseph, cet IDE doit permettre de coder plus rapidement avec le langage Vala, un langage de programmation que l'on retrouve dans Gnome pour développer plus rapidement les applications. La version 1.0 devrait sortir rapidement. La version actuelle de l'IDE propose la coloration du code, la compilation, une architecture plug-in. Sites : <http://www.valaide.org/doku.php?id=:fr:start> et <http://live.gnome.org/Vala>

■ **Nat System** a lancé NatJet 2.0 il y a quelques semaines. Cette version renforce la productivité de développement d'une application JEE. Pour arriver à ce résultat, l'outil permet de créer une perspective Eclipse NatJet pour retrouver les vues Eclipse dans ces projets, une nouvelle organisation des panneaux et fenêtres, un nouvel assistant de création de projet. L'internationalisation est une des grandes nouveautés de cette v2, sans oublier l'arrivée de nouveaux composants graphiques. Site : <http://www.natsystem.fr>

Concours

Blu Age élu par Logica pour son concours international de l'innovation

Le groupe Logica, un groupe international majeur des services informatiques, a sélectionné Netfective et Blu Age, comme le " champion " français, parmi les quarante PME postulantes en France à son concours international de l'innovation. Un des huit finalistes (France, Pays-Bas, UK, Suède, Danemark, Portugal et Inde) a été sélectionné par le jury international d'experts de Logica. " Netfective nous a convaincu en proposant une réponse aux préoccupations majeures de nos clients que sont la réduction des coûts et la réactivité. Leur approche globale du MDA et leur expertise technique pointue nous laissent penser qu'il s'agit d'une innovation concurrentielle pour plusieurs de nos clients dans les années à venir. Logica partage en outre avec Netfective une démarche commune reposant sur la compréhension préalable des enjeux métier de nos clients allée à la recherche de performance et au goût pour l'innovation au sens de la capacité à challenger l'existant " déclare Ivan Bertrand, directeur chez LogicaManagement Consulting, en charge de l'innovation et Président du jury.



Objecteering

Your projects deserve a tool *

La convergence SOA, UML2, BPMN, EA, pour le développement guidé par le modèle

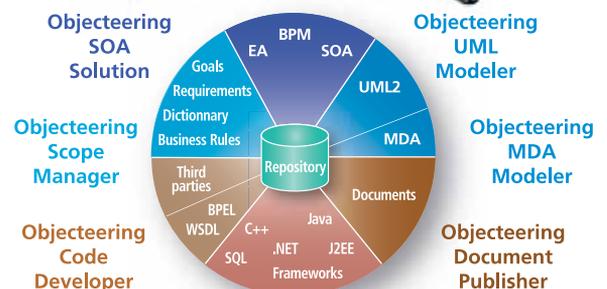
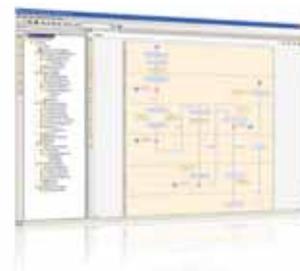
- outil **intégré** assurant la modélisation complète des applications et systèmes,
- **facile** à utiliser : à chaque acteur, à chaque phase, la représentation dédiée,
- **alignement métier** du SI grâce au référentiel partagé et la traçabilité totale entre toutes les vues et phases,
- haute **productivité** grâce à la génération automatique de la documentation et du code.

* Vos projets méritent un outil.



www.objecteering.com

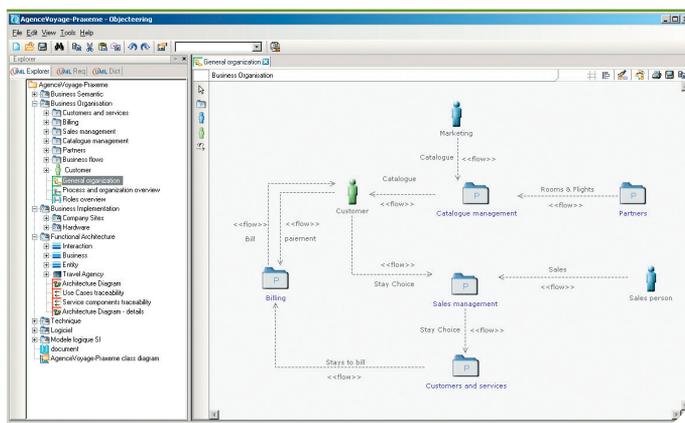
Tél. : 01 30 12 16 60 - sales@objecteering.com



Objecteering
SOFTWARE
The model-driven development company

Modélisation : marché et tendances

La modélisation a le vent en poupe en ce moment et pas uniquement le Model Driven. L'arrivée prochaine de Microsoft sur ce marché avec le support d'UML (même partiel) et surtout le futur Oslo, la plate-forme transversale de modélisation de l'éditeur va-t-elle bouleverser les éditeurs d'outils et dynamiser le marché ? Les avis divergent.



Aujourd'hui, les tendances de la modélisation se résument à :

- la modélisation de type UML via des diagrammes
- les mouvances Model Driven : le modèle pilote le développement, le test, etc.
- le DSL (Domain Specific Language)

L'arrivée de Microsoft dans l'antre de l'OMG, qui s'occupe d'UML et de ses évolutions, a fait frémir quelques éditeurs, dont IBM, même si officiellement tout va bien et que l'on félicite Microsoft de son ralliement. Il faut avouer que cette arrivée n'est pas anodine pour les éditeurs du secteur, notamment sur plate-forme Windows, car jusqu'à présent, Microsoft ne proposait rien de consistant hormis des outils DSL. Avec un support partiel, dans un premier temps, d'UML dans la gamme Visual Studio, la concurrence deviendra particulièrement rude.

La crise a-t-elle un impact sur le marché de la modélisation ?

" Crise ou pas, le modèle permet d'améliorer la qualité, d'être plus efficace et " time to market ", avec un

gain de temps sur les phases amont. " commente **Yan Chronberg** (directeur commercial Objectteering Software).

Le modèle peut donc aider l'entreprise. Cela passe notamment par la disponibilité d'outils capables de modéliser l'ensemble du SI, bref à avoir une vue globale. Et cela permettra aussi de mettre en place un référentiel répertoriant l'ensemble des modèles et d'y collecter les informations adéquates (le type, la sémantique, etc.). Les défis actuels que Yan Chronberg pointe du doigt sont les suivants :

- capacité à interconnecter différents modèles
- définir les règles
- assurer et contrôler la transformation
- distinguer chaque modèle

Tout cela permet d'établir une cartographie sémantique métier, de l'architecture logique et ainsi de définir les objectifs à atteindre et les transformations entre le code et le modèle à réaliser.

Le DSL une bonne réponse ?

" Les DSL ont un intérêt. Microsoft en est un bon exemple, et l'éditeur vient de rallier l'OMG. UML a l'avantage d'être plus universel " précise Y. Chronberg. La question est de savoir si UML est suffisamment riche pour répondre à des besoins spécifiques. La réponse est clairement non, même si UML est extensible par des profils. C'est la force du DSL que d'être spécifique et de répondre à un besoin précis. L'un des problèmes principaux des DSL était un certain manque d'outils

ou tout du moins des outils peu maniables. UML et DSL vont inévitablement se marier pour offrir un spectre fonctionnel plus large.

L'arrivée de Microsoft sur le marché UML

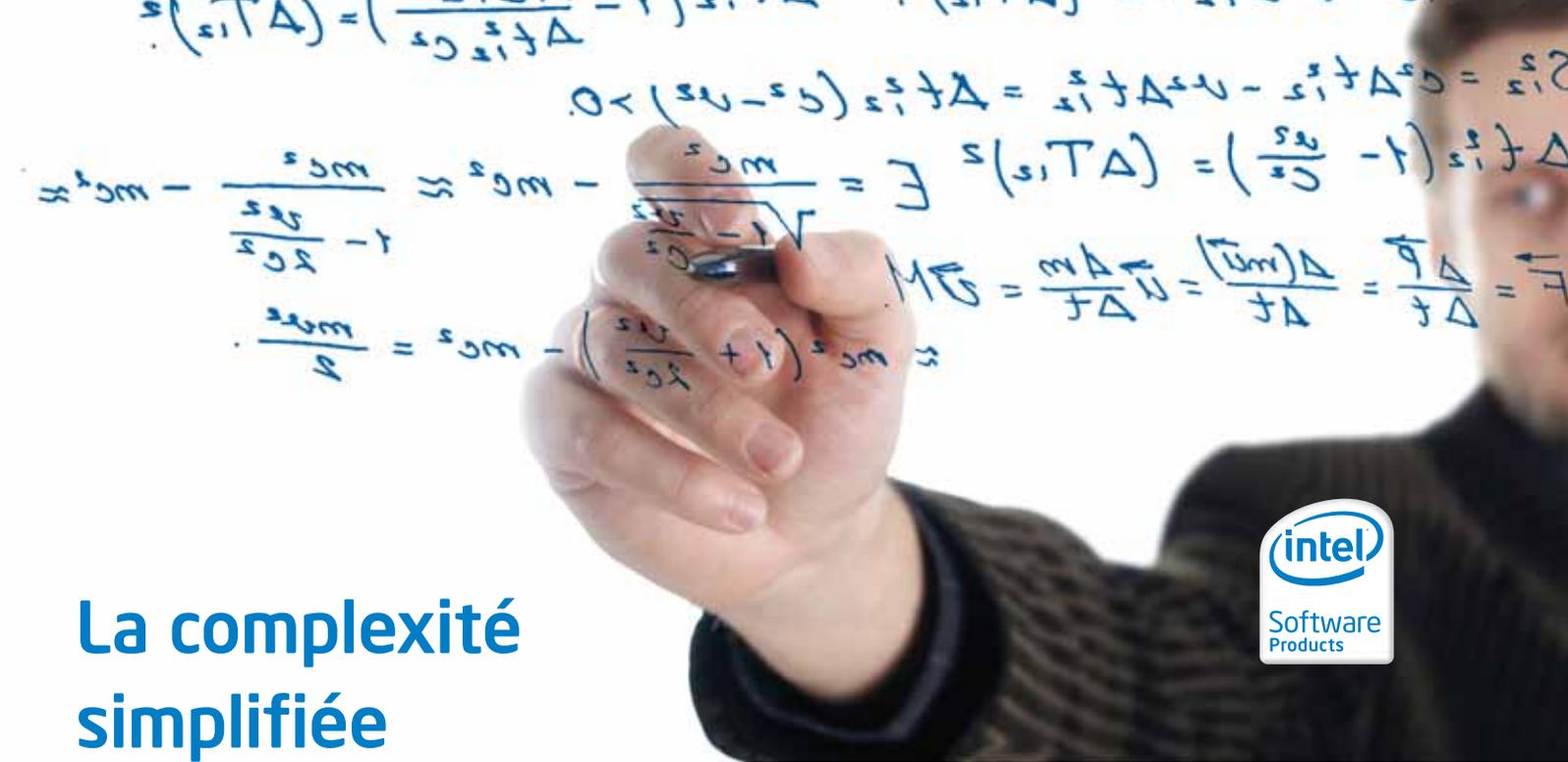
" Il y a à la fois de la crainte et un côté positif. Cela peut aider UML à être davantage adopté ", recadre Yan Chronberg. Quoiqu'il en soit, l'arrivée prochaine de l'éditeur est regardée avec prudence par la concurrence. Si aujourd'hui le marché de la modélisation Windows est occupé par plusieurs acteurs, il est certain que l'outillage de Microsoft peut modifier à terme la donne, notamment avec la disponibilité de fonctions de modélisation dans Visual Studio. Dès à présent, les éditeurs doivent se préparer à ce petit *big bang*. Mais d'autre part, pour Microsoft, UML, même s'il a de l'intérêt est encore trop limité, trop complexe à appréhender pour un développeur. Cela va être intéressant de voir comment l'éditeur va et veut étendre le modèle actuel. L'une des pistes, comme nous le verrons plus loin, est le modèle exécutable prôné dans Oslo. L'exécution d'un modèle peut se révéler particulièrement intéressant côté serveur par exemple. Et permettrait des ajustements très rapidement en modifiant le modèle. L'autre question est de savoir si la vogue actuelle autour du SaaS, du service en ligne va aussi toucher la modélisation. La réponse est oui. Objectteering travaille sur le sujet même s'il reste l'écueil du modèle économique.

■ Francois Tonic

Modeliosoft : nouvel éditeur UML

Un nouveau venu dans le marché UML, BPMN et MD : Modeliosoft avec la solution Modelio. L'outil supporte la génération vers Java, .Net et C++. Trois versions sont disponibles : Modelio Free Edition, Express Edition (100 euros) et Enterprise Edition (plus de 300 euros). Cette dernière version devrait être disponible lorsque vous lirez ces lignes. Avec Modelio, l'objectif est d'adresser le développeur, la petite entreprise.

www.modeliosoft.com



La complexité simplifiée

Compilez et personnalisez pour le traitement haute performance. Créez le logiciel le plus rapide possible et utilisez les dernières technologies dans les architectures Intel.

NOUVEAU ! Intel® Compiler Suite 11 Professional Edition pour Windows* et pour Linux. Contient les compilateurs Intel C++ et Visual Fortran, Math Kernel Library (Intel MKL), Intel Integrated performances (Intel IPP) et Intel Threading Building Blocks (Intel TBB).



NOUVEAU! Intel® Cluster Toolkit Compiler Edition 3.2. Fournit un package de logiciels contenant les compilateurs Intel C++ et Fortran pour toutes les architectures Intel. PLUS tous les outils Cluster Intel qui vous aident à développer, analyser et optimiser la performance de vos applications parallèles sous clusters Linux ou Windows Compute Cluster Server (CCS)

Intel® Math Kernel Library est une bibliothèque de traitement numérique hautement optimisée pour les applications mathématiques, scientifiques, engineering et financières

Intel® MPI Library fournit une implémentation flexible de MPI pour un développement message-passing interface plus aisé sur des architectures multi réseaux.

‘Dans une dizaine d’années, un programmeur qui ne pensera pas d’abord “parallèle” ne sera pas un programmeur.’

James Reinders
Chief Software Evangelist
of Intel Software Products

WWW.SOSDEVELOPERS.COM/INTEL.HTM

- Intel Compiler Suite Windows ou Linux + maintenance 1 an: 733 € ht
- Intel Cluster Toolkit Compiler Edition + maintenance 1 an: 880 € ht
- Intel Math Kernel Library Windows + maintenance 1 an: 293 € ht
- Intel MPI Library + maintenance 1 an: 339 € ht



Plus d’informations sur : www.sosdevelopers.com/intel.htm
Tel : 0825 07 06 07 ▪ infos@comsoft.fr

Oslo La future plate-forme de modélisation de Microsoft

La CTP (Community Technical Preview) d'Oslo disponible depuis novembre 2008 a été la première concrétisation de l'investissement fait par Microsoft dans le développement d'applications, piloté par les modèles. L'idée fondamentale est d'offrir un espace de modélisation permettant de fédérer l'ensemble des modèles existants propres à des domaines spécifiques (processus métier, bases de données, processus SharePoint...) et offrant aux architectes, analystes, développeurs, la possibilité de définir de nouveaux modèles.

L'ambition de Microsoft est de mettre les modèles au cœur de la conception, du déploiement et de la supervision des applications .NET et d'offrir ainsi un référentiel commun à l'ensemble des acteurs impliqués dans la gestion du cycle de vie logiciel : analystes métiers, architectes, exploitants... Ces modèles ne sont plus statiques mais pilotent l'ensemble des briques logicielles de la plate-forme .NET (ASP.NET, WCF, WF...), ils ne sont plus spécifiques et isolés mais connectés et partagés. Ainsi, les différents acteurs disposent d'une meilleure visibilité sur les applications .NET, ces dernières gagnent en flexibilité et globalement la productivité des développements s'en trouve améliorée. [Fig.1] Oslo, nom de code de la future plate-forme de modélisation de Microsoft, est composé de trois briques principales :

- Un langage dénommé " M " pour la définition des modèles et des langages textuels spécifiques (DSL textuels)
- Un outil de modélisation nommé " Quadrant " permettant de définir et d'interagir avec les modèles au tra-

vers d'une interface graphique évoluée

- Un référentiel supportant l'ensemble des modèles et les rendant disponibles pour les outils et les composants de la plate-forme

Le langage " M "

A l'origine Oslo ne comprenait qu'un référentiel de modèles et un outil graphique de modélisation permettant leur création et leur manipulation. Les équipes Microsoft ont rapidement conclu que le succès d'Oslo passerait par la mise à disposition d'un langage permettant aux utilisateurs de définir et de manipuler leurs modèles. Ainsi est né le langage " M ".

" M " est un langage déclaratif pour définir des modèles de domaine et des langages textuels de domaines spécifiques comprenant :

- **MGraph** pour sérialiser les données propres à chaque modèle sémantique sous la forme de graphes avec une syntaxe similaire à JSON. Ce sont les instances des modèles.
- **MSchema** pour la définition des modèles sémantiques construits sur les trois concepts de base valeurs, types et extents définis comme suit :

- Une valeur est simplement une donnée qui respecte les règles du langage " M ".

- Un type définit un ensemble de valeurs. Le langage " M " n'est pas orienté objet et propose à l'inverse un système structurel de types pour lequel la compatibilité et l'équivalence de types est déterminée par la structure et non par des déclarations explicites.

- Un extent fournit un stockage dynamique pour les valeurs.

Le choix de SQL Server pour la persistance des modèles induit une forte affinité de MSchema avec le modèle relationnel. Il a été dicté par la volonté de capitaliser sur les investissements réalisés par Microsoft dans le domaine de la base de données et du décisionnel. Une fois les modèles définis et déclarés dans le référentiel, les données MGraph peuvent être interrogées via le langage de requête fourni par " M " similaire à LINQ. Les fichiers MSchema ont l'extension .m.

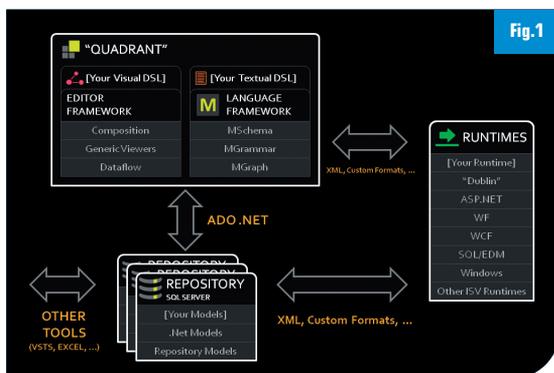
- **MGrammar** pour créer des parseurs GLR à même de peupler les modèles sémantiques précédemment définis. Les flux de texte sont ainsi transformés en instances MGraph. Les fichiers MGrammar ont l'extension .mg.

Le SDK d'Oslo fournit l'éditeur de texte Intellipad supportant l'IntelliSense pour faciliter la création des modèles, la validation des instances MGraph au regard de leur modèle et la définition des grammaires de domaines spécifiques.

Afin d'illustrer les concepts clés du langage " M ", nous avons défini un modèle sémantique pour modéliser les cercles. Un cercle se définit par ses coordonnées et son rayon. [Fig.2] Notre modèle est peuplé soit par la définition d'une instance MGraph en langage " M ", soit par l'analyse du fichier texte en se basant sur la grammaire de notre exemple.

Le SDK d'Oslo fournit un compilateur MSchema nommé **m.exe** qui génère

Architecture Oslo.



Développez 10 fois plus vite

WEBDEV

Ajax en 1 clic

WEBDEV 14 est l'environnement de développement professionnel qui permet de développer jusqu'à 10 fois plus vite tous les types de sites et d'applications reliés aux données de votre entreprise.

Le **WEB 2.0** est facile: l'activation d'**AJAX** dans vos sites s'effectue naturellement: un simple clic dans l'éditeur de code indique que le code à exécuter est de type «**Ajax**». **WEBDEV 14** est certainement le seul environnement au monde à proposer autant de souplesse et de puissance.



Sous l'éditeur de code de **WEBDEV 14**: un clic et le traitement programmé devient «**Ajax**»

WEBDEV 14 est un **AGL complet**: langage L5G, génération de code **PHP**, débogueur, Web Services, gestionnaire de sources, installateur, base de données SQL intégrée et lien avec toutes les bases du marché, composants, éditeur d'états PDF et code-barres, règles métier, dossier, outils de déploiement et d'administration..., tout est inclus, en français.

*Vous aussi réalisez vos sites **WEB 2.0** 10 fois plus vite... avec **WEBDEV 14**.*

UN OCEAN DE WEB 2.0

UN CODE MULTI-PLATEFORME :
Windows, .Net, Java, PHP, J2EE, XML, Internet, Ajax, Pocket PC, SmartPhone, Client riche ...



DEMANDEZ LE DOSSIER GRATUIT

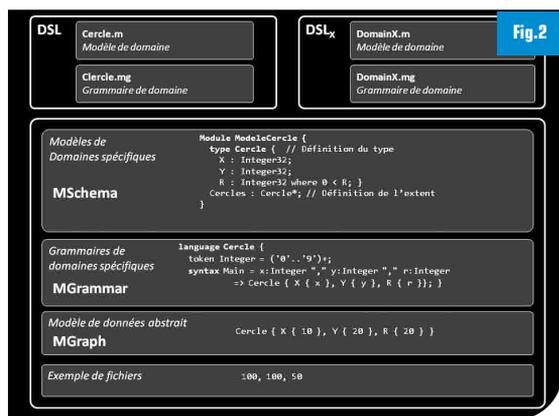
252 pages + DVD + 112 Témoignages.
Tél: **04.67.032.032** ou **01.48.01.48.88**
info@pcsoft.fr

Fournisseur Officiel de la Préparation Olympique

www.pcsoft.fr



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



Modèle Cercle. à partir d'une définition M le script SQL correspondant ou une image du modèle au format OPC (Open Packaging Conventions) avec l'extension **.mx**. L'utilitaire **mx.exe** publie à partir de l'image générée le modèle correspondant dans une base de données SQL ou directement dans le référentiel d'Oslo. Concernant les grammaires de domaines spécifiques, l'utilitaire **mg.exe** génère à partir d'un script de grammaire M une image au format OPC avec l'extension **.mgx**. Le parseur **mgx.exe** crée à partir de cette image et d'un fichier texte les instances MGraph correspondantes.

XAML

Le format pivot retenu pour la définition des modèles est XAML (Extensible Application Markup Language), langage basé sur XML, initialement utilisé pour la définition des interfaces graphiques WPF et des workflows de WF. Dans la version 4.0 du Framework .NET, les différentes versions des moteurs XAML, WF et WPF, ont été unifiées pour ne faire plus qu'un. Ces nouvelles API étendues sont dorénavant utilisées par toutes les briques du framework .NET (WCF, WF, WPF) et par Oslo.

Quadrant, l'éditeur de modèles d'Oslo

L'outil de modélisation Quadrant est le moyen le plus convivial pour manipuler les modèles au travers d'une interface graphique basée sur Windows Presentation Foundation. Il met à disposition de ses utilisateurs des espaces de modélisation personnalisés. Ainsi, les analystes peuvent manipuler des modèles orientés métiers d'assez haut niveau alors que

les profils plus techniques descendront dans le détail au travers de vues spécialisées : diagrammes de conception, topologie de déploiement...

Quadrant fournit un ensemble de modèles prédéfinis (modélisation métier, processus...) et permet également de créer ses propres modèles. Dans un premier temps, il faut définir son modèle avec le langage " M " et le publier dans le référentiel au travers de scripts SQL générés automatiquement. Ensuite, Quadrant permet de créer graphiquement des instances de ce modèle et valide les données saisies au regard des contraintes définies.

Quadrant s'appuie sur un ensemble extensible de viewers (table, liste, diagramme...) et de designers (processus métier, applications composites, données...) pour représenter et manipuler les données modélisées dans le référentiel Oslo. Ainsi, une collection d'entités peut être visualisée sous forme d'un tableau ou d'une liste graphique. Il est également possible de manipuler les dépendances entre entités au travers de vues maîtres/détails ou de diagrammes. L'utilisateur peuple ses espaces de travail avec des représentations graphiques personnalisées de ses données et navigue facilement d'une représentation à l'autre en fonction de son axe d'analyse. Quadrant propose également un explorateur permettant de naviguer dans les modèles du référentiel et leurs instances. L'utilisateur glisse et dépose une instance sur son espace de travail puis choisit la représentation la plus adéquate.

Référentiel des modèles

Les données des modèles sont stockées dans une base de données SQL Server 2008 afin d'offrir un stockage commun à toutes les phases du cycle de vie logiciel répondant aux attentes d'une solution d'entreprise : sécurité, performance, disponibilité, montée en charge.

Le déploiement d'un modèle dans Oslo se traduit par la génération d'un script SQL de création d'ensemble des artefacts SQL correspondants.

Dans notre exemple, les objets SQL sont définis dans le schéma portant le nom du module *ModeleCercle*, les données sont stockées dans la table Cercles et le domaine de valeur du rayon est contrôlé par une contrainte au niveau de la table.

Une fois publiés, le modèle et les instances de notre exemple sont accessibles depuis Quadrant et visualisables au travers des viewers standards. Quadrant offre la possibilité, au travers de son éditeur de modèles, de personnaliser la vue par défaut associée à un modèle, de lui affecter une configuration par défaut... Tous ces paramétrages sont stockés dans le référentiel en tant qu'instances des modèles de configuration Quadrant. Afin d'avoir une vision globale d'une application, toutes les informations de conception et les composants générés lors de la phase de réalisation sont stockés dans le référentiel Oslo. Dans un premier temps, l'application est modélisée dans Quadrant. Ensuite, à partir de ces informations, les composants sont développés dans Visual Studio, puis publiés dans le référentiel Oslo. Pour créer un package de déploiement, les fichiers et ressources sont exportés du référentiel en se basant sur la topologie de déploiement. Le référentiel d'Oslo est une base de données ouverte sans couche d'accès propriétaire et met à disposition ces données aux travers de vues. La lecture ou modification des données se font au travers des mécanismes standard d'accès aux données.

Conclusion

Cette version pré-alpha présente les briques fondamentales d'Oslo - référentiel des modèles, langage et outil graphique de modélisation - et l'approche retenue par Microsoft pour unifier le développement piloté par les modèles de la conception au déploiement. La prochaine CTP aura gagné en maturité et illustrera certainement

le cycle de vie logiciel des applications .NET des prochaines années.



■ **Thomas Marencic**
Architecte - Microsoft Services

WINDEV® Mobile



**DÉVELOPPEZ VOS APPLICATIONS
POUR POCKET PC, TÉLÉPHONE,
TERMINAL MOBILE**



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

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

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

Toutes les fonctionnalités d'un AGL professionnel sont offertes. Tous les aspects de la mobi-

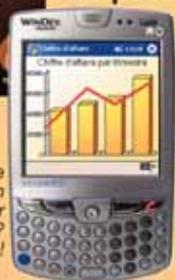
lité sont gérés: accès direct, réplication, WiFi, Bluetooth, 3G, Internet, socket, ActiveSync, réseau, J2EE, SMS, RFID, lien avec votre S.I., **codes-barres automatiques...**

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

(Logiciel **professionnel**)

**VERSION
EXPRESS
GRATUITE**
Téléchargez-la !

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



UN CODE MULTI-PLATEFORME :
Windows, .Net, Java,
PHP, J2EE, XML,
Internet, Ajax, Pocket
PC, SmartPhone,
Client riche ...

DEMANDEZ LE DOSSIER GRATUIT

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

Tél: **04.67.032.032** ou **01.48.01.48.88**
info@pcsoft.fr

Fournisseur Officiel de la Préparation Olympique

www.pcsoft.fr



Hibernate Survival Guide

2^e partie

Ceci est le second volet de la série d'articles expliquant le fonctionnement d'Hibernate et comment l'utiliser au mieux. Le mois dernier, nous avons vu ce qu'était le cache de premier niveau et combien il était fondamental. Ce mois-ci, nous nous attaquons à un autre point épineux : la gestion des liens inverses dans une relation.

Pour une meilleure compréhension, commençons d'abord par quelques définitions :

- Un lien est une information qui permet à partir d'une entité A d'atteindre le ou les entités B liées à A pour une raison donnée (A est propriétaire de B par exemple). Un lien est un attribut appartenant à A. Ce peut être soit une simple référence vers un B (on parlera alors de lien " unique "), soit une collection de B (on parlera alors de lien " multiple "). C'est une notion qui est directement implémentable en Java. Mais pas en Base de données.
- Une relation est un lien libre (c'est-à-dire non asservi) ou un couple de liens asservis. Un lien est asservi à un autre, si leurs valeurs doivent être maintenues en cohérence. Par exemple si A désigne B par le premier lien du couple, il faut que B désigne A par l'autre lien asservi. Une relation n'appartient ni à A, ni à B puisqu'en fait, elle est située entre les deux entités. Une relation est matérialisée en Base de données par une clé étrangère. Elle n'a pas de représentation directe en Java. Il existe trois types de relations : les relations 1 pour 1, 1 pour N et N pour N.
- Un lien L1 est dit inverse de L2 si L1 et L2 sont asservis, c'est-à-dire s'ils forment, ensemble, une relation. Notons que si L1 est inverse de L2, L2 est obligatoirement inverse de L1. Un lien ne peut être inverse que, d'au plus, un autre lien.

Mapper les relations

Une relation (c'est-à-dire un lien libre, ou un couple de liens asservis) est mappée sur une clé étrangère (relation 1-1, 1-N), sur l'égalité des identifiants (relation 1-1) ou à l'aide d'une table intermédiaire (relation 1-N ou N-N). Pour chaque type de relation (sauf N-N), il existe donc plusieurs stratégies qui toutes offrent avantages et inconvénients.

Leur étude fine ne sera pas abordée dans cet article. Avec Hibernate, le mapping est précisé au niveau des liens : il faut poser une annotation sur chaque lien qui forme une relation. Hibernate (JPA en fait) propose quatre annotations :

- **@OneToOne** qui indique qu'une entité A (possédant le lien annoté) désigne au plus une entité B et que cette entité B n'est attachée qu'à cette entité A. Cette annotation doit donc être posée sur un lien unique (c'est-à-dire sur un attribut de type référence vers B)
- **@OneToMany** qui indique qu'une entité A (possédant le lien annoté) désigne une ou plusieurs entités B, mais qu'un B ne peut être attaché qu'à un seul A. Cette annotation doit donc être posée sur un lien multiple (c'est-à-dire sur un attribut collection de références vers B)
- **@ManyToOne** qui indique qu'une entité A (possédant le lien annoté) désigne au plus une entité B, mais que cette entité B peut être attachée à plusieurs entités A. Cette annotation doit donc être posée sur un lien unique (c'est-à-dire sur un attribut de type référence vers B)

- **@ManyToMany** qui indique qu'une entité A (possédant le lien annoté) désigne une ou plusieurs entités B, et que ces entités B peuvent être attachées à plusieurs A. Cette annotation doit donc être posée sur un lien multiple (c'est-à-dire sur un attribut collection de référence vers B)

Pour un couple de liens asservis, il faut faire attention à ce que les deux annotations soient compatibles :

- Un lien annoté par **@OneToOne** ne peut avoir comme inverse qu'un autre lien annoté par **@OneToOne**.
- Un lien annoté par **@OneToMany** ne peut avoir comme inverse qu'un lien annoté par **@ManyToOne** (et vice-versa)
- Un lien annoté par **@ManyToMany** ne peut avoir comme inverse qu'un autre lien annoté par **@ManyToMany**.

Mais comment indiquer à Hibernate qu'un lien est asservi à un autre et qu'ils forment tous deux une relation ? On le fait en précisant le nom du lien inverse sur l'un d'entre eux à l'aide de l'attribut "**mappedBy**" des annotations **@OneToOne**, **@OneToMany**, **@ManyToOne** et **@ManyToMany**. Notez que la valeur de cet attribut est un nom d'attribut et non un nom de classe ! Comme souvent, on utilise le nom de la classe désignée comme nom du lien (exemple : attribut " client " de la classe " Contrat " désigne un objet de type " Client "), cela peut prêter à confusion ...

On pourrait traduire " mappedBy " par " asservi à ". Dans un couple de lien asservis, un et un seul des deux liens doit avoir son attribut " mappedBy " renseigné. Hibernate utilise l'autre lien comme référence pour le mapping. Les informations complémentaires de mapping (s'il y en a) doivent être portées par le lien **qui ne possède pas** d'attribut mappedBy sur son annotation).

Petite précision : un lien libre, c'est-à-dire qui n'a pas d'inverse, ne doit pas avoir d'attribut mappedBy renseigné !

Que se passe-t-il si l'attribut " mappedBy " est oublié ? Et bien Hibernate considère que les liens ne sont pas asservis et qu'ils forment chacun une relation indépendante !

Entité propriétaire du lien

L'attribut " mappedBy " est beaucoup plus important qu'il n'y paraît ... En effet, lors des mises à jour, Hibernate, pour une relation donnée possède deux liens. Or en face, il n'y a qu'une clé étrangère (ou un enregistrement dans une table de relations pour les relations N-N et certaines relations 1-N). Hibernate ne fait la mise à jour qu'une fois (et non, pour chaque lien, c'est-à-dire deux fois). Hibernate, donc, " choisit " le lien qui sera utilisé comme source d'information lors de la mise à jour. L'autre lien est ignoré.

Ce lien sélectionné est celui **qui ne possède pas** l'attribut " mappedBy ". Cet attribut sera donc mis à jour en même temps que l'entité qui le possède. Cette entité est donc "**propriétaire**" (au sens d'Hibernate) de la relation. Attention ! Cette propriété là est purement technique ! Elle n'a rien à voir avec la notion de composition/agrégation.


FarPoint Spread for ASP.NET à partir de € 718

FarPoint

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

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


DXperience Enterprise à partir de € 1 009

Devexpress®

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

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


ActiveReports for .NET à partir de € 465

DATA DYNAMICS

Logiciel de rédaction de rapports le plus populaire, sans runtimes ni royalties.

- Produisez une grande variété de rapports depuis vos applications .NET
- Inclut le nouveau contrôle de diagrammes intégré
- Assistant convivial de conversion pour rapports Microsoft Access
- Exportez les rapports aux formats PDF, Excel, RTF, HTML, texte et TIFF
- La version Professional Edition inclut le module Report Designer


Help & Manual à partir de € 271

ecsoftware

Créez une aide et une documentation de formats multiples.

- Éditeur XML WYSIWYG avec formatage et édition, de style traitement de texte
- Création par utilisateurs multiples simultanés
- Sortie dans tous les formats d'aide Windows standard, y compris HTML, WinHelp, MS Help 2.0, Visual Studio Help, aides sur navigateurs, PDF et RTF
- Sources de projets XML pour intégration d'outils tiers de localisation et création de contenus

tion bien connue des concepteurs objets. Si le code est écrit rigoureusement, c'est-à-dire :

- Si les entités liées A et B sont toutes les deux attachées (i.e. contenues dans le cache de premier niveau de la même session Hibernate),
- Et si les deux liens formant la relation sont mis tous les deux à jour et ceci de façon cohérente (on indique à A qu'il désigne B et à B qu'il désigne A),

Alors, le choix par Hibernate de l'un ou l'autre lien pour effectuer la mise à jour n'a pas d'importance pour le développeur. Mais – car il y a un " mais ", Hibernate ne vérifie rien et part du postulat que le code qu'on lui soumet n'est pas bogué. Donc, si une de ces règles n'est pas respectée – ce qui arrive très, très facilement – Hibernate peut, selon les cas, effectuer correctement la mise à jour ou pas. Sans jamais rien dire. En conséquence, il est impératif de bien vérifier que lorsque l'on met à jour un lien, on met aussi l'inverse à jour. Et il faut s'assurer que les deux entités sont bien attachées à la même session Hibernate. Et surtout, il ne faut pas compter sur Hibernate pour signaler quoi que ce soit en cas d'oubli. Juste pour donner une idée de la difficulté qu'il peut avoir à mettre à jour correctement un couple de liens, prenons le cas d'un lien de type 1-1 : La classe Voiture est liée à la classe " Moteur " par son attribut " moteur ". Le moteur connaît la voiture sur laquelle il est installé à l'aide du lien " voiture ". Très naïvement, nous pourrions écrire :

```
void affecterMoteur(Voiture v, Moteur m) {
    v.setMoteur(m);
    m.setVoiture(v);
}
```

[Fig.1]

Et bien, ce code est faux ! En effet, il se peut qu'au moment où cette méthode est appelée, la voiture possède déjà un moteur et que le moteur qu'on veut lui affecter est déjà placé sur une autre voiture. Il faut donc au préalable casser les liens existants. La nouvelle version de notre méthode sera donc :

```
void affecterMoteur(Voiture v, Moteur m) {
    if (m.getVoiture() != null)
        m.getVoiture().setMoteur(null) ;
    if (v.getMoteur() != null)
        v.getMoteur().setVoiture(null) ;
    v.setMoteur(m);
    m.setVoiture(v);
}
```

Okay ? Et bien non ! Ce code est toujours faux ! Car " v " ou " m " peuvent être nuls. Le code juste est le suivant :

```
void affecterMoteur(Voiture v, Moteur m) {
    if (v.getMoteur()==m)
        return ;
    if (m!=null && m.getVoiture() !=null)
        m.getVoiture().setMoteur(null) ;
    if (v !=null && v.getMoteur() !=null)
        v.getMoteur().setVoiture(null) ;
    v.setMoteur(m);
    m.setVoiture(v);
}
```

Le premier test n'est pas strictement indispensable, mais il permet d'éviter des effets de bord que pourraient induire l'appel inutile aux méthodes " set " (comme celui de positionner un flag " dirty " par exemple).

On peut voir ici, que maintenir la cohérence des liens pour une même relation n'est pas un exercice trivial ...

Juste pour vous convaincre, qui n'a pas écrit un constructeur d'entité de ce type ?

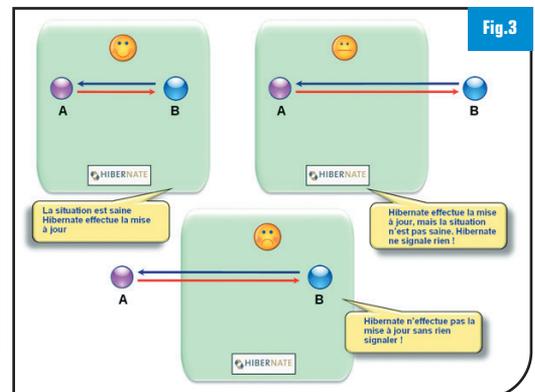
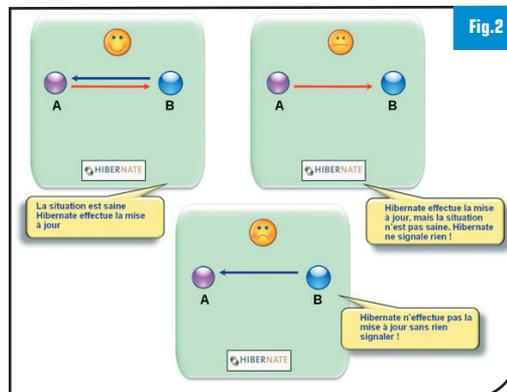
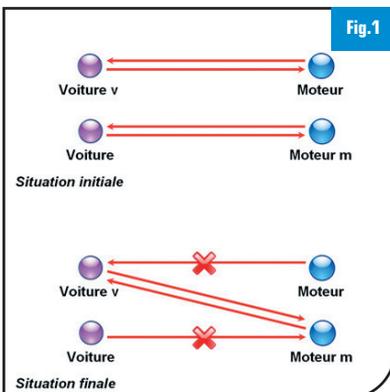
```
public Voiture(Marque marque, String immatriculation, Moteur moteur) {
    this.marque = marque ;
    this.immatriculation = immatriculation ;
    this.moteur = moteur ;
}
```

Et oui ... une erreur est si vite arrivée ... Surtout lorsqu'on est " aidé " en cela par Eclipse.

Mais alors, pourquoi Hibernate ne se charge pas directement de mettre à jour automatiquement les liens inverses (comme le faisaient les EJB 2.0) ? C'est un choix délibéré (de l'équipe Hibernate/JPA, Gavin King en tête). L'idée est de respecter la sémantique de Java. En gros, il s'agit d'apporter un service nouveau (la persistance) à un agrégat d'objets sans en changer la logique classique.

Choisir une entité propriétaire

Comment choisir l'entité propriétaire d'une relation ? Le principe le plus simple à appliquer est le suivant : il faut préférer l'entité mappée à la table qui possède la clé étrangère. En JPA c'est obligatoire. Avec les extensions Hibernate, on peut déroger à cette règle, mais au prix généralement d'un nombre d'accès à la Base de données plus important.



Attention au cas des entités 1-N. L'entité qui possède la clé étrangère est celle qui possède le lien annoté par @ManyToOne. Mais vous avez aussi le droit de décider que l'entité A propriétaire est celle qui possède la collection de B (c'est-à-dire le côté @OneToMany). Comme l'entité A propriétaire est incapable de disposer dans sa propre table d'une clé étrangère qui référence plusieurs T-uple, Hibernate va tenter d'utiliser une table intermédiaire ... Et ceci afin de laisser la table des B en complète gestion à la classe B. Vous pouvez forcer Hibernate à utiliser B en ajoutant une annotation @JoinColumn. Mais vous sortez alors de la garantie JPA.

Pour les relations 1-1 et 1-N, donc c'est facile. Mais pour les relations N-N utilisant donc une table intermédiaire, quel est le bon choix ? En fait, il semblerait que choisir l'une ou l'autre des entités soit équivalent dans ce cas. Il existe donc toute une variété d'outils de choix : la pièce de monnaie, le dé (pair/impair), ...

Maladie, diagnostic et soin

Tout ce qui vient d'être dit dans cet article doit vous aider à trouver où est la faille dans votre programme lorsque Hibernate ne fait plus correctement son travail. Le symptôme caractéristique d'une erreur dans la gestion des inverses est la mise à jour incomplète :

- Les entités A et B sont modifiées, Les modifications apportées à A comme à B ont été correctement mises à jour, mais la relation reste inchangée. Dans ce cas vous avez probablement oublié de mettre à jour un des deux liens. Et pas de chance, il s'agissait du lien porté par l'entité propriétaire, la seule prise en compte par Hibernate. [Fig.2]
- Autre cas de figure qui arrive fréquemment : vous avez bien mis à jour les deux côtés de la relation, mais la relation n'est néanmoins

pas mise à jour. Regardez si les autres attributs de l'entité propriétaire ont été mis à jour, eux, si ce n'est pas le cas, c'est que votre entité n'est pas attachée et que vous êtes en train de mélanger entités attachées et détachées, ce qui risque d'être rapidement très fâcheux ... [Fig.3]

Ce deuxième cas de figure est traitre. Il arrive fréquemment lorsqu'une pièce de code ajoute une nouvelle entité B à la liste des entités B de A. Hélas, vous avez oublié d'appeler

```
session.persist(b);
```

L'entité n'a pas été insérée en Base de données et donc elle est restée en dehors du champ d'Hibernate. Certes, vous l'avez ajoutée à la liste des B de A qui, elle, est attachée. La belle affaire ! C'est B qui est propriétaire du lien. Lorsque A est sauvegardé, la collection des B est purement et simplement – et silencieusement – ignorée par Hibernate, qui aurait parfaitement pu se rendre compte de l'erreur mais qui ne le fait pas !

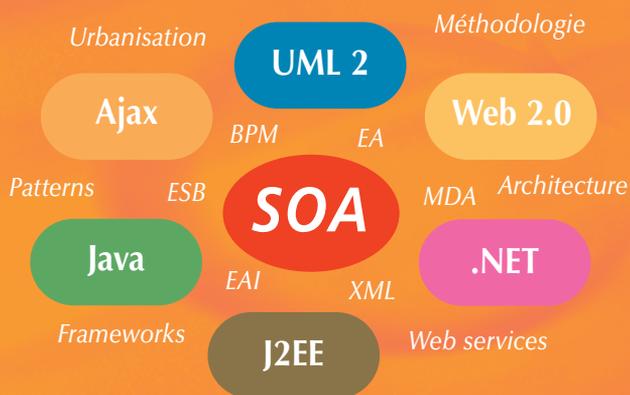
Il est bien sûr possible d'utiliser la création en cascade (cascade={CascadeType.PERSIST}) à la place d'un appel à persist() pour corriger l'anomalie.



■ Henri Darmet

Directeur Technique - Objet Direct / Homsys Group
Objet Direct, filiale à 100% de Homsys Group est une société de conseil, de services et de formation, spécialisée sur les technologies objet et Web. Conseil en méthodologie, en architecture et en urbanisation du SI, développement applicatif, édition et distribution de logiciels.
www.objetdirect.com

Soyez prêts pour les nouvelles architectures SOA et Web 2.0 !



SOA est devenu en peu de temps le mot-clé des développements logiciels. SOA est une nouvelle façon de faire qui s'appuie sur un ensemble de technologies existantes: UML, J2EE, .Net, XML, etc. Maîtriser SOA implique de maîtriser ces technologies pour les associer efficacement au sein d'une nouvelle approche.

SOFTEAM Formation, forte de son expérience en Méthodologie, Architecture et Développement, a construit un cursus complet de formation SOA qui vous permet de débiter dès les phases amont, de poursuivre en architecture, et d'aller jusqu'à la réalisation dans le langage de votre choix.

Nouveau catalogue Formation 2009 :

UML pour la maîtrise d'ouvrage	2 j
Analyse et conception avec UML	4 j
SOA Alignement Métier du Système d'Information	2 j
SOA Architecture d'Entreprise (EA)	2 j
SOA Méthodologie pour SOA	2 j
SOA Architecture technique SOA	2 j
SOA Développement de Web Services en Java	3 j
SOA Développement de Web Services en C#	3 j
Architecture distribuée: la synthèse	2 j
Programmation orientée objet avec Java	4 j
Développement d'applications JEE 5	5 j
Développement d'applications JEE 5 Front End	4 j
Développement d'applications JEE 5 Back End avec EJB 3	3 j
Maîtrise du framework (Struts / JSF / Spring / Hibernate)	3 j
Développement d'applications .NET / C#	4 j
Développement d'applications RIA avec	2 j
(Ajax / Dojo / GWT / FLEX3)	4 j

Convergence SOA, UML2, BPMN, EA

Modélisation EA, BPMN, SOA avec Objectteering SOA Solution	2 j
Analyse et Conception UML2 avec Objectteering Modeler	5 j
Expression de besoins en UML2 avec Objectteering Scope Manager	1 j
Architecture MDA avec Objectteering MDA Modeler	2 j
Génération de code Java, .NET C#, C++ avec Objectteering Developer	1 j

SOFTEAM Formation

Calendrier complet et inscriptions en ligne

www.softeam.fr



Tél. : 01 53 96 84 30 - Fax : 01 53 96 84 01
Paris : 21 avenue Victor Hugo - 75016
Rennes - Nantes - Sophia Antipolis

SOFTEAM
Think Object

Choisir son framework **AJAX**

Pour ceux qui découvriront les joies d'AJAX, un petit rappel peut s'avérer nécessaire. AJAX est un acronyme désignant ni plus ni moins une utilisation conjointe de plusieurs technologies : le HTML et le CSS qui structurent et affichent les pages ; le Javascript qui interagit avec le modèle DOM – modèle représentant le document de façon structurée – dynamiquement ; et l'objet XMLHttpRequest qui est le maillon d'AJAX, indispensable à l'échange de données client–serveur.



Plusieurs réponses à cela, selon votre cas de figure. Si vous n'utilisez pas - encore - AJAX, les intérêts sont les suivants :

- limiter les requêtes d'accès aux bases de données en ne mettant à jour qu'une partie de l'information affichée.
- L'utilisateur peut effectuer plusieurs actions qui déclenchent des appels AJAX. Ces différentes requêtes ne se bloquent pas et sont traitées en parallèle par le serveur.
- Aucun plug-in n'est nécessaire à l'installation côté client. Tous les navigateurs actuels proposent l'objet XMLHttpRequest.

Si vous utilisez déjà AJAX, mais en écrivant vous même l'intégralité de vos fonctions Javascript, peut-être devriez vous envisager l'utilisation d'une bibliothèque ou d'un framework (ensemble de bibliothèques) facilitant les développements.

Prenez toutefois garde à une utilisation mal pensée ou mal employée d'AJAX et de ses frameworks. Le fonctionnement implique de prendre en compte de nouveaux paramètres, comme l'impossibilité de mettre une page en favori, au sens où on l'entend, avec son contenu à un instant T donné, ou encore de nouvelles failles de sécurité qui peuvent apparaître.

Actuellement, les différentes solutions sont dans leurs dernières versions, compatibles avec la plupart des navigateurs Internet, ce qui permet de s'affranchir d'une certaine

lourdeur à valider ses applications à destination de plusieurs navigateurs.

Comment choisir ?

De nombreux frameworks existent dans le monde du Libre, offrant la possibilité de développer ses applications web en y intégrant AJAX. Cet article a pour but de présenter un lot de bibliothèques et de frameworks parmi les plus usités par la communauté des développeurs. Différents niveaux de frameworks sont à distinguer : nous trouvons donc les frameworks de bas niveau et les frameworks dits de haut niveau. Un framework de bas niveau donne la possibilité au développeur d'interagir au maximum avec son implémentation d'AJAX à l'aide des fonctions génériques proposées par la bibliothèque. Les effets AJAX ou les éléments dont le fonctionnement se basera autour d'AJAX peuvent être réalisés par le développeur lui-même. Les frameworks de haut niveau, quant à eux, fournissent d'emblée une série d'éléments et/ou d'effets prêts à l'emploi. Nous allons voir au cours de cet article ces solutions qui répondraient à différents scénarii.

Pour choisir une solution AJAX (moteur ou framework), il y a un certain nombre de points à considérer : la gestion des événements, l'exploitation et les éventuelles extensions du modèle DOM, la propreté du code, qualité de la documentation, activité

de la communauté, modularité de la bibliothèque / framework, nombre et variété des effets visuels...

Chaque critère peut avoir un poids différent en fonction de l'entreprise, du projet, des habitudes de l'équipe de développement. Au fil de cet article nous aborderons ces différents éléments de réponse, qui finalement détermineront la solution vers laquelle le choix se portera.

Les différences entre les solutions résident majoritairement dans l'approche qu'on souhaite aborder pour développer son application. Il est ainsi primordial de se demander si on est à la recherche d'une solution nous permettant de développer des "grosses" applications à caractère professionnel, avec des éléments d'interface puissants et variés, au détriment d'une durée de prise en main qui pourrait être assez longue ; ou si au contraire l'application développée fera usage des interactions offertes par AJAX de façon restreinte.

Un moteur, sinon rien

On utilise le terme de moteur pour des "extensions" se présentant sous la forme d'un simple fichier Javascript. Prototype est un des moteurs AJAX les plus utilisés par la communauté de développeurs web (<http://www.prototypejs.org/>). L'intégration du moteur est de ce fait vraiment simpliste. Un moteur tel que Prototype, à l'inverse de frameworks plus lourds - solutions présentées

LEONARDI

W4-LYRIA : la modélisation au
coeur des applications métier



ihm

EN TOUTE SIMPLICITÉ !

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

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



NOUVELLE
VERSION
LEONARDI
V4.2
open source

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



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

plus loin -, nécessite énormément de développement en Javascript et donc de bonnes connaissances sur le langage. Il faut alors se demander si la compétence requise est présente au sein de son équipe.

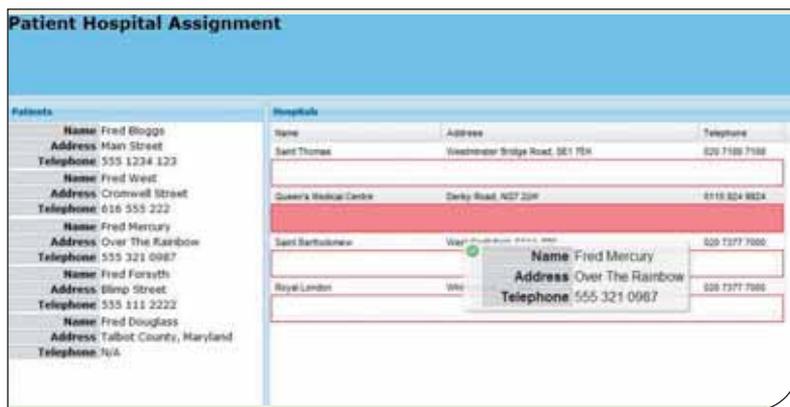
Un autre facteur à ne pas négliger dans le choix d'une solution est la documentation. Sa qualité, la présence d'exemples, et son maintien à jour au gré des versions sont des éléments qui peuvent impacter grandement les temps d'adaptation et de développement de votre application. Dans le cadre de Prototype, la documentation est plutôt minimaliste et nécessite un bon niveau en programmation Javascript. De ce fait, l'écriture d'une application complexe, avec une interface évoluée, peut vite s'avérer fastidieuse.

Afin de maîtriser cette programmation Javascript, Prototype propose des fonctions raccourcis - helpers - permettant de simplifier la manipulation du modèle DOM. Le choix de Prototype est finalement souvent retenu au sein des applications exploitant Ruby On Rails, le modèle objet s'en approchant.

L'extension et l'exploitation du modèle DOM est un critère pouvant influencer le choix d'une solution AJAX plutôt qu'une autre. A l'instar de Prototype, une autre bibliothèque propose des raccourcis pour manipuler les données souhaitées. Il s'agit de jQuery (<http://jquery.com/>). Basée sur CSS3 et XPath, jQuery permet de parcourir le DOM de son document de façon précise et rapide grâce à des alias bien pensés.

Une documentation des plus utiles...

Par rapport à Prototype, la syntaxe proposée par jQuery est relativement simple. De surcroît, cette bibliothèque bénéficie d'une documentation bien fournie, entourée par une communauté extrêmement active. Lors du choix de votre toolkit AJAX, pensez à bien consulter ces documentations en ligne qui vous permettront de vous forger un bon avis sur l'étendue de l'API, sa facilité d'approche, et éventuellement sur l'activité de la communauté qui se trouve



derrière. Elle est susceptible de proposer des améliorations, des exemples, qui auraient pour répercussion directe de diminuer les temps d'apprentissage d'une nouvelle solution, et finalement favoriser un choix plutôt qu'un autre.

Des effets visuels facilement

Si la mise en place d'effets et la création d'une application web interactive vous tient à coeur, votre choix peut se porter sur Script.aculo.us. (<http://script.aculo.us/>)

Script.aculo.us est une bibliothèque Javascript basée sur le moteur Prototype. Tandis que Prototype demande de fortes connaissances en Javascript pour mettre au point des effets visuels intéressants, Script.aculo.us en propose une multitude, directement prêts à l'emploi. Si votre choix d'utiliser AJAX est justifié par l'agrément de vos pages avec des effets visuels rendant l'utilisation de votre site/application plus agréable et plus ergonomique, essayer cette surcouche de Prototype peut s'avérer une bonne expérience, tant la mise en place et les premières utilisations des effets sont simples et rapides. Ces effets vont du simple surlignement de caractères au *Drag and Drop* "glisser-déposer" d'éléments. Conférer au contenu un dynamisme et une interactivité est un jeu d'enfant et ce, avec très peu de code à écrire au final.

Des widgets pour mes interfaces...

Si dans vos critères de choix, vous souhaitez pouvoir tirer parti de plusieurs bibliothèques, ExtJS est la

solution qui s'offre naturellement à vous (<http://extjs.com/>). Via un système d'adaptateur qui lui est propre, ExtJS peut être utilisée conjointement avec d'autres bibliothèques telles que jQuery ou YUI. Ce paramètre est à prendre en considération si les atouts d'ExtJS vous séduisent, mais que vous souhaiteriez y trouver une complémentarité par le biais d'une seconde bibliothèque pour adjoindre une série d'effets ou de manipulation de l'arbre DOM, impossible de manière standard.

L'autre grande qualité d'ExtJS réside dans sa très grande variété de widgets disponibles. Vous souhaitez mettre en place une interface réactive, interactive et ce, assez rapidement ? Cette bibliothèque est riche en éléments d'interfaces : combo-box, grid, form, panel etc. Ils sont extensibles à loisir, le développeur peut donc les personnaliser selon ses souhaits.

Et pourquoi pas un framework complet ?

Un framework devient encore plus intéressant lorsqu'il est utilisable avec vos outils habituels de développement et qu'il est capable d'interagir avec eux. Ce point doit être pris en compte dans votre grille de décision, un outil adapté pour exploiter pleinement un framework peut décupler la productivité. En l'occurrence, si votre équipe de développement (ou vous-même) utilisez Dreamweaver, le site d'Adobe vous invitera tout naturellement à vous tourner vers Spry. (<http://labs.adobe.com/technologies/spry/>)

Spry est le premier véritable framework AJAX que nous vous présentons ici. Disponible sous licence BSD,

Le Savoir est Pouvoir

Nouveau! NetAdvantage for Silverlight Data Visualization
Rendre les projets de Business Intelligence possibles pour
chaque entreprise



Composants pour interfaces utilisateur pour

- **Concevoir** des tableaux de bord riches
- **Visionner** les bases de données métiers
- **Donner du pouvoir** aux décideurs

Pour de plus amples infos : infragistics.com

Sales-europe@infragistics.com

N°Vert 0800 667 307



NetAdvantage®

Data Visualization Silverlight

mapping

charts

gauges

zoombar

timeline

& more!

la politique du framework, développée par Adobe, est de se présenter comme une extension au HTML. Dès lors, pas besoin d'être un expert en Javascript.

Au niveau de l'architecture, Spry peut très facilement s'intégrer avec les différentes technologies serveurs du marché : PHP, ColdFusion, ASP.Net... Le choix de la technologie proposée par Adobe est un moyen simple d'incorporer des données XML, JSON ou HTML au sein d'une page écrite en HTML, CSS, et avec un strict minimum de Javascript.

L'atout-clé de Spry réside dans sa proximité avec le HTML. Si vous êtes en passe de choisir un framework qui doit être très facilement exploitable par un webdesigner, Spry est le framework à adopter, à plus forte raison si vous êtes un utilisateur de Dreamweaver et d'autres produits de la firme comme Fireworks. Si votre projet nécessite une interface évoluée, personnalisée, que le graphiste / webdesigner doit pouvoir mettre en place facilement, Spry répond à ces critères. Cependant, comme nous l'avons vu plus haut, l'activisme de la communauté peut jouer un rôle déterminant, ce qui ne semble pas être le point fort de ce framework dépourvu de vraie roadmap, argument dont le poids est à mettre en balance avec celui d'une exploitation simplifiée à l'extrême, destinée aux webdesigners.

Quid de la modularité ?

Un framework, contrairement à un moteur comme Prototype ou à une bibliothèque – est souvent synonyme de lourdeur. On y trouve de multiples composants, des couches qui s'entrecroisent, des fichiers Javascript de plus en plus lourds à charger... Or si l'on souhaite opter pour l'utilisation d'un framework en particulier, il est primordial de se poser la question sur nos besoins réels et nos attentes finales quant à l'exploitation d'AJAX. Est-ce pour bénéficier d'effets simplement ? Étendre le modèle DOM ? Ou effectuer des requêtes de manière asynchrone ? Ces questions permettent de se prémunir contre les soucis évoqués il y a quelques

lignes. Les frameworks en proposant parfois "trop", un d'entre eux a fait le choix de la modularité : Mootools. (<http://mootools.net/>)

Dès l'étape de téléchargement, Mootools laisse le développeur construire son package en ne sélectionnant que les composants qu'il compte utiliser. Le framework Mootools, en plus d'être compact, permet entre autres de mettre en place des effets dynamiques. La grande particularité par rapport aux précédents frameworks est son approche Orientée Objet. Qu'il s'agisse de créer un élément fonctionnant autour d'AJAX ou d'utiliser un effet, tout passe par des objets. Cette approche est clairement à retenir pour un développeur ou une équipe de développeurs habitués à cette approche de la programmation et la philosophie qui en découle.

Pour les architectures JEE, .Net..

Jusqu'ici, la plupart des solutions évoquées sont des solutions majoritairement mises à contribution avec PHP en guise de technologie côté serveur. Mais la plupart d'entre elles fonctionnent également avec ASP.NET, parfois Ruby On Rails ou Coldfusion. Si votre application s'intègre plutôt au sein d'une architecture JEE, il existe des frameworks

parfaitement adaptés, tels que Sweetdev (<http://www.ideotechnologies.com/sweetdev-ria.html>) ou Echo2 (<http://echo.nextapp.com/site/>), la troisième mouture étant en version bêta. Pour des projets dans un contexte .NET, Atlas de Microsoft est disponible. Parfaitement intégré aux outils Microsoft, son utilisation devient fastidieuse en dehors de l'environnement Microsoft.

En conclusion...

J'espère que cette série de situations probables, de cas de figures spécifiques et propres à chaque projet vous aura permis d'avoir une vision élargie des solutions existantes autour des bibliothèques et frameworks du monde AJAX. Gardez en tête que l'utilisation d'AJAX doit être justifiée par une valeur ajoutée apportée au produit, et que le choix d'un toolkit doit permettre de répondre aux besoins exprimés pour un projet donné, sur une architecture donnée, au sein d'une équipe de développement qui possède des méthodes de travail qui lui sont propres. De cette façon vous pourrez tirer le meilleur parti d'AJAX.

■ **Wilfried Lawson**
wilfriedlawson@hotmail.com
Étudiant à l'EPITA au sein de la spécialisation MTI, Multimédia et Technologies de l'Information

Petit panorama des solutions Ajax du marché

Nom	Avantages	Inconvénients
Prototype	Raccourcis simplifiant la manipulation du DOM et la programmation javascript	Documentation peu fournie
Scriptaculous	Multitude d'effets graphiques Facile à utiliser	Faible documentation
Mootools	Programmation Orientée Objet Extension du DOM Grande modularité	Communauté qui pourrait être + active
ExtJS	Riche collection de widgets personnalisables Documentation de l'API bien réalisée Forte communauté de développeurs Adaptateurs rendant possible l'utilisation d'autres frameworks AJAX en parallèle	Temps d'apprentissage pour utilisation poussée
jQuery	exploitation puissante du DOM Syntaxe simple Communauté réactive	On cherche... la communauté fixant les bugs très rapidement !
Spry	Très accessible Intégration Dreamweaver	Peu d'évolutions
Sweetdev	Intégration environnement JEE Nombreux widgets	Faible communauté

egilia[®]

LEARNING

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

Formations certifiantes en informatique & management

- ✓ **Certifications comprises** avec toutes nos formations
- ✓ **Ordinateur portable** offert avec les supports
- ✓ **Abonnements** : *L'Informaticien, Hakin9, Management, Programmez, Solutions & Logiciels...* offerts !
- ✓ **30 jours de coaching** (support technique)
- ✓ Formations **éligibles DIF, FONGECIF, OPCA...**
- ✓ Garantie **“Enchanté ou Invité”**
- ✓ **Votre carte personnelle d'accès à SMARTcenter** et à la SMARTLibrary (accès à vie à toutes les formations EGILIA)
- ✓ **Certificate of Excellence** en fin de formation...



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

Découvrez les
nombreux avantages
sur www.egilia.com !

CONTACTEZ NOS CONSEILLERS FORMATION

 **N° National 0 800 881 558**

APPEL GRATUIT DEPUIS UN POSTE FIXE

www.egilia.com

WinDev : l'atelier de développement tout-en-un !

“Développez 10 x plus vite avec WinDev”, c’est le fameux slogan de PC Soft. Comparativement à d’autres langages, on n’est pas loin de la vérité... WinDev est facile d’apprentissage, intuitif grâce à un langage qui lui est propre (WLangage) et en français, ce qui facilite la recherche des fonctions, une fois celles de base connues.

WinDev limite également considérablement le nombre de lignes de code. La création de maquettes se fait en quelques glissements de composants, ce qui permet de fournir un visuel ergonomique et la cinématique d’une application en un temps restreint. Exemple : la réalisation d’un exécutable bilingue permettant la présentation de la cinématique de 7 écrans comportant 30 champs de saisie, 20 tables, 15 boutons, 5 listes, 2 zones répétées en 2 heures. [Fig.1]

Le revers de la médaille nécessite une certaine rigueur dans le développement car il est possible de déclarer des variables à tous niveaux (global au projet, global à un composant, local à un événement), de créer des procédures globales ou locales, de faire une partie du code en anglais et une autre en français, d’imbriquer des "GOTO MonParagraphe" à l’intérieur d’un code procédural. Il faut donc être organisé sinon le code d’une application peut rapidement devenir illisible.

Comment se mettre à WinDev

WinDev est assez simple d’apprentissage ; une formation de 5 jours à un développeur ayant déjà une première expérience peut être suffisante pour avoir en main l’ensemble des éléments de base permettant de créer une application. L’idéal est de se faire accompagner par un consultant expérimenté sur un sujet concret de l’entreprise ou un sujet permettant d’appréhender un large spectre de composants et fonctionnalités WinDev. Certaines étapes principales doivent avoir été abordées : comment créer les IHM, l’accès aux données via l’éditeur d’analyse WinDev ou non, les fonctions permettant la manipulation des composants et des données, comment créer un exécutable et éditer une installation avec WDIInst et comment créer un état avec WDEtat. L’auto-apprentissage est également possible mais nécessite évidemment beaucoup plus de rigueur et de temps car, même si l’aide en ligne est riche, il est toujours plus facile d’avoir un professionnel en face de soi pour répondre aux questions.

Peut-on facilement passer d’un langage comme JAVA ou PHP à WinDev ?

Passer d’un langage comme PHP ou JAVA à WinDev est surprenant, l’inverse est tout aussi vrai. WinDev est d’une grande souplesse et un développeur habitué à JEE réagira au fait qu’on puisse faire un peu tout et n’importe quoi n’importe où dans WinDev. Un développeur WinDev passant à JEE pensera de façon quasi certaine une chose : avec WinDev j’aurai déjà terminé depuis longtemps. Les outils PC Soft permettent la génération d’autres langages : WebDev génère du code PHP compatible avec PHP4 et PHP5, il est également possible d’y intégrer des fichiers Flex.

Il est également possible de faire la saisie du code directement en PHP dans l’éditeur WebDev : en un clic sur le bandeau de code d’initialisation de la page on passe la partie serveur de "WL" à "PHP". De la même façon pour le Javascript, il suffit de faire passer "WL" à "JS" sur la partie de code dédiée au chargement de la page du côté navigateur. WebDev peut aussi s’interfacer avec différents services proposés sur le Web par des sites en effectuant un mashup au niveau du serveur ou directement en code navigateur.

Exemple d’un extrait de code d’une page PHP générée par WebDev à partir d’un projet "Dynamique PHP" : la page PHP ainsi que la feuille de style sont générées. **Attention**, code généré avec la V12 (extrait).

```
<?php
//#12.00Tj Mon_Projet
//-----
// Include standard (définition des types, fonctions utilitaires)
//-----
$CheminRepRes='./res/';
require_once($CheminRepRes.'WD12.0/WD120Page.php');
require_once($CheminRepRes.'WD12.0/WD120Ajax.php');
```



Fig.1



Fig.2

```
require_once($CheminRepRes.'WD12.0/WD120HF.php');
// Equivalent de [%URL(%)]
$gszURL = 'PAGE_Page1.php';
session_start();
// protection contre register_globals = on
unset($PAGE_PAGE1);
if (SID != '') $gszURL .= '?'.SID;
ChangeAlphabet(0, false);

$gtabCheminPage = array();
```

```
//-----
// Déclaration de la page et de ses champs
//-----
// le 'if (isset())' gère le cas où session.bug_compat_42 est à VRAI
if (!isset($PAGE_PAGE1)) {$PAGE_PAGE1= new CPage();
$PAGE_PAGE1->Nom = 'PAGE_PAGE1';
$MaPage = &$PAGE_PAGE1;
```

Comment fonctionne WinDev ?

WinDev comporte un ensemble d'outils intégrés qui permettent d'aborder l'ensemble du cycle de vie d'un projet : un éditeur de modélisation UML et Merise, un éditeur d'analyses qui permet la gestion des connexions et des fichiers de diverses bases de données, et bien d'autres encore comme l'éditeur d'interface graphique, de dossier technique, de fenêtres, de chartres graphiques, de requêtes, de code, de travail collaboratif, de composants, d'états, d'aide en ligne, d'installation, de maintenance, de tests et de RAD

De plus, il possède un centre de contrôle de suivi de projets dont le but est de gérer le planning de réalisation de sa création à la livraison finale. Il comprend notamment :

- centre de contrôle topologique
- centre de contrôle de planning
- centre de contrôle qualité pour gérer les retours clients (bogues et suggestions que l'utilisateur peut communiquer via chaque applicatif
- centre de contrôle de données
- archivage de projet

Il est ainsi possible de connaître l'ensemble des tâches à réaliser, les développeurs concernés, les durées, l'état d'avancement etc. Ce centre permet aux équipes d'évoluer vers l'industrialisation des méthodes. Les applications bénéficient automatiquement de fonctionnalités par défaut pour l'utilisateur comme le menu contextuel des tables permettant l'export vers Word, Excel, XML, OpenOffice,

la création de fichier PDF et l'envoi vers l'imprimante par défaut ou l'ajout de somme, moyenne et comptage dans une colonne ou le positionnement d'une minuterie sur un bouton (le développeur peut bien entendu désactiver ces fonctions). [Fig.3]

La notion de composant permet de créer facilement son propre framework de développement. Les principaux concepts disponibles sont : notion d'héritage, héritage multiple d'IHM, modèle de champs, composant et composant interne, collection de procédures, fenêtre et page interne, Web service, brique de code.

Quelques fonctionnalités de WinDev

Le WLanguage

Le WLanguage est un langage propriétaire ne pouvant être manipulé qu'avec les outils PC Soft. Il est donc utilisé dans WinDev, WebDev et WinDev mobile. Il est exécuté par une machine virtuelle ou converti en code natif par un compilateur "just in time". Il peut s'appuyer sur le framework Java ou PHP pour WebDev pour une partie de leurs fonctionnalités. C'est un langage procédural qui permet la programmation orientée objet ou séquentielle. La programmation peut être faite en anglais ou en français et Exemple :

```
sHeure est une chaîne
sHeure = HeureVersChaîne(Maintenant())
Info("Il est : " + sHeure)
```

```
sHeure is a string
sHeure = TimeToString(Now())
Info("Il est : " + sHeure)
```

L'éditeur de code et les composants

L'éditeur de code est découpé en parties permettant d'affecter les ordres de programmation à différents niveaux d'un composant. Exemple dans une fenêtre : les déclarations globales qui contiennent en général les variables typées, le code peut être développé à l'initialisation de cette fenêtre, à la fermeture, à la prise et à la perte de focus, à la modification de la taille et à chaque modification. Le code peut être mis également sur d'autres traitements comme par exemple les événements souris, les touches enfoncées ou relâchées. A noter que les variables peuvent être déclarées localement à tous ces types de traitements.

WinDev comporte une large palette de composants (appelés également champs) parmi lesquels : champ de saisie, bouton graphique ou non, liste, combo, table, champ calendrier sur un champ de type date, zone répétée, champ multimédia permettant de jouer un AVI

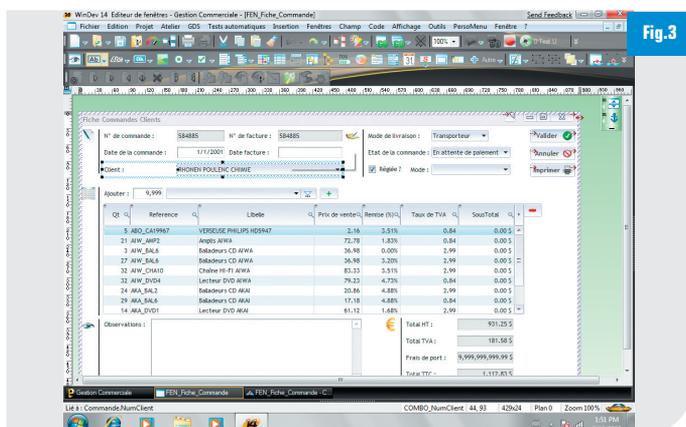


Fig.3

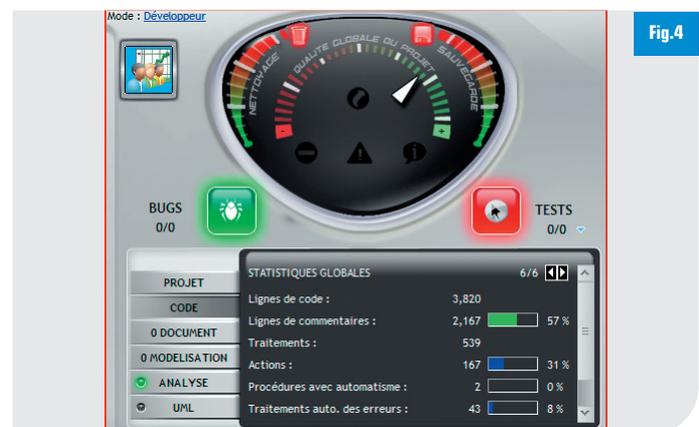


Fig.4

ou un Flash dans une fenêtre, interrupteur, sélecteur, potentiomètre, jauge, arbre, champs html et XAML, champ ou liste image). Ces composants se glissent facilement sur les IHM avec une aide à la saisie au niveau des fonctions de programmation et des aspects graphiques pointus permettant de définir une transparence sur le cadre extérieur d'un onglet, des transitions visuelles entre le passage d'un onglet à un autre sans programmation, des effets de sélection et de survol sur les sélecteurs et interrupteurs ou les barres de mise en forme RTF sur les champs de saisie, activation et désactivation du grisage des fenêtres inactives en modifiant le pourcentage de grisage et adaptation à la luminosité de l'écran.

Exemple de programmation d'un champ combo :

```
gsCodeAnimateur est une chaîne

gsCodeAnimateur = COMBO_ANIMATEURS[COMBO_ANIMATEURS]
//gsCodeAnimateur contient la valeur de l'élément sélectionné

gsCodeAnimateur = COMBO_ANIMATEURS
//gsCodeAnimateur contient l'indice de l'élément sélectionné
(position dans la liste)

i = 3
gsCodeAnimateur = COMBO_ANIMATEURS[i]
//gsCodeAnimateur contient la valeur de l'élément positionné
en 3ème position dans liste

ListeSelectPlus(COMBO_ANIMATEURS,ListeCherche(COMBO_ANIMATEURS,
"TOUS"))
//sélectionne et vient positionner l'élément " TOUS " dans la liste

ListeSelectPlus(COMBO_ANIMATEURS,6)
//sélectionne et vient positionner le 6ème élément de la liste

ListeSelect(COMBO_ANIMATEURS)
//renvoie l'indice de l'élément sélectionné dans la liste
```

L'éditeur de requêtes

L'éditeur de requêtes permet de créer automatiquement des requêtes sur les fichiers de données présents dans l'éditeur d'analyse. La création d'une requête est très simple et intuitive ; il suffit de sélectionner le type de requête (select, insert, update, delete, union ou mode SQL) et de choisir le fichier et les colonnes. La requête est automatiquement générée en code SQL et peut être testée immédiatement. Il est possible de passer des paramètres (même dans les rubriques calculées), de faire des tris, des comptes des sommes des moyennes et de créer des liaisons et des jointures.

Exemple :

```
REQ_SEL_CONTRAT.Param1 = Val(SAI_NUMERO)

SI PAS HExécuteRequête(REQ_SEL_CONTRAT) ALORS
  Erreur("Erreur d'initialisation de la requête"+RC+ErreurInfo())
  HFermeConnexion(ConnexionOracle)
  FinProgramme()
FIN
HLitPremier(REQ_SEL_CONTRAT)

TANTQUE PAS HEnDehors(REQ_SEL_CONTRAT)
  ListeSelectPlus(COMBO_TYPE,ListeCherche(COMBO_TYPE,REQ_SEL_
CONTRAT.TYPE,Faux))
  SAI_DATE_DEBUT = REQ_SEL_CONTRAT.DATE_DEBUT
  TableAjouteLigne(TABLE_CONTRAT, REQ_SEL_CONTRAT.NUMERO, REQ_
SEL_CONTRAT.OTE)
  HLitSuivant(REQ_SEL_CONTRAT)
FIN
HAnnuleDéclaration(REQ_SEL_CONTRAT)
```

Deux versions de l'éditeur de requêtes sont disponibles : l'éditeur intégré fourni avec WinDev , WebDev ou WinDev Mobile, utilisable lors du développement d'une application et l'éditeur utilisateur, uniquement pour WinDev et version indépendante de WinDev utilisable pour un utilisateur final

L'accès aux données

WinDev possède son propre SGBD : HyperFileSQL. Il supporte de nombreux autres systèmes de gestion de bases de données, dont l'accès se fait soit par l'intermédiaire des protocoles ODBC ou OLE DB ou soit par accès natif comme pour les bases classiques Oracle, SQL Serveur, MySQL, PostgreSQL, Informix, Sybase... [Fig.5]

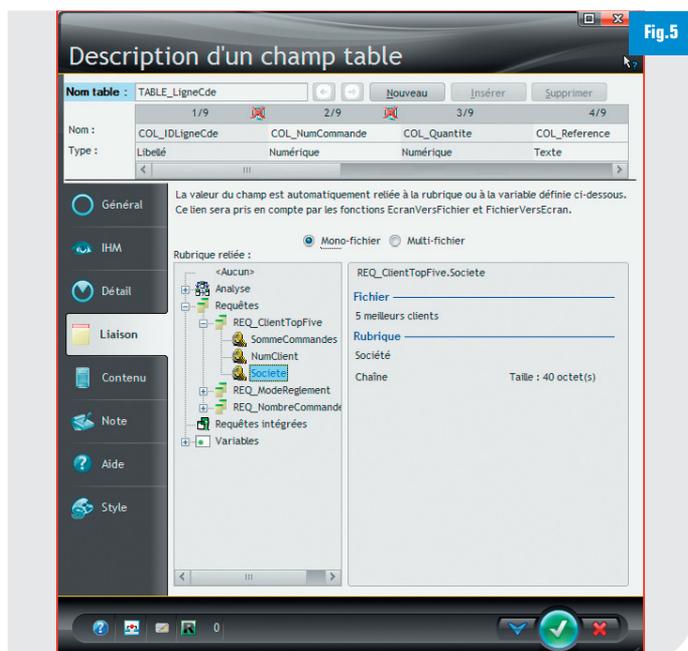


Fig.5

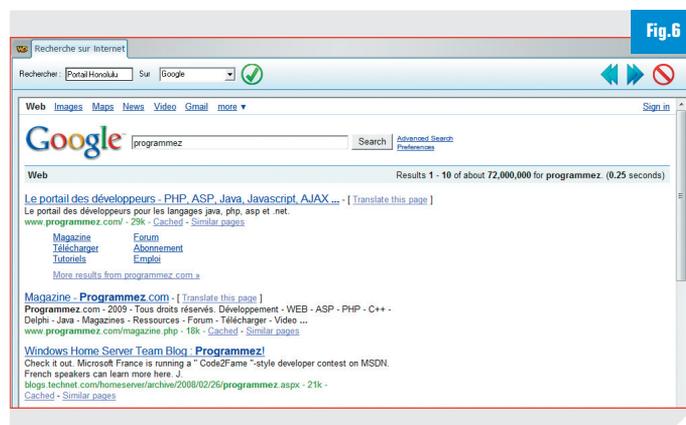


Fig.6

Des ordres spécifiques sont définis en fonction du type d'accès :

```
SQLExec("SELECT QUANTITE FROM PRDUIT", "REQ")

ReqProduit est une Source de Données
HExécuteRequêteSQL(ReqProduit, "SELECT QUANTITE FROM PRDUIT")

HExécuteRequête(REQ_SEL_QUANTITE_PRODUI)
```

Un spectre applicatif varié

Les produits PC Soft sont présents dans de nombreux domaines applicatifs :

- la VoIP pour centres d'appels sur PC-BX qui gère tous types de médias ;
- l'informatisation de trafic portuaire (tracking de navires en liaison avec la capitainerie et les agents maritimes et conférences virtuelles et portail portuaire via WebDev ;
- sur les chaines de télévision avec le pilotage des sources vidéo et audio (WinDev traduit les voix au format PCM et les transforme en WAV), et la gestion du décrochage publicitaire (géré par le multithreading) ;
- l'analyse des bancs d'essais des moteurs d'engins agricoles et le suivi des compétences des opérateurs ;
- progiciel de traduction des échanges de données informatisées (EDI) basé sur XML et sur des conversions en feuilles XSLT pour les feuilles de style

Google

WinDev et WebDev permettent les interactions natives avec les applications Google. Exemple de code qui récupère une liste de produits et l'affiche :

```
MaConnexion est une gglConnexion

MaConnexion.Email = " Nom@gmail.com "
MaConnexion.MotDePasse = " motdepasse "
GglConnecte(MaConnexion)
MesProduits est un tableau de gglProduits
MesProduits = GglListeProduits(MaConnexion)
TableAffiche(TB_PRODUIITS)
```

[Fig.6]

Google Maps

Avec WinDev il est possible de récupérer une carte et de l'afficher, de demander le dessin d'un itinéraire sur la carte, la longitude et la latitude, le taux de zoom etc.

Exemple de types et de propriétés :

GglMarquer, GglChemin sont des exemples de types possédant de nombreuses propriétés : ..Latitude, ..Longitude, ..Taille, ..Couleur, ..Opacité etc.

Google Agenda :

En WLangage, récupérer la liste de ses agendas, des rendez-vous de chacun d'entre eux et de les modifier sont des traitements réalisables nativement.

Exemple de types et de propriétés :

GglAgenda, GglEvenement sont des exemples de types possédant de nombreuses propriétés : ..Titre, ..DatePublication, ..Contenu, ..Auteur etc.

Google Contacts :

WinDev permet par exemple de réaliser une application qui synchronise les contacts d'un téléphone portable avec ceux de Google.

GglContact est un exemple de type possédant les propriétés ..Nom, ..Email, .. Téléphone etc.

La gestion des documents Google Docs et Google Spreadsheets avec les fonctions GglRécupèreDocument et GglEnvoiDocument permet de manipuler les documents par eux-mêmes (upload de documents, recherche parmi les documents, récupération de la liste des documents et leurs détails grâce aux propriétés ..Auteur ou ..Titre par exemple).

Via un ActiveX dans son application ou un Plug In dans le site, GoogleEarth n'est pas oublié ; nativement en WLangage il est possible de positionner des marqueurs, du texte, de déplacer la caméra, zoomer à la souris ou définir un chemin.

Conclusion

Connaissant WinDev depuis la version 5, j'ai eu l'occasion de le pratiquer à tous niveaux de poste (de développeur à responsable informatique), j'apporte maintenant mon expertise en tant que consultant. Pour résumer en quelques mots ce retour d'expérience, je dirai que WinDev a souvent été critiqué à cause de son manque d'ouverture lié au dongle obligatoire, limitant de ce fait sa reconnaissance au niveau étudiant par exemple, bien qu'il existe des versions éducation WinDev et WebDev à tarif adapté. En tant que produit initialement francophone, il a pu peut être limité dans son expansion à l'international (PCSoft est présent dans plusieurs dizaines de pays, ndr) et enfin sa facilité lui a parfois valu, à tort, la critique d'être un outil " amateur ".

WinDev se distingue en effet par sa facilité d'utilisation et une productivité remarquable. Voici un exemple concret : un projet de 5 000 jours homme développé initialement en WinDev a voulu être réécrit dans un langage " nouvelle techno " que je ne cite volontairement pas. Le budget initial a été porté à 25 000 j/h et je pense que le seul résultat positif de ce projet a été d'avoir permis à des dizaines de prestataires d'en vivre pendant quelques années ! La dernière version offre encore un meilleur confort au développeur avec une meilleure adaptabilité aux grands écrans et une refonte de l'interface et permet en quelques clics de générer des IHM pour une multitude d'environnements (Windows 32 et 64 bits, Windows Mobile, Linux, Java, .Net etc.) et de bases de données.

Au fil du temps, le succès de WinDev ne s'est pas démenti. Ne soyons pas trop puriste, gardons l'esprit PME (pragmatique, malin, efficace), WinDev a de beaux jours devant lui pour des équipes sachant l'utiliser de façon optimale. Un service informatique n'est plus uniquement une course en avant vers la technologie et cela depuis bien des années, mais bien un service au premier sens du terme, c'est-à-dire l'apport d'efficacité aux utilisateurs pour faciliter le business de l'entreprise.

■ **David Vandevelde**

Consultant en gestion de services et de projets, expertise WinDev (audit, formation, réalisation) – d.vandevelde@laciteduweb.com

Tout ce qui faut savoir po

C'est devenu un rituel à Programmez ! A l'occasion du salon Solutions Linux, nous préparons un grand dossier Linux/ Open Source. Si dans les dernières éditions nous avons dévoilé les dessous des grands projets open source (OpenSuse, Gnome, Ubuntu, Eclipse, MySQL, OpenOffice...), cette année nous avons voulu nous concentrer sur le développeur et pourquoi et comment développer pour et sur Linux.



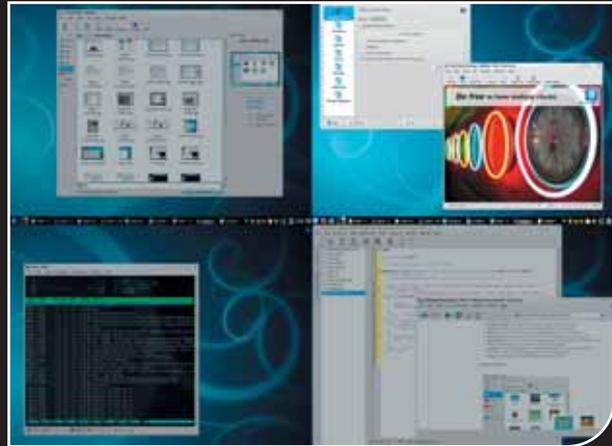
Pour développer sur Linux !

De nombreux lecteurs travaillent sur Windows, mais Linux intrigue et intéresse. Comment donc y passer sans trop de douleur quand on est développeur ?

Tout va dépendre du type d'application que l'on développe. Pour un développeur Java, Linux est une machine de développement comme une autre. De nombreux outils y sont disponibles. Eclipse ou NetBeans fonctionnent parfaitement dessus. Pour du développement C++, Linux offre aussi une bonne plateforme. Longtemps, Linux a souffert d'une image austère pour le développeur habitué à un Visual Studio, à Dreamweaver. Les environnements de type RAD n'étaient pas les plus fréquents.

Aujourd'hui, Linux dispose de plusieurs environnements graphiques très performants : Eclipse, NetBeans, Ajusta, Kdevelop, Qt-creator, MonoDevelop. Par contre, sur le développement web pur et le design web, les choses évoluent peu. Il y a quelques années, nous avons eu l'initiative Nvu, mais qui depuis n'est plus maintenue et nous attendons toujours son successeur. Adobe qui sort Flex et Air pour Linux n'a visiblement pas prévu de versions Linux de Photoshop, de Dreamweaver et de l'ensemble de la Creative Suite. Il existe bien des petits outils comme BlueFish mais il manque toujours un outil web complet.

Pour un développeur arrivant sur Linux et ne connaissant pas grand chose à notre Tux favori, le monde Linux peut paraître un peu déconcertant : nombreuses distributions, plusieurs modèles de déploiement des applications, différentes librairies



d'interface, des implémentations pouvant différer d'une distribution à une autre, des versions différentes, etc. Quand on est habitué à Windows ou à MacOS X, c'est un peu la panique...

Il faut aussi se poser la question de ce que l'on veut faire avec Linux. Est-ce mon nouveau poste de développement ? Qu'est-ce que je veux développer ? Est-ce un développement d'applications pour Linux ou un développement sur Linux (c'est-à-dire des applications qui sont indépendantes du système comme des applications web) ? Selon l'application, les outils et l'approche changent. Si vous souhaitez développer une application Linux, vous devrez déjà savoir sur quelle(s) distribution(s) tournera le projet, le langage que vous souhaitez utiliser. Sans oublier le choix des librairies (et les versions) et l'interface graphique. Avec tout cela, vous disposerez de votre système de déploiement cible... Reste à installer l'outillage nécessaire et à apprendre le modèle de développement Linux...

Dans ce dossier nous allons traverser l'Atlantique pour un reportage spécial au cœur des laboratoires de recherche et développement de Mozilla, nous comprendrons comment fonctionnent les Linux Standard Base, le développement d'une application avec Gnome. Sans oublier le témoignage de développeurs. Nous mènerons l'enquête sur la légalité du code source et les problèmes légaux que l'on doit considérer dans un projet.

Bon Tux !

■ Dossier coordonné par François Tonic





La parole aux développeurs !

Quand on débute la programmation sur Linux, le développeur non-linuxien peut être un peu perdu, ne pas savoir quoi faire. La première question à se poser est : *“que veut-on développer ?”*. De là, on peut circonscrire les besoins en outils, en librairies. Puis vient le choix de la distribution Linux à prendre pour son développement.

“Le choix dépend du type de besoin et du type d'application à développer. Mais les outils restent fondamentalement les mêmes d'une distribution à une autre. Bref, on aura peu de différences”, commente Arnaud Dupuis (consultant Uperto). Le réel défi pour le développeur sera de choisir et de vérifier les librairies : leurs versions,

les disponibilités sur les distributions. *“Même dans une distribution, on peut avoir plusieurs versions. Il faut vérifier les versions présentes dans les distributions. Donc prudence !”,* résume **Anne Nicolas** (director of Engineering, Mandriva).



Compatibilité, incompatibilité

D'une manière générale, les librairies peuvent se révéler complexes à gérer. Heureusement, il est possible de se simplifier le travail en utilisant des outils de vérification, contrôlant les versions, la présence par défaut de telle librairie. La Linux Foundation dispose de LSB (Linux Standard Base) et SuSe propose en ligne l'outil Build Service. Mais cela reste un réel souci pour le développeur, par exemple entre KDE 3.x et KDE 4 dont le modèle rompt radicalement, avec notamment l'utilisation des versions de Qt (librairie graphique) différentes. Mais souvent, le développeur ayant choisi sa distribution ne va pas aller voir ailleurs. *“Je pense que chacun campe sur ses positions. Il fallait une entité (Linux Foundation, ndr) pour imposer des règles”,* poursuit Anne Nicolas. Même si LSB et son outil AppChecker sont incomplets, ils aident réellement le développeur. *“Il peut y avoir des problèmes de compatibilité mais heureusement, cela est*

(très) rare, surtout quand on change de version majeure comme de Qt 3.3 à Qt 4.” nuance A. Dupuis.

Et pour les tests alors, on fait comment ? Une des solutions est de passer par des machines virtuelles pour tester les applications développées, c'est pratique et rapide. Cependant, quand l'application a une grande dépendance avec le matériel, mieux vaut une distribution installée en dur sur un poste.

Sources ou binaires ?

Si votre application est un projet ou si votre projet développé doit être déployé sur des postes de travail, l'approche du package ne sera pas identique. Dans le premier cas, on pourra alors choisir la distribution des sources permettant une compatibilité plus grande avec les distributions, à condition que l'utilisateur sache les récupérer et les compiler.

Dans le cas du binaire, il faut générer les binaires pour chaque système, ou tout du moins, vérifier la bonne compatibilité de celui-ci. Quand un client demande un développement spécifique sur Linux, souvent, il imposera sa distribution ou harmonisera avec une unique distribution. Cela simplifiera grandement le déploiement.

Linux : un marché en puissance ?

Si aujourd'hui Linux sur le serveur est commun notamment grâce au LAMP, qu'en est-il ailleurs ? *“Le développeur sur Linux croît, mais lentement. Le poste de travail reste en retrait”,* commente A. Dupuis. En entreprise, le poste de travail Linux commence à se répandre d'où la bonne dynamique actuelle des SSII open source et classiques. Or tant que le poste de travail, ou que l'ordinateur Linux ne se développe pas plus auprès du grand public,

les développements d'application Linux auront du mal à augmenter, même si le marché connaît une croissance surtout auprès des grandes entreprises, des administrations. Mais si le nombre d'utilisateurs n'augmente pas significativement, les éditeurs ne porteront pas leurs outils, faute d'un marché suffisant...

Les outils s'améliorent

Une des critiques récurrentes était le manque d'outils graphiques à l'instar d'un Delphi, Visual Studio. *“Longtemps ce fut une faiblesse. Mais aujourd'hui, nous disposons d'IDE performants : Eclipse, Qt Creator, Mono...”*, commente Anne Nicolas. Effectivement, côté IDE, le développeur sur Linux dispose de très bons IDE pour la plupart des développements. Le seul point faible demeure le développement web avec l'absence d'un équivalent à Dreamweaver et d'outils de design comparables à Creative Suite d'Adobe et Expression de Microsoft. Cependant, même si on développe en C#, ASP.Net ou Flex, les outils existent. On dispose de Mono et de Moonlight (équivalent de Silverlight). Et Adobe propose depuis quelques mois les versions Linux de Flex et de Air.

Comment passer à notre Tux adoré ?

Il faut bien entendu se former ou s'auto former. Il est très important de bien comprendre la structure de Linux, comment fonctionne la plateforme, les modèles de développement, l'offre logicielle, le packaging applicatif (et les différences entre les différents formats). Des éditeurs, comme Novell ou Mandriva proposent des cycles de formations et/ou des ressources dédiés aux développeurs.



Venez dialoguer avec l'intégrateur Open Source de référence

CONSEIL - SOLUTIONS - FORMATION - HOSTING

Sur le stand Alter Way (F30)

Alter Way
Silver sponsor
solutions
linux
opensource

- Keynote
- 3 Tables rondes
- 8 conférences

→ Des sessions technos live

PHP

- Entraînement certification PHP
- Les nouveautés de PHP 5.3
- Retour d'expérience : mise en place d'un projet critique avec PHP
- Zend Platform
- Comment installer Ingres et le driver PHP
- Sécurité PHP

Web

- Ergonomie et accessibilité, les bonnes pratiques d'un projet Web
- Le Web plus riche avec Ajax
- Délivrance, séparation de l'intégration et du développement Web
- Plone, l'outil de CMS le plus complet du Web

Cycle Business Intelligence Open Source

- ETL Talend
- Reporting en Open Source
- Use case : Intégration de briques BI Open Source dans un portail d'entreprise Liferay

Gestion de contenu Open Source

- Plone 3
- eZ Publish : Tour d'horizon du produit et des nouveautés eZ Publish 4.1

Messagerie d'entreprise

- Remplacez Ms Exchange par OpenXchange
- OpenXchange en SAAS

Infrastructure et sécurité

- Retour d'expérience sur la virtualisation
- Haute disponibilité avec MySQL
- Protégez votre Système d'Information avec Astaro

→ Un espace recrutement

→ Des événements

1/04 à 12 h
Remise du trophée OpenXchange

1/04 à 18 h
Avec la communauté MySQL - LeMug.fr
→ Retour d'expérience : MySQL chez Yahoo
→ Nouveauté MySQL : MySQL Proxy, Partitionnement
→ Haute disponibilité avec MySQL

et d'autres événements encore



→ Des entraînements certification PHP

→ Des interlocuteurs disponibles pour vous écouter et répondre à vos questions



www.alterway.fr

OPEN SOURCE

Le libre, autrement

Alter Way



Portrait de développeur

Michel Loiseleur est responsable technique OSSA (Linagora). Développeur et utilisateur Linux et open source, il nous dévoile ses outils, son quotidien.

Programmez ! : comment aborder le développement d'une application Linux ? De quelle manière choisir les outils, les librairies, voire sa distribution ?

Michel Loiseleur : Le choix est vaste lorsque l'on veut développer une application Linux. Rien que pour les toolkits graphiques, on dispose de Gtk, Qt, Tcl/Tk, Mono ou wxWidgets. Ces toolkits sont accessibles via des bindings pour la majorité des langages de programmation. Il est de plus en plus courant de voir sur un poste de travail Linux une application codée en plusieurs langages, avec un coeur en C/C++ et des extensions ou une partie graphique codée dans un langage de script comme le Python ou le Ruby. On peut citer comme exemple Amarok ou Battle For Wesnoth. Concernant les outils, CMake a fait ses preuves et est la référence pour tout nouveau projet

qui ne veut pas perdre son temps sur des problématiques de compilation.

P. ! : Faut-il cibler la distribution, les librairies, les versions, l'interface, pour éviter tout problème de compatibilité ?

M. L. : C'est effectivement plus simple, surtout lorsque l'on procède uniquement à la vente de binaires. Mais ce n'est pas obligatoire, on peut tout à fait inclure les bibliothèques sensibles dans son paquet binaire. Pour les toolkits graphiques, la rétro-compatibilité binaire est assurée sur l'ensemble des distributions. Autrement dit, une application Gtk 2.12 fonctionnera sur l'ensemble des distributions Linux disposant d'au moins cette version de Gtk. C'est identique pour une application Qt.

P. ! : Aujourd'hui, le poste de développement Linux se renforce avec l'arrivée de Flex, Air, Moon, Moonlight, etc. quelles sont encore les absences d'outils qui peuvent gêner le développeur ?

M. L. : Je n'ai jamais eu à utiliser un outil sous Windows pour développer.

Même les tests de compatibilité avec Internet Explorer ou Chrome peuvent se faire sous Linux, grâce à Wine qui a fortement évolué depuis le soutien de Google sur le projet. Un autre outil très populaire s'appelle rdesktop, il permet d'établir une connexion graphique avec une machine Windows, en utilisant la technologie Terminal Server.

P. ! : on parle souvent de petits problèmes de compatibilité entre les distributions, comment les éviter ?

M. L. : Pour les bibliothèques, il est recommandé d'être compatible avec les versions empaquetées sur les debian stables. Elles représentent 20 % du parc, et sont généralement les moins en avance en terme technologique. Elles constituent ainsi une base de référence afin d'obtenir une application qui soit déployable et fonctionnelle sur un grand nombre de distributions.

P. ! : en équipe, vaut-il mieux définir un master d'outils ou laisser les développeurs monter leur poste ?

M. L. : Il vaut mieux laisser les développeurs monter leur poste. Par contre, rien n'empêche de mutualiser les configurations en fournissant des modèles partagés pour les environnements populaires.

P. ! : Quelle distribution, conseillez-vous, celle qui vous paraît la plus stable ?

M. L. : Les Ubuntu sont très populaires. Personnellement, je conseillerai les Fedora, qui sont généralement plus intégrés et graphiquement plus esthétiques. La Fedora est souvent considérée comme instable, mais sa forte communauté fait que quelques mois après la sortie d'une version majeure, elle est tout aussi stable que ses concurrentes.

“Les tests de compatibilité avec Internet Explorer ou Chrome peuvent se faire sous Linux, grâce à Wine”

Générer sa propre distribution OpenSuSe en ligne

Novell héberge un service particulièrement intéressant : Suse Studio. Pour faire simple, il permet de créer sa propre distribution SuSe en ligne puis de générer le système que l'on peut ensuite distribuer avec son application, ses données, ses utilisateurs, etc. Particulièrement simple d'utilisation, on peut configurer une base MySQL dans l'interface, mettre en place un autostart d'application ou encore inclure des scripts. Le build peut avoir le format iso, d'une machine virtuelle ou encore d'un Linux Live. Et bientôt, il sera compatible avec le cloud computing d'Amazon !

Suse Build est très pratique quand on souhaite livrer un master prêt à déployer avec

des applications spécifiques, ses projets pour son entreprise, des clients ! D'autre part, toujours pour aider le développeur, l'administrateur, Novell propose un autre service, le Build Service. Il permet de générer rapidement des packages pour les distributions Linux en gérant les problèmes de dépendances, de versions des librairies, etc. Le service Build s'interface avec le SuSe Studio, pratique pour récupérer d'emblée des packages. D'autre part, on peut mettre à jour soi-même les packages grâce à un outil en ligne de commande. Compatible LSB.

Sites : http://en.opensuse.org/Build_Service
<http://susestudio.com/>





Bien débiter le développement sous Linux

Qu'entend-on exactement par Linux ? Le terme est devenu flou et désigne tantôt uniquement le noyau (kernel), à la base du système complet, et tantôt la distribution, à voir comme une compilation de logiciels adjoints à ce noyau, et mis en forme pour répondre à différents besoins. La première étape lorsque l'on passe dans l'univers linuxien est le choix de sa distribution.

La distribution GNU/Linux Ubuntu nous semble tout indiquée pour bien débiter. Ubuntu est une distribution stable, facile d'accès pour les débutants car orientée interface graphique, et qui propose un nombre de logiciels conséquent permettant un emploi instantané. Cet article traitera donc de la programmation pour les utilisateurs de cette distribution.

Premiers pas

L'utilisation de la console est intéressante par bien des points. Elle permet à l'utilisateur d'exécuter des commandes qu'il va lui-même entrer. Bien que cela puisse rebiter dans un premier temps un utilisateur venant du monde Windows, très graphique, cette pratique représente très souvent un gain de temps considérable. Prenons comme exemple la copie de fichiers. Si vous souhaitez réaliser une copie de façon "graphique", vous devez vous déplacer jusqu'au répertoire contenant votre fichier, cliquer sur celui-ci pour le copier, puis enfin vous déplacer jusqu'à la destination pour le coller. Pour réaliser cela par le biais de la console il vous suffit d'entrer la commande "cp <nom du fichier à copier> <destination>". La console est disponible graphiquement via un terminal, qui émule la console. Vous trouverez ce terminal dans le menu Applications -> Accessoires -> Terminal. Vous pourriez penser qu'écrire entièrement la destination du fichier est plus long que de cliquer directement sur les différents répertoires, et c'est là que l'autocomplétion intervient.

L'autocomplétion permet à un utilisateur, lors de la saisie d'une commande, de limiter la quantité d'informations à saisir avec son clavier. Ainsi pour copier un fichier dans le répertoire document, il vous suffit d'entrer la lettre "d" et d'appuyer sur la touche "tab" de votre clavier, touche permettant l'autocomplétion par défaut, et le mot "document" s'auto complétera. Si toutefois, un autre répertoire commençant par la lettre "d" se trouvait au même endroit, l'appui répété sur la touche "tab" permet de switcher entre les différentes possibilités. Parmi toutes les commandes existantes, une est à connaître sans doute avant les autres. Il s'agit de la commande "sudo" permettant à l'utilisateur de lancer la commande qui suivra en tant qu'administrateur, ce qui sera indispensable pour l'installation de nouvelles applications. Un administrateur possédant plus de droits que l'utilisateur lambda, l'accès à tous

les dossiers lui sera autorisé, en l'occurrence l'accès au dossier dans lequel se font les installations.

Installation

L'installation de nouveaux logiciels sous Ubuntu, et sous Linux en général, est des plus simples. Il existe différents moyens sous Ubuntu d'installer un logiciel "packagé". Le premier moyen est d'utiliser Aptitude ou Synaptic, le second étant, comme sous Windows, de rechercher ces "packages" sur Internet et de les télécharger pour les installer. Aptitude est un gestionnaire de paquets permettant l'installation, la suppression et la mise à jour de logiciels. Il est utilisable en ligne de commande ou avec une véritable interface interactive. En ligne de commande, il vous faudra utiliser la ligne suivante "sudo aptitude install nom_du_paquet". Ici l'application Aptitude prend en argument "install", indiquant que nous souhaitons procéder ici à l'installation d'un nouveau paquet. "Sudo" quant à lui, permet de lancer l'application Aptitude en tant qu'administrateur comme expliqué précédemment. Bien sûr, cela nécessite de connaître le nom exact du paquet que l'on souhaite installer. Utiliser Aptitude par son interface graphique permet la recherche de paquets. Lancez donc Aptitude sans argument, soit "sudo aptitude", puis utilisez la touche "/" pour exploiter les fonctions de recherche de paquets. Comme le suggère le menu d'aide, utilisez ensuite la touche "+" pour ajouter un paquet à la liste des paquets à ajouter, puis la touche "g" pour télécharger / installer.

Synaptic quant à lui est entièrement graphique. On peut l'assimiler à l'utilitaire d'ajout/suppression de programmes sous Windows. Il permet de rechercher des paquets et propose aussi d'installer les dépendances pour chacun d'entre eux. Un paquet peut en effet avoir besoin d'autres paquets pour fonctionner, comme par exemple des bibliothèques particulières. Le lancement de Synaptic peut se faire via l'entrée du menu System -> Administration -> Synaptic Package Manager. Quel que soit le gestionnaire de paquets utilisé, l'installation se fait immédiatement après le téléchargement.

Dans le dernier cas - téléchargement du paquet par un site Internet -, il suffit la plupart du temps d'extraire le contenu du dossier compressé pour que le logiciel soit directement utilisable. Il arrivera parfois que l'utilisateur



se retrouve en présence des sources du logiciel, et non pas d'un fichier binaire directement exécutable. L'utilisateur devra alors compiler ce code source, c'est-à-dire générer les fichiers binaires requis. Pour ce faire, la solution utilisée dans la très grande majorité des cas est une solution à base de Makefile. Un Makefile est généralement un fichier appelé " Makefile " directement exploité par la commande " make " pour exécuter un ensemble d'actions, comme la compilation d'un projet, l'archivage de fichiers, la mise à jour d'un site, etc.

Notre helloworld

C'est au travers de la création d'un projet basé sur le langage de programmation " C " que l'initiation à ces makefiles se fera. Nous allons réaliser un projet bien connu puisqu'il s'agit d'un classique " Helloworld ". Un éditeur de texte sera indispensable et je vous conseille Emacs pour commencer, outil phare de l'univers du Libre, ainsi que l'utilisation de la console pour prendre de bonnes habitudes. Ouvrez un premier fichier " hello.c " qui contiendra le corps de la fonction d'affichage en lançant la commande " emacs hello.c ", et entrez -y les lignes suivantes :

```
void Hello(void)
{
    printf("Hello World\n");
}
```

Et sauvegardez grâce à la combinaison de touches Ctrl+x, Ctrl+s. Ouvrez ensuite un second fichier nommé hello.h, celui qui contiendra le prototype de votre fonction, c'est-à-dire les lignes suivantes :

```
#ifndef HELLO_H
# define HELLO_H

void Hello(void);

#endif
```

Finalement, afin de conclure notre projet " Helloworld " en C, ouvrez un fichier main.c, fichier contenant notre fonction principale " main " et qui effectuera un appel à notre fonction " Hello " :

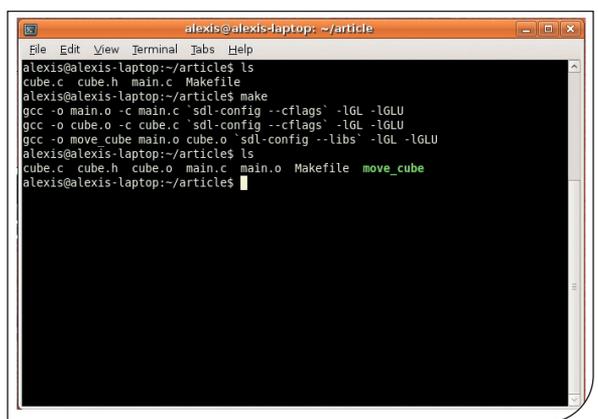
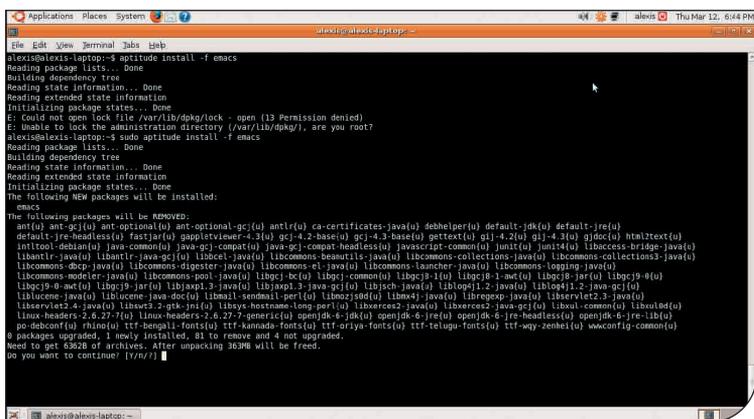
```
#include "hello.h"

int main(void)
{
    Hello();
    return 0;
}
```

Maintenant que le projet est fini, il nous faut créer un makefile afin de le compiler et d'obtenir un binaire (l'équivalent des ".exe" sous Windows). Ouvrez un fichier que vous nommerez " Makefile " et insérez-y les lignes suivantes :

```
hello: hello.o main.o
    gcc -o hello hello.o main.o
hello.o:
    gcc -o hello.o -c hello.c
main.o:
    gcc -o main.o -c main.c
```

Les trois mots devant les " : " sont des règles, à comprendre comme "commandes". La première commande " hello " requiert ici que les commandes hello.o et main.o soient exécutées avant elle pour pouvoir fonctionner. Passons donc à la règle de construction hello.o, celle-ci utilise le compilateur libre gcc assurant la compilation des fichiers C. À partir du fichier " hello.c ", spécifié ici par l'option " -c ", gcc va créer un fichier objet (langage machine). Ce fichier aura pour nom celui précisé par l'option " -o ". Il en va de même pour la commande main.o. Une fois ces deux commandes lancées avec succès, la commande " hello " peut alors s'exécuter. Une fois de plus le compilateur Gcc est utilisé, cette fois-ci pour créer notre binaire final nommé " hello " en assemblant les deux fichiers objets générés précédemment. Une fois le makefile sauvegardé, lancez la commande " make " dans votre console, suivie de la commande " ls ", permettant de lister les répertoires et fichiers, afin de voir apparaître votre binaire " hello " correctement construit dans votre répertoire. Il vous suffit maintenant de lancer votre binaire via un simple appel à la commande " ./hello " qui invoquera votre programme du même nom, le point " . " désignant le



répertoire courant. L'outil make permet aussi de lancer seulement certaines commandes parmi toutes celles définies au sein du fichier (les fameuses " règles "). En spécifiant simplement le nom de la commande voulue après " make " comme par exemple " make main.o ", qui, tapée dans la console, ne génère que le fichier " main.o ". Libre à vous de rajouter vos propres règles. On peut imaginer par exemple une commande clean telle que ci-dessous :

```
clean:
  rm -rf *.o
```

où " rm " (remove) permet d'effacer ici tout les fichiers d'extension " .o " dans l'optique de nettoyer le répertoire quand on le souhaite via un simple appel à " make clean ". Il y a plusieurs avantages à utiliser des fichiers makefile plutôt que de rentrer les commandes de compilation directement. Le simple fait d'écrire ces lignes de compilation une fois pour toutes et de ne pas avoir à les ré-entrer à chaque modification du code en est un parfait exemple. L'autre avantage souvent mis en avant est le mécanisme interne faisant que seuls les fichiers qui le nécessitent (ayant subi des modifications) sont recompilés. Cet avantage pouvant paraître mineur à première vue devient déterminant lorsque l'on travaille sur un fichier avec plusieurs milliers de fichiers.

Développement d'un jeu vidéo

Voici venu le moment de faire vos débuts dans la création d'un jeu vidéo. Maintenant que nous savons créer un projet et comment le compiler, pourquoi ne pas entamer l'utilisation de bibliothèques permettant d'obtenir des résultats plus graphiques ? Conjointement au langage C nous allons ici exploiter la bibliothèque SDL qui facilite la création d'applications multimédias en deux dimensions. Nous allons utiliser cette bibliothèque pour effectuer la gestion des événements (pressions sur les touches du clavier) et l'illustre bibliothèque OpenGL pour la génération de tous nos objets en trois dimensions.

La bibliothèque SDL - pour Simple DirectMedia Layer - permet un accès rapide et simple à une sortie vidéo tout en permettant de gérer le son, le clavier et la souris. Nous sommes alors capables d'intercepter pendant l'exécution de notre jeu l'appui sur une touche du clavier ou un clic de souris, pour pouvoir agir en conséquence, une fois cet événement identifié. On peut donc imaginer la traditionnelle gestion des mouvements via les touches directionnelles de notre clavier. Quant à OpenGL (Open Graphics Library) il s'agit d'une interface de programmation (API) proposant un nombre de fonctions assez conséquent pour créer des objets aussi bien 3D que 2D.

Afin de mieux comprendre comment SDL permet d'agir sur le contenu de votre jeu, c'est-à-dire les différents éléments créés grâce à OpenGL, une explication technique s'impose. SDL va être le moteur de votre jeu, elle va gérer l'affichage de votre fenêtre de jeu ainsi que les différents événements. Techniquement, cette gestion se fait à l'aide

PRENEZ DE LA HAUTEUR

Web > PHP

Formation PHP Expert



"Montez en puissance sur PHP et préparez vous à passer la certification PHP"

Au Programme des 5 jours :

- * Certification PHP et rappels
- * La programmation orientée objet PHP 5
- * Les bases de données et PDO
- * XML et les services Web
- * Flux et réseau
- * Sécurité
- * Optimisation des performances
- * Debug et audit de code



Prochaines sessions

Paris 2009
20 Avril, 11 Mai, 08 Juin
6 et 27 Juillet

Lyon 2009
20 Avril, 6 et 27 Juillet

Tarif
1995 € HT

LE SPECIALISTE DE LA FORMATION POUR L'OPEN SOURCE

Informations

01 45 28 09 82

www.anaska.com

anaska

Alter Way GROUP

AUTHORIZED
MySQL.
EDUCATION CENTRE



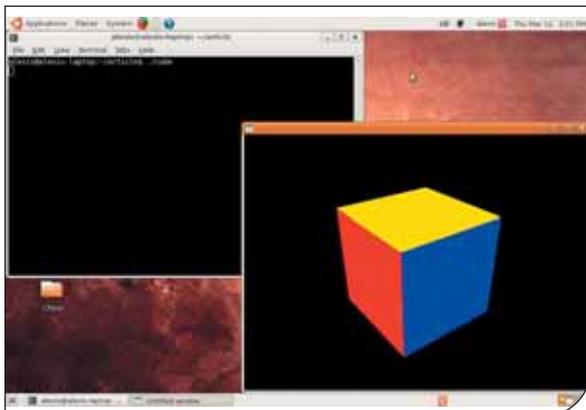
d'une grande boucle, qui se termine quand vous le désirez, le mieux bien sûr étant à la fermeture de l'application. Chaque tour de boucle va correspondre à un rafraichissement de vos éléments OpenGL : mettre à jour leur apparence, leur position, leur angle d'inclinaison... Durant chaque tour de boucle nous vérifions si un événement s'est produit ou non, afin de pouvoir effectuer les modifications puis de tout réafficher. Nous pouvons par cette technique faire naître une impression de mouvement de différents objets.

Commençons par créer un cube grâce à OpenGL. Utilisons pour cela la fonction `glBegin(GL_QUADS)` qui indique que nous allons tracer un carré pour dessiner la première face.

```
glBegin(GL_QUADS) ;

glColor3d(1.,0.,0.);
glVertex3i(1,1,1);
glVertex3i(1,-1,1);
glVertex3i(-1,-1,1);
glVertex3i(-1,1,1);
```

Nous utilisons la fonction `glColor3d` afin de colorer la face du cube. Comme vous pouvez le remarquer, cette fonction s'utilise sur une face du cube, il est donc possible d'obtenir un cube possédant des faces de couleurs différentes au final. Les trois paramètres de cette fonction correspondent aux trois couleurs primaires RGB, définies par des nombres flottants représentant l'intensité de chaque composante. Le triplet (1,0,0) donnera ici une face rouge. Viennent ensuite quatre fonctions `glVertex3i` quiinstancient des points de l'espace, représentant les quatre coins de notre face. Chacun de ces points est représenté par trois coordonnées. Effectivement, nous avons pour but de créer un cube, un élément 3D. Le repère dans lequel se trouvera ce cube étant en 3D, nous devons préciser chacune des coordonnées de chaque point. Les trois valeurs en paramètre de la fonction `glVertex3i` sont les valeurs de *x*, *y* et *z* dans l'espace. L'opération ci-dessus devra donc être répétée six fois, afin de créer les six faces carrées composant un cube. OpenGL n'offre pas seulement la possibilité d'associer une couleur à une face. Il est bien sûr possible d'y associer une image,



une texture, afin d'obtenir des éléments de décor réaliste.

Vive OpenGL !

A chaque tour de boucle, votre cube sera redessiné à l'écran. Il n'est pas nécessaire de toujours rafraichir tous les éléments composant votre décor : on peut imaginer ne rafraichir alors que les éléments capables de se mouvoir afin de ne pas ralentir le jeu et éviter les calculs inutiles. Mais pour le moment, notre cube reste inanimé car les paramètres fournis à notre fonction de dessin sont inchangés au cours du temps. Au lieu d'utiliser la fonction `glVertex3i(1, 1, 1)` avec des valeurs prédéfinies, il nous faudrait donc faire usage de variables, et si l'originalité ne vous effraie pas, je propose `glVertex3i(x, y, z)`. Il est donc maintenant possible de modifier les valeurs de *x*, *y* ou *z* à chaque tour de boucle, afin de redéfinir les coordonnées de chaque point, et de redessiner le cube à un emplacement différent dans l'espace, en créant ainsi l'illusion du mouvement.

Il reste maintenant à récupérer une pression sur une touche du clavier, et d'associer ce geste au mouvement d'un objet. La bibliothèque SDL propose pour cela un ensemble de macros permettant de savoir quelle action l'utilisateur est en train de réaliser. Si nous voulons faire avancer notre cube à l'aide de la flèche " haut " de notre clavier, il nous suffit de tester à chaque tour de boucle que la touche " up " est pressée pour augmenter la profondeur (*y*) de 1 et ainsi voir le cube s'éloigner à chaque rafraichissement. On obtiendrait une boucle de ce type :

```
Case SDL_KEYDOWN :
switch (event.key.keysym.sym){
case SDLK_UP:
y += 1;
...
draw_cube(x,y,z);
```

Cela signifie que pendant le tour de boucle, si une touche est enfoncée, et qu'il s'agit de la touche " up " du clavier, alors la variable " *y* " est augmentée de 1. A la fin de la boucle, le cube est redessiné avec les nouvelles coordonnées, et notre cube se déplace alors dans l'espace.

Une fois notre cube capable de se mouvoir, il nous faut compiler notre projet comme nous l'avons fait précédem-





ment. Pour cela, il va falloir relier les bibliothèques SDL et OpenGL à notre projet lors de la compilation afin que toutes les fonctions et macros que nous avons utilisées soient reconnues. Pour nous faciliter la tâche, la bibliothèque SDL nous fournit deux commandes bien utiles : "sdl-config -cflags" qui servira lors de la création des différents fichiers objets et la commande "sdl-config -libs" qui servira lors de la création de notre binaire final. En ce qui concerne OpenGL, nous utiliserons l'option -l permettant de relier une bibliothèque à notre projet de la manière suivante : -lGL. Si votre code ne se trouve que dans un seul fichier C, nous devrions obtenir un makefile de ce type :

```
CFLAGS=`sdl-config -cflags`  
LDFLAGS=`sdl-config -libs`  
  
all : main.o  
    gcc -o cube main.o $(LDFLAGS) -lGL -lGLU  
main.o:  
    gcc -o main.o -c main.c $(CFLAGS) -lGL -lGLU
```

"CFLAGS" et "LDFLAGS" sont ici deux variables définies au début du makefile afin d'obtenir des commandes plus facilement compréhensibles au premier coup d'œil ; les bibliothèques GL et GLU seront reliées à notre projet comme nous le souhaitons.

Il peut arriver que vous ayez besoin de faire fonctionner votre jeu, ou tout autre programme, sur d'autres distributions. Les problèmes seront rencontrés lors de la compilation sur d'autres distributions que celle sur laquelle votre

projet a été développé. Des bibliothèques peuvent être situées à des endroits différents sur le disque dur par exemple. C'est ici qu'intervient l'utilité d'un fichier de configuration, nommé configure, en plus de votre makefile. Ce fichier configure est un script exécutable permettant de créer un fichier d'options pour le makefile, souvent nommé Makefile.rules, dont le contenu diffèrera en fonction des distributions sur lesquelles votre projet sera compilé. Ce fichier "configure" pourrait faire la différence entre certaines architectures, et ainsi créer des variables adaptées en fonction de la machine sur laquelle nous nous trouvons. Le Makefile.rules créé n'a plus qu'à être inclus dans notre Makefile par le biais de la ligne "include Makefile.rules" placée au début de votre Makefile. Si nous voulions porter notre projet de jeu sur une distribution Linux telle que Fedora, nous n'aurions pour ce projet pas de changements à effectuer, mais dans le cadre d'un projet plus massif, nécessitant de nombreuses bibliothèques ou d'autres programmes, il peut être utile que le système possède bien tous les pré-requis.

Les portes du jeu vidéo vous sont maintenant ouvertes. Je laisse votre imagination s'occuper des idées de jeux, de décors ou de jouabilité que vous recherchez. Il est bien sûr possible d'utiliser SDL et OpenGL avec d'autres langages que le langage C, je laisse à votre convenance le choix de celui avec lequel vous vous sentez le mieux.

■ Alexis Ageorges - alexis.ageorges@gmail.com

Etudiant à l'EPITA, spécialisation Multimédia et Technologie de l'information.

Les savoirs du **D**éveloppeur



9782100518319 • 368 pages • 38 €

9782100515721 • 312 pages • 34 €

9782100526741 • 464 pages • 39 €

9782100523368 • 560 pages • 38 €

Tous nos ouvrages sont disponibles en librairie

Catalogue complet  dunod.com


DUNOD
ÉDITEUR DE SAVOIRS



Développement rapide avec GNOME/GTK+ et Python

Gnome est une des deux plates-formes les plus populaires sous Linux, avec KDE, il est disponible par défaut sur la plupart des distributions les plus utilisées, comme Ubuntu, Red Hat, Fedora, OpenSuse et Debian. Le toolkit GTK+ est au cœur de l'environnement GNOME. Il fournit tous les éléments nécessaires à la génération d'une interface graphique pour une application : les fenêtres, les boutons, les listes, etc. GTK+ n'est pas un langage de programmation en soi, c'est un ensemble d'API, utilisables avec plusieurs langages de programmation, au travers de ce qu'on appelle des "bindings".

GTK+ et les autres briques de la plate-forme GNOME sont écrits en C à la base. Après, les développeurs font le nécessaire pour utiliser ces API dans le langage de programmation de leur choix. Un développeur Java peut ainsi utiliser la plate-forme GNOME à travers les bindings Java-GNOME, un développeur C# utilisera le binding GTK#, un développeur C++ peut utiliser la famille de bindings gtkmm (GTK minus minus) et ainsi de suite. Chaque binding respecte les normes d'utilisation du langage. Le binding Python de GTK+ s'appelle PyGTK. Pour qui connaît un peu GTK+, les normes de nomenclature sont assez simples. La classe GtkImage, par exemple, est dans l'interface Python comme gtk.Image. La fonctionvoid

```
gtk_image_set_from_file(GtkImage *image, const gchar *filename)
```

se transforme naturellement en

```
gtk.Image.set_from_file(filename)
```

Dans cet article, nous allons voir comment utiliser les outils GNOME, et la plate-forme de développement, pour développer rapidement des applications simples, permettant des cycles de développement très courts, préconisés dans des méthodologies de développement comme SCRUM ou XP. Ces outils sont parfaitement adaptés au développement des premiers prototypes d'applications, même complexes, grâce aux retours d'expériences rapides. Nous allons créer un éditeur de texte simple. La première étape sera la création de l'interface avec l'outil Glade. Glade génère un fichier XML qui définit l'interface, que nous allons charger par la suite dans notre application, avant de connecter des signaux à des callbacks, afin d'implémenter quelques fonctionnalités de base. Le grand avantage de l'utilisation de Python est bien sûr de n'avoir pas besoin de faire un processus de compilation. Comme c'est un langage interprété, nous exécutons notre application Python simplement avec la ligne de commande :

```
$ python myscript.py
```

ou, comme il est normal de faire sous Linux, de rajouter la première ligne

```
#!/usr/bin/python
```

et de rendre exécutable le fichier myscript.py avant de l'exécuter directement sur la ligne de commande :

```
$ chmod +x myscript.py
```

Les pré-requis

Sur Ubuntu, outre les paquets installés par défaut, je vous conseille d'installer les paquets suivants :

- python-gtk2
- python-gtk2-doc
- python-gtk2-tutorial
- python-gnome2-extras
- python-gnome2-extras-doc
- devhelp.

python-gtk2 et python-gnome2-extras installeront l'ensemble des bibliothèques et modules nécessaires pour créer des applications GNOME en Python. devhelp est un navigateur et lecteur de documentation développeur qui vous installera l'ensemble des documentations de la plate-forme GNOME. C'est un outil indispensable pour le développeur d'applications GNOME. Les modules python-gtk2-doc, python-gtk2-tutorial et python-gnome2-extras-doc installent des documentations pour les bindings Python. Après l'installation, ils seraient intégrés automatiquement dans le navigateur devhelp. Malheureusement pour les développeurs français, la plupart de ces documents ne sont disponible qu'en anglais.

Construire l'interface graphique

Pour créer notre interface graphique, nous allons utiliser l'outil Glade. Glade est l'application de création d'interfaces GNOME par défaut. Depuis la version 2.12, GTK+ inclut une nouvelle fonctionnalité pour charger dynamiquement des interfaces graphiques, GtkBuilder. Auparavant, une bibliothèque séparée, Libglade, a servi au même effet, mais cette bibliothèque n'est plus officiellement encouragée. Malheureusement, les formats de fichiers utilisés par GtkBuilder et Libglade ne sont pas 100% compatibles, et comme GtkBuilder est très récent, au moment où j'écris ces lignes, la version stable de Glade ne génère pas encore de fichiers en format GtkBuilder. Nous allons donc utiliser un outil inclus dans les versions de GTK+ récentes, gtk-builder-convert, pour convertir les fichiers de type Libglade en fichiers de type GtkBuilder. La pro-





Le Meilleur du Test de Performance Web en toute simplicité et au meilleur rapport qualité-prix



- Développement automatisé de Cas-Tests et Test de Charge
- Rapports d'analyse des Tests de performance et de charge
- Analyse des Serveurs et de l'impact sur les performances
- Interface utilisateur et Rapports en Français

Découvrez comment Web Performance Suite peut faire gagner vos applications Web en Qualité, Fiabilité et Performance

Evaluation gratuite de Web Performance Suite sur <http://www.kapitec.com/Pub/WP?id=118>

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



Power Your Web Projects

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



chaîne version de glade, 3.5.0, qui devrait voir le jour en mars 2009, générera nativement des fichiers GtkBuilder. [Fig.1]

Au premier lancement, nous voyons un espace vide où nous allons créer notre application, avec sur la gauche l'ensemble des widgets GTK+ dont on dispose. La documentation sur les widgets GTK+ est très complète. Pour résumer la philosophie GNOME très rapidement, il y a les widgets de type containers, et les autres. Les fenêtres sont des widgets de type GtkWidget, eux-mêmes des containers, qui apparaissent au niveau du gestionnaire de fenêtres. À l'intérieur de ces widgets containers, nous pouvons disposer d'autres widgets de tout type: des boutons, des libellés, des widgets pour l'entrée des données comme le texte ou les numéros, des images, et ainsi suite. Nous pouvons également y mettre d'autres containers, qui à leur tour contiendraient des widgets. Pour chacun de ces widgets, il y a des signaux possibles qui vont être envoyés lorsque nous agissons sur le widget. Par exemple, pour un widget de type "GtkButton", les signaux suivants sont émis sur des événements clavier ou souris: "activate", "clicked", "enter", "leave", "pressed" et "released".

Pour chaque signal, nous avons le choix de laisser le moteur GTK+ gérer le signal, ou d'intercepter nous-mêmes le signal dans notre interface, avec une fonction qu'on appelle un "call-back". Pour l'instant, démarrons notre application. Dans la partie "Niveaux supérieurs", nous avons une sélection de fenêtres de premier niveau possible, dont une fenêtre de sélection de police ou de sélection de couleurs. Pour notre application, nous allons choisir la fenêtre la plus simple. Pour disposer des widgets à l'intérieur d'une fenêtre il est nécessaire d'utiliser des widgets containers. Les containers les plus couramment utilisés sont des GtkVBox (boîte verticale) et le GtkHBox (boîte horizontale). Nous allons insérer une boîte verticale dans notre fenêtre,

avec 3 éléments. [Fig.2]

Dans ce VBox, nous allons rajouter une barre de menus en haut, une vue avec défilement qui contient une vue de texte au milieu, puis une barre d'état en bas, pour terminer



Figure 2: Insertion de GtkVBox

notre fenêtre principale. Notre menu de départ contient quelques entrées que nous n'allons pas utiliser, mais pour l'instant, nous pouvons les laisser telles quelles. À droite de notre fenêtre Glade,

nous voyons maintenant une jolie hiérarchie de widgets qui s'est créée, suite à notre manipulation. Puis, nous sauvegardons le tout sous le nom "ProgrammezText.glade". Une fois sauvegardé, nous pouvons le convertir en format GtkBuilder avec la commande :

```
$ gtk-builder-convert ProgrammezText.glade ProgrammezText.ui
```

Le code complet de l'application est disponible en téléchargement sur le site de Programmez. Maintenant, il est temps de charger notre fenêtre avec Python, et de regarder à quoi ressemble l'application. Avant de développer votre première application GTK+ en Python, assurez-vous que Python, PyGTK et toutes ses dépendances sont installées. Dans votre éditeur de texte préféré, commencez un fichier ProgrammezText.py.

```
#!/usr/bin/python

import gtk

class ProgrammezText:

    def __init__(self):

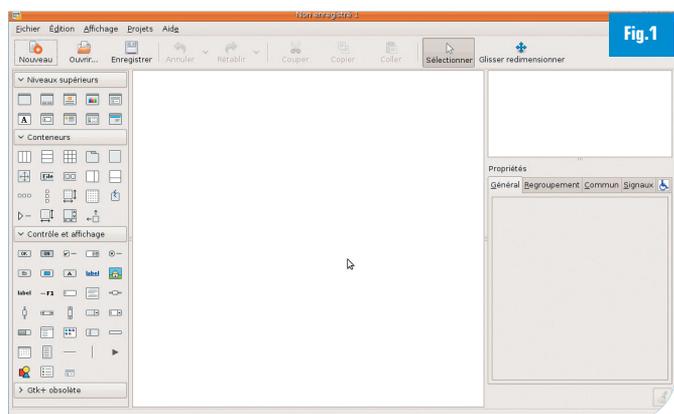
        builder = gtk.Builder()
        builder.add_from_file("ProgrammezText.ui")

        self.window = builder.get_object("window1")
        builder.connect_signals(self)

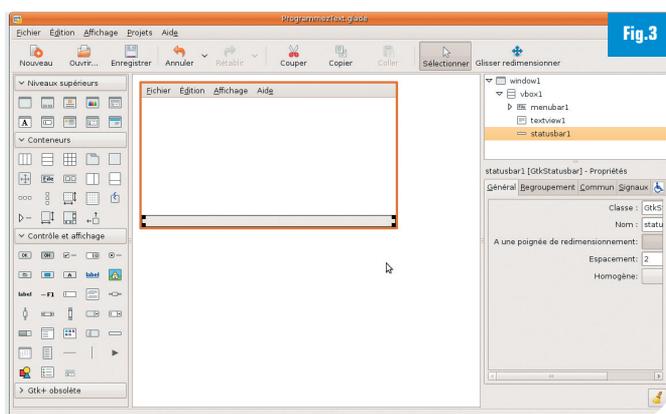
editor = ProgrammezText()
editor.window.show()

gtk.main()
```

Dans ce fichier, avec le constructeur de notre objet ProgrammezText, nous chargeons notre interface ProgrammezText.ui, puis nous initialisons notre fenêtre, avant de l'afficher et de passer la main à la boucle d'événements GTK+. Une fois lancée, l'interface s'affiche, mais quand nous fermons la fenêtre, notre application ne se termine pas. De la même façon, quoi qu'on fasse dans notre application, rien ne se fait au-delà de l'affichage. Pour rendre notre application utile, nous devons associer du code avec les signaux qui sont émis par des actions dans l'interface graphique.



Premier ouverture de glade



ProgrammezText : Notre application prend forme





Faire fonctionner l'interface

Dans Glade, sur la droite, dans l'onglet "Signaux", Nous pouvons voir toutes les signaux émis par le widget actuellement sélectionné. Notre fenêtre, "window1", contient un signal "destroy" dans la rubrique GtkWidget, qui est émis lors de la fermeture de fenêtre. Pour terminer proprement notre logiciel, il faut que nous appelions la fonction "gtk_main_quit()" à ce moment, ce qui permettra à GTK+ de détruire tous les objets qu'il gère et de quitter proprement l'application. Nous avons deux possibilités pour cela : en cliquant sur "<Saisir ici>", nous avons le choix entre plusieurs fonctions qui pourraient convenir, dont "gtk_main_quit" et un callback avec un nom spécifique du widget ("on_window1_destroy") qui est généré automatiquement. Si nous choisissons "gtk_main_quit" dans ce menu, nous n'avons plus rien à faire pour fermer proprement notre logiciel. [Fig.4]

Mais imaginons que nous voulions faire un véritable éditeur de texte, il faudrait avertir l'utilisateur lors de la fermeture d'un fichier modifié, ce qui nous oblige à avoir la main. Nous pouvons également employer cette même fonction lorsque l'utilisateur choisit l'entrée menu "Quitter". Nous décidons donc d'utiliser un callback spécifique, "text_quit", que nous saisissons à la main, pour accomplir cette tâche. Puis nous sauvegardons notre fichier Glade, et le convertissons encore une fois en fichier GtkBuilder.

Dans la classe ProgrammezText, rajoutons la fonction :

```
def text_quit(self, widget, data=None):
    gtk.main_quit()
```

Normalement, si tout va bien, notre application doit se terminer proprement maintenant.

Pour l'instant, j'ai laissé les noms des objets, et les noms de callbacks par défaut, mais vous pouvez bien sûr modifier les noms des objets, et mettre à la place vos propres noms de fonctions, selon les normes de programmation que vous utilisez.

Pour terminer cette première version, nous allons rajouter une fonctionnalité importante: la sauvegarde de fichier.

Sur l'entrée de menu "Fichier/Sauvegarde", je change le nom du

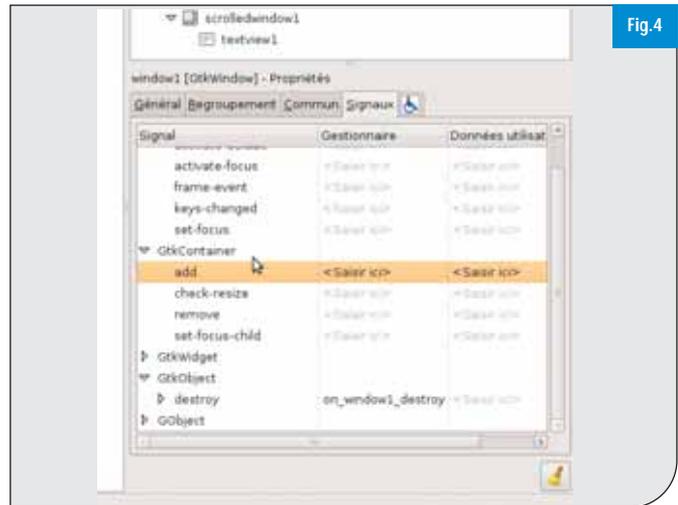


Fig.4

Les signaux dans Glade

Salon RECRUTEMENT
Lesjeudis.com
INFORMATIQUE - DEVELOPPEMENT - AMOR - FINANCE

+ de 100 000 postes
en **informatique**
à pourvoir par an

**Vous trouverez
forcément un emploi
près de chez vous !**



Renseignements et pré-inscriptions : <http://www.lesjeudis.com/salons-informatiques>



widget en "menu_save" pour que ce soit plus facilement repérable, puis j'y associe le callback "on_menu_save_activate". Ensuite, on sauvegarde, et on fait encore un passage par gtk-builder-convert. La fonction de sauvegarde utilise un *dialog* du sélecteur de fichiers pour permettre à l'utilisateur de choisir un répertoire et un nom de fichier, avant d'y sauvegarder les contenus de notre buffer de texte. Dans la méthode `__init__`, pour nous permettre de mémoriser le nom du fichier en cours, nous rajoutons une ligne :

```
self.filename = None;
```

Puis, nous rajoutons la méthode suivante à l'objet `ProgrammezText` :

```
def on_menu_save_activate(self, widget, data=None):
    filename = None;
    if self.filename == None:
        selecteur = gtk.FileChooserDialog("Sauvegarde...", self.window,
            gtk.FILE_CHOOSER_ACTION_SAVE,
            (gtk.STOCK_CANCEL, gtk.RESPONSE_CANCEL,
            gtk.STOCK_SAVE, gtk.RESPONSE_OK))
        reponse = selecteur.run()
        if reponse == gtk.RESPONSE_OK: filename = selecteur.get_filename();
        selecteur.destroy()

    if filename == None: return

    try:
        buf = self.textview.get_buffer()
        text = buf.get_text(buf.get_start_iter(), buf.get_end_iter())
        f = open(filename, "w")
        f.write(text);
        forcclose()

        self.filename = filename
    except:
        self.error_message("Erreur lors de sauvegarde de fichier %s" % filename)
```

D'abord, nous créons un *dialog* de type *FileChooser* (sélecteur de fichiers), qui héritera de notre fenêtre principale, et qui sera de type sauvegarde (sachant que plusieurs types de sélecteurs de fichiers sont possibles - ouvrir, sauvegarder, sélectionner un dossier, ou créer un dossier). Puis nous ajoutons deux boutons "Annuler" et "Sauvegarder", auxquels nous attachons des valeurs de réponses qui seront retourné par l'exécution du *dialog*. [Fig.5]

Par la suite, le *dialog* est lancé et affiché de façon modale (donc, bloquante pour l'application tant que nous n'avons pas terminé avec), et si l'utilisateur a choisi "Sauvegarder", nous récupérons le nom de fichier choisi.

Enfin, nous utilisons l'API de base pour notre text view (`Gtk.TextView`) pour récupérer les contenus du buffer, avant de l'écrire dans le fichier choisi, et d'enregistrer le nom du fichier dans le champ *filename* de notre éditeur de texte.

Si par malheur l'écriture ne peut pas se faire, nous nous protégeons, et nous montrons à l'utilisateur une erreur.

Il est facile d'activer de la même façon des autres fonctions de fichier comme l'ouverture de fichier, la sauvegarde lors de la fermeture, ou implémenter une fonctionnalité copier/coller. La suite est laissée comme exercice pour le lecteur enthousiaste !

Il est facile d'activer de la même façon des autres fonctions de fichier comme l'ouverture de fichier, la sauvegarde lors de la fermeture, ou implémenter une fonctionnalité copier/coller. La suite est laissée comme exercice pour le lecteur enthousiaste !

Conclusion

Le développement d'applications GNOME avec Python peut être très rapide, grâce à l'outil Glade et la capacité native de `GtkBuilder`. De plus, en utilisant Python, nous évitons un cycle de compilation onéreux, et nous pouvons constater tout de suite les résultats des changements effectués. Grâce à la quantité et qualité des tutoriaux existants pour `PyGTK` et la qualité de la documentation des APIs, il est possible d'être opérationnel très rapidement. GNOME contient des bibliothèques puissantes, comme `DBus` pour les messages inter-processus, `GStreamer` pour la gestion du multimedia, ou `GIO` et `gvfs` pour l'abstraction des systèmes de fichiers. Tous sont disponibles dans des applications Python, permettant la création rapide d'applications complexes.

■ Dave Neary - *Projet Gnome*

Ressources :

- [1] GNOME: <http://www.gnome.org>
- [2] KDE: <http://www.kde.org>
- [3] GTK+: <http://www.gtk.org>
- [4] Java-GNOME: <http://java-gnome.sourceforge.net/>
- [5] GTK#: <http://www.mono-project.com/GtkSharp>
- [6] gtkmm: <http://www.gtkmm.org/>
- [7] Python: <http://www.python.org/>
- [8] GTK+ API: <http://library.gnome.org/devel/gtk/stable/>
- [9] PyGTK documentation: <http://www.pygtk.org/reference.html>
- [10] GtkImage: <http://library.gnome.org/devel/gtk/stable/GtkImage.html>
- [11] gtk.Image: <http://www.pygtk.org/docs/pygtk/class-gtkimage.html>
- [12] gtk_image_set_from_file: <http://library.gnome.org/devel/gtk/stable/GtkImage.html#gtk-image-set-from-file>
- [13] gtk.Image.set_from_file: <http://www.pygtk.org/docs/pygtk/class-gtkimage.html#method-gtkimage--set-from-file>
- [14] Devhelp: <http://live.gnome.org/devhelp>
- [15] GTK+ documentation: <http://www.gtk.org/documentation.html>
- [16] PyGTK tutorial: <http://www.learningpython.com/2006/05/07/creating-a-gui-using-pygtk-and-glade/>
- [17] PyGTK presentation and example programs: <http://www.pixelbeat.org/talks/pygtk/>

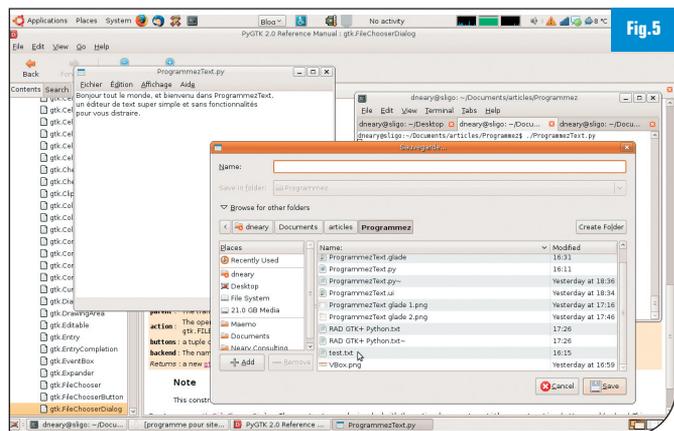


Fig.5

Notre application tourne !



VOUS AUSSI, DÉVELOPPEZ 10 FOIS PLUS VITE



DEMANDEZ LE DOSSIER GRATUIT : 244 pages + DVD + Version Express gratuite + 112 Témoignages.
Tél: 04.67.032.032 ou 01.48.01.48.88 Mail: info@pcsoft.fr



WINDEV : élu «Langage le plus productif du marché»
www.pcsoft.fr



Validez vos applications Linux avec AppChecker

Concevoir une application Linux qui fonctionne sur toutes les distributions du marché n'est pas trivial. Il y a beaucoup de raisons à cela. En premier lieu, la grande liberté qu'ont les développeurs de distribution pour concevoir leur système basé sur le noyau GNU/Linux. Il en résulte des différences de comportement qui font que les applications ne fonctionneront pas aussi bien sous une distribution que sous une autre. Cet article est fait pour apporter des éléments de réponse aux développeurs qui ciblent l'environnement Linux. Pour cela, nous allons principalement nous appuyer sur Appchecker, l'outil de référence de la Linux Foundation.

Le Linux Standard Base (LSB) est un projet réalisé en collaboration avec différentes distributions Linux sous l'égide de la Linux Foundation. Son objectif consiste en la standardisation de la structure interne des systèmes d'exploitation basés sur le noyau Linux. LSB est donc une extension de la spécification POSIX. LSB reprend également quelques autres standards tout en apportant ses propres règles. La finalité du LSB est de développer et promouvoir un ensemble de spécifications

ouvertes qui permettront à terme d'améliorer la compatibilité entre distributions Linux et ainsi de permettre aux logiciels de s'exécuter sur tout système en tant que binaire. Tout cela sans nécessiter de cycle de compilation spécifique. Au delà, le LSB aide à coordonner les efforts de recrutement dans le secteur du logiciel afin de favoriser le portage et le développement d'applications Linux.

Afin d'assurer un contrôle de qualité, le LSB offre une certification aux produits qui passent avec succès le cycle de validation. Gage de qualité, cette certification devrait rapidement être adoptée par les entreprises et les professionnels du secteur du logiciel libre. Bien que la percée soit lente, on compte déjà un certain nombre de distributions et d'applications certifiées. LSB spécifie, par exemple, les bibliothèques standard, un certain nombre de commandes et d'utilitaires qui étendent le standard POSIX. Sont également spécifiés, l'organisation du système de fichier, les niveaux d'exécution (/etc/init.d), le système d'impression incluant les spoolers comme CUPS ainsi que les outils tels que Foomatic et quelques autres extensions au système graphique communément utilisé sous Linux : X Window System.

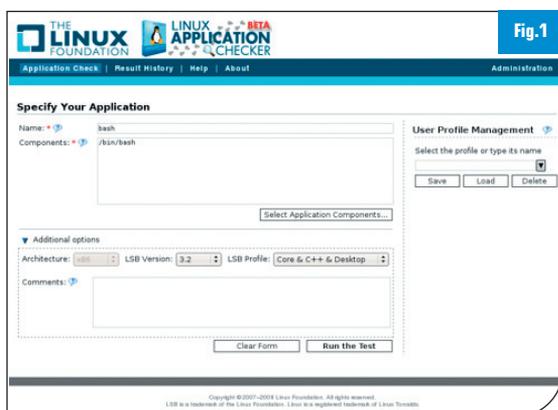
nant d'organismes externes à la fondation Linux tels que le projet Debian. Un exemple concret : le LSB spécifie que les packages logiciels doivent être sous la forme d'un paquet RPM. Or, Debian et son format DEB ne souhaite pas changer tout son système. L'idée de revoir tout le système de gestion logicielle des distributions semble irréaliste à ses développeurs. Cependant Debian inclut tout de même un support optionnel pour le LSB. Ainsi l'utilisateur pourra notamment utiliser l'application Alien qui lui permettra d'installer les paquets RPM. La suite de test a également été critiquée à plusieurs reprises pour être bogguée et incomplète. En 2005, Ulrich Drepper a critiqué LSB pour des tests mal écrits qui pouvaient entraîner des incompatibilités entre des composants pourtant certifiés par ces tests et des systèmes Linux également certifiés. Heureusement, la couverture des tests a été améliorée depuis et l'outil Linux Application Checker corrige les défauts majeurs imputés.

Appchecker : présentation de l'outil

Linux Application Checker, ou AppChecker, est un outil récent qui permet d'aider les développeurs à cibler leurs applications sur Linux. Il se base sur un framework de tests développé par l'Académie des Sciences Russes et utilise le travail de la Linux Founda-

Quelques critiques malgré tout...

On lui reproche de ne pas prendre en considération les remarques prove-





tion en reprenant les règles préconisées par le LSB. Pour l'anecdote, l'Académie des sciences de Russie est à l'origine d'un célèbre jeu vidéo, Tetris, conçu par l'un de ses salariés Alexey Pajitnov en 1985. AppChecker analyse les binaires d'une application et non le code source, ainsi il suffit de faire pointer l'outil vers l'application à analyser. Le système automatisé compare la table de symbole de l'application avec les bibliothèques connues parmi les 30 du LSB. Une fois la comparaison terminée, un rapport détaillé est généré et référence l'ensemble des résultats des tests. Ces informations donnent aux développeurs une forte indication de la capacité de son application à fonctionner sur chacune des distributions. Quelques cas connus de problèmes ne peuvent pas être détectés ou entraîneront des erreurs alors qu'il n'y en a pas. Par exemple, l'agencement d'une structure de données 64-bits sera différente de sa version 32-bits. AppChecker n'est pas en mesure de traiter les différences entre ces deux versions. Un autre problème : Appchecker ne peut pas prendre en compte les modifications mineures apportées à la nouvelle version d'un logiciel dans le comportement de ses fonctions.

Installation

Pour utiliser App Checker, vous devrez créer un compte sur le réseau de développeurs Linux, dont le site est en bêta (<http://ldn.linuxfoundation.org>). Il s'agit, entre autres, d'un réseau social qui vous permettra de maintenir des relations dans le milieu du développement Linux et de suivre des blogs, bookmarks et autres flux RSS. La communauté se veut également le point d'intersection entre les développeurs, les vendeurs et les distributeurs d'applications Linux. Le LDN (Linux Developer Network) est maintenu par son community manager, Brian Proffitt. Celui-ci est connu, entre autres, pour contribuer à l'écosystème du logiciel libre. Il est notamment éditeur du site web linuxtoday.com et a rédigé un livre sur le système Red Hat. Une fois votre compte créé, vous pourrez télécharger l'archive

des outils au format tar.gz. L'outil est disponible pour différentes architectures : x86, s390, PPC... Sur PC standard, on choisira la version x86. Ensuite, on va extraire l'archive. Vous pouvez utiliser la commande tar suivante ou votre bureau Linux si celui-ci le permet (clic droit puis extraire) :

```
> tar -xvzf Linux-app-checker-local-2.0-3.i486.tar.gz
```

Nous allons maintenant lancer l'application qui est en fait un serveur web embarqué. Par défaut le port utilisé est le 8889, vous pouvez cependant en choisir un autre en le précisant dans la ligne de commande :

```
> ./linux-app-checker/bin/app-checker-start.pl 8080
```

Cette commande va tenter de reconstruire votre navigateur préféré et de le lancer. Si celle-ci échoue, essayez manuellement de vous connecter à l'adresse <http://localhost:8080>. Ce serveur sera également accessible depuis une autre machine via son adresse ip ou son nom dans le cas où un serveur DNS est en place. [Fig.1]

Prise en main

L'interface web est particulièrement simple à prendre en main. Pour illustrer la procédure de validation d'une application que vous venez de développer, nous allons analyser le programme bash. Pour cela, cliquez sur "Application check" dans le menu. Vous pouvez à présent sélectionner les composants binaires à analyser. Nous allons ajouter le binaire de bash qui se trouve dans le dossier /bin en cliquant sur le bouton "Select application components...". Vous remarquerez que dans la liste des options, vous pourrez sélectionner la version de la spécification LSB à utiliser. Nous laisserons les paramètres par défaut puis nous lançons l'analyse. [Fig.2]

Dans notre cas, l'analyse ne durera que quelques secondes et nous aurons à peine le temps de prendre conscience des tests que va exécuter App Checker. L'outil est capable d'analyser des paquets RPM, des scripts Shells, Perl et Python mais également

Posix

POSIX ou "Portable Operating Interface" consiste en une collection de standards spécifiés par l'organisation IEEE. Celle-ci définit les API relatives au Shell et aux utilitaires pour les logiciels libres ou non compatibles avec les systèmes Unix. Au-delà des systèmes Unix, les développeurs de systèmes peuvent être amenés à respecter ces standards. C'est le

cas de Windows qui pour certains points est conforme avec les standards de l'IEEE. POSIX existe depuis 1988 mais ce n'est que quelques années après sa création qu'il a adopté le nom qu'on lui connaît aujourd'hui, suite à la proposition de Richard Stallman pour remplacer le nom un peu trop standard de IEEE-IX. Richard Stallman est le père de la fondation GNU et le développeur de nombreux outils open source.



Fig.3

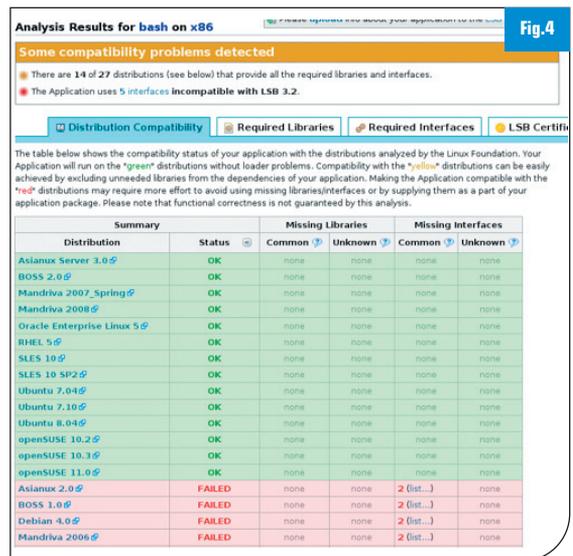


Fig.4



des bibliothèques ainsi que des exécutables ELF. Ici, nous nous trouvons dans le dernier cas. [Fig.3]

On appréciera le nombre de systèmes supportés par l'analyse et la précision du rapport. En effet, AppChecker ne se limite pas à indiquer le statut d'exécution des tests. Il apporte également les précisions nécessaires à la correction des problèmes. Ici, je suis sur une distribution Ubuntu 8.04. [Fig.4]

Si on regarde le détail des erreurs sur Debian 4.0, il apparaît clairement que le binaire bash de cette distribution requiert des interfaces qui ne sont pas présentes. Pour obtenir la certification, il faut passer avec succès l'ensemble des tests sur la totalité des distributions. Vous pourrez prendre connaissance des informations relatives à ce certificat sur l'onglet "LSB Certification". Vous pourrez ensuite envoyer les informations de certifications sur la plate-forme de la fondation Linux via l'outil LSB Navigator. Il est possible de comparer les résultats de vos tests successifs au sein de la page "Result history" qui reprend l'ensemble des tests que vous avez effectués localement par le passé.

Pour aller plus loin

Bien que simple et efficace, l'interface web ne conviendra pas nécessairement à toutes les utilisations. Si vous faites partie de ceux qui utilisent l'intégration continue, vous allez certainement préférer employer les scripts en ligne de commande afin de les intégrer dans des solutions comme Conti-

LSB Distribution Testkit

AppChecker est destiné aux développeurs d'application, cependant il existe son pendant pour les développeurs de distributions Linux : le LSB Distribution Testkit Manager (LSB DTK Manager). Il consiste également en une interface web qui permettra d'exécuter l'ensemble des tests au sein de cet outil unifié. Un outil en ligne de commande permet également d'accéder aux fonctionnalités de tests dans l'objectif d'apporter la possibilité d'écrire des scripts. Son utilisation facilitera l'exécution de tests en les rendant plus attractifs aux développeurs. L'objectif ultime de cet outil est de certifier les distributions. Xandros Server 2.0 est le premier produit à être certifié par la fondation Linux grâce au LSB Distribution TestKit. Les fonctionnalités clés du kit de test sont les suivantes : mise à jour automatique des suites de tests à partir du site de la Linux Foundation, intégration d'un système de certification, rapports de tests unifiés, l'exécution des tests en un clic.

num. Cela vous permettra de valider vos binaires après chaque compilation sur vos serveurs d'intégration. De la même manière si vous souhaitez créer des cibles de validation dans vos scripts de compilation (Makefile, Ant, Maven...), les outils suivants vous seront utiles :

- lsb-appchk
- lsb-appchk.pl
- lsb-appchk.py
- lsb-pkgchk

Ils permettent de tester des programmes (Perl, Python, binaire, paquets...). Le résultat sera directement affiché dans le Shell. De la même manière qu'avec l'interface web, vous pourrez analyser plusieurs fichiers et spécifier la version du standard LSB à utiliser. Actuellement nous en sommes à la révision 3.2, la version 4.0 est pour le moment en bêta.

Conclusion

Parmi les systèmes certifiés, on peut citer Oracle Enterprise Linux, openSUSE, Red Hat Enterprise Linux ou encore Ubuntu. Comme nous avons pu le voir, AppChecker se destine aussi bien aux industriels qu'aux développeurs "amateurs". La dernière version comprend 1450 tests et le standard ne cesse d'évoluer et de s'améliorer. Bien que AppChecker soit une référence dans le domaine, OLVER Core test suite réalise le même travail que celui-ci. OLVER effectue des tests fonctionnels et de conformité de la majorité des API systèmes de Linux. Les tests sont générés automatiquement à partir des scénarios de tests et des spécifications formelles. Il y a donc un véritable environnement de tests et de validations disponible sous Linux. Il est important pour les développeurs de s'y intéresser afin d'assurer la meilleure interopérabilité possible.

■ Loïc Guillois

Continuum : solution d'intégration continue

Apache Continuum est un serveur d'intégration continue destiné aux entreprises ayant en charge de nombreux projets. Cet outil permet la mise en place de compilations journalières, de distributions, gestion des rôles et s'intègre avec les outils de compilation les plus populaires : Maven, Ant... ainsi que les gestionnaires de code source les plus répandus. Continuum permet d'industrialiser les processus liés à la compilation

et à l'intégration d'application en phase de développement. Son utilisation couplée à AppChecker pourra permettre de s'assurer quotidiennement qu'une application Linux respecte la LSB et pourra s'exécuter sans encombre sur les distributions Linux cibles. Il existe également d'autres solutions plus simples et plus légères. On peut notamment citer LintBuild qui est également développé en Java.

Distribution: Debian 4.0
Present Libraries:
<ul style="list-style-type: none"> • lib.so.6 • libdl.so.2 • libcurses.so.5
Missing Interfaces:
<ul style="list-style-type: none"> • <code>_stack_chk_fail</code> GLIBC_2.4 (lib.so.6) • <code>access</code> GLIBC_2.4 (lib.so.6)





Pourquoi et comment utiliser Linux Standard Base ?

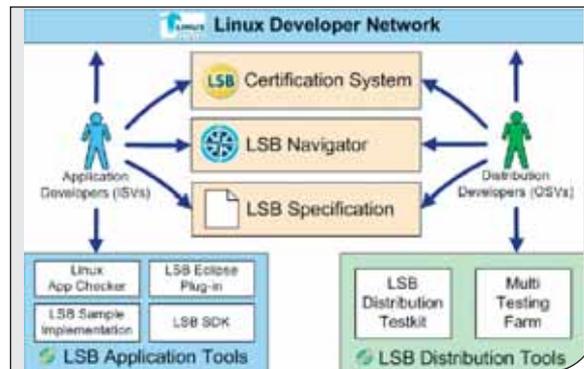
Il existe plus de 300 distributions Linux (<http://distrowatch.com/>). Si vous voulez distribuer votre logiciel pour qu'il fonctionne sur toutes ces distributions Linux, vous devez résoudre tout un ensemble d'incompatibilités entre elles.

En effet, pour une plate-forme matérielle donnée, toutes les distributions Linux utilisent dans la plupart des cas les mêmes briques: le noyau Linux, la glibc, le serveur X11 venant de Xorg... mais elles ne font pas toujours les mêmes choix d'intégration. Par exemple, elles n'utilisent pas les mêmes versions, ne compilent pas les logiciels avec les mêmes options, ne placent pas les fichiers aux mêmes endroits, n'installent pas les mêmes logiciels par défaut... Si l'on veut creuser un peu plus, en supposant que vous n'utilisiez que des composants distribués par toutes les distributions, vous allez vous heurter aux formats de diffusion des logiciels et aux conventions de nommage.

Les packages de déploiement

Dans le monde Linux, il existe principalement 2 formats de distribution : les paquetages au format RPM utilisés par les distributions issues de Red Hat, SuSE, Mandriva et les paquetages au format DEB utilisés par les distributions issues de Debian et Ubuntu. Il existe des outils de conversion comme Alien, donc vous pourriez vous dire que tout est parfait. Mais en fait, le problème ne vient pas du format en lui-même, car les formats RPM et DEB sont assez proches, mais de la façon de les utiliser. Là encore chaque distribution a fait son choix de nommage des paquetages et si vous voulez dépendre d'un paquetage pour être sûr d'avoir le bon support dans le système, et bien ce sera impossible. Par exemple, le paquetage du langage Python va s'appeler python dans une distribution et python2.5 dans une autre. Un autre problème provient du fait que les distributions n'installent pas toujours les fichiers aux mêmes endroits, donc si vous voulez faire référence à ces fichiers cela va être compliqué. Il y a aussi des problèmes identiques pour les scripts d'initialisation de services, les descriptions de menus, les soname des bibliothèques dynamiques...

Une solution utilisée par la plupart des projets est de diffu-



ser leurs logiciels sous forme de source. De cette façon, plus de problème de compatibilité binaire. Cela suppose que les utilisateurs savent compiler et résoudre les problèmes de dépendance. Une autre façon de faire, qui est souvent utilisée par les gros logiciels sous Linux ou sous les différents UNIX, est de compiler en statique ou de distribuer tous les composants dont vous avez besoin, et vous en assurez la maintenance et la gestion des alertes de sécurité vous-même... Si cela n'est pas le cas ou si vous ne pouvez pas distribuer les sources, la LSB a été créée pour résoudre ces problèmes.

Qu'est-ce que la LSB ?

LSB est un ensemble de standards créés, sous le " contrôle " de la Linux Foundation (<http://www.linuxfoundation.org/>) par différents acteurs du monde Linux: distributeurs Linux et éditeurs de logiciels. La LSB est basée sur des standards existants comme POSIX et SUS. Quand des standards n'existaient pas, la LSB a standardisé les usages de fait. La LSB est une façon spéciale de fabriquer les binaires et de construire des paquetages. Pour arriver à cela, les spécifications définissent :

- comment créer des paquetages et des logiciels pour qu'ils puissent s'installer et fonctionner partout.
- comment éviter les collisions de noms avec les autres diffuseurs de logiciels.
- quelles sont les commandes et les bibliothèques dynamiques présentes et utilisables par tous les logiciels.
- quelles sont les interfaces binaires utilisables (ABI).

Comment utiliser la LSB ?

Pour créer des binaires et paquetages conformes à la LSB, il faut :

- Générer des binaires ELF.
- Utiliser le chargeur LSB qui charge les bibliothèques LSB. Par exemple sur x86, il faut utiliser `/lib/ld-lsb.so.3`.
- Utiliser uniquement des bibliothèques et des commandes faisant partie de la LSB.



- Se conformer au standard FHS pour le placement des fichiers dans l'arborescence.
- Utiliser le sous-ensemble LSB des possibilités de RPM. Pour éviter les conflits de nommage, la LSB spécifie d'utiliser un préfixe unique. Pour obtenir un préfixe unique, vous pouvez enregistrer votre préfixe auprès de l'organisme LANANA (Linux Assigned Names And Numbers Authority) à l'adresse suivante: <http://www.lanana.org/> ou alors utiliser votre nom de domaine Internet.

Ce préfixe doit être utilisé pour les fichiers et répertoires:

- /opt/<prefix>/ pour les programmes, données statiques et bibliothèques.
- /var/opt/<prefix>/ pour les données dynamiques.
- /etc/opt/<prefix>/ pour les fichiers de configuration.
- /etc/init.d/lsb-<prefix>-* pour les scripts d'initialisation.
- /etc/{crond.d,cron.daily,cron.weekly,cron.monthly}/lsb-<prefix> pour les crontabs.

Ce préfixe doit aussi être utilisé dans les noms de vos paquetages de cette manière: lsb-<prefix>-*.

Générer des binaires LSB

Vous pouvez utiliser les commandes lsbcc et lsbcc++ pour générer des binaires compatibles LSB. Ces commandes sont en fait des wrappers autour de cc et c++ qui y ajoutent les bons paramètres pour avoir les chemins des bibliothèques LSB et du chargeur dynamique LSB automatiquement.

Exemple sur une machine x86_64:

```
$ lsbcc -o prog prog.c
$ ldd prog
libm.so.6 => /lib64/tls/libm.so.6 (0x00002aaaaabc3000)
libc.so.6 => /lib64/tls/libc.so.6 (0x00002aaaaad19000)
/lib64/ld-lsb-x86-64.so.3 (0x00002aaaaaab000)
```

Si vous avez besoin d'utiliser des bibliothèques non disponibles dans la LSB, vous avez les choix suivants :

- Édition de lien statique
- Distribuer les bibliothèques avec vos packages.
- Utiliser des packages LSB qui fournissent cette bibliothèque.

Packaging

Pour fabriquer des packages RPM compatibles LSB, il faut désactiver auto-requires et auto-provides pour éviter de tirer des dépendances du système et n'avoir que les dépendances sur les paquetages LSB. Pour cela il faut mettre cette directive dans votre fichier .spec:

```
AutoReqProv: no
```

Il faut ensuite ne dépendre que du package lsb et explicitement des packages dont vous avez besoin:

```
Requires: lsb >= 3.2, lsb-<whatever>
```

En option, vous pouvez redéfinir les macros suivantes pour utiliser les macros habituelles de rpm comme %configure et %makeinstall:

```
%define _prefix /opt/<prefix>
%define _mandir %_prefix/man
%define _docdir %_prefix/doc
%define _datadir %_prefix/share
%define _localstatedir /var/opt/<prefix>
%define _sysconfdir /etc/opt/<prefix>
```

Scripts d'initialisation

La LSB définit des utilitaires pour ajouter ou enlever des services lancés au démarrage. Utiliser les commandes suivantes dans vos scriptlets rpm pour ajouter/supprimer vos services:

```
if [ $1 = 1 ]; then
    /usr/lib/lsb/install_initd /etc/init.d/lsb-<
prefix>--<name>
fi

if [ $1 = 0 ]; then
    /usr/lib/lsb/remove_initd /etc/init.d/lsb-<
prefix>--<name>
fi
```

Il faut utiliser les en-têtes suivants dans vos scripts pour que les commandes install_initd et remove_initd fonctionnent:

```
### BEGIN INIT INFO
# Provides: lsb-ourdb
# Default-Start: 2 3 4 5
# Short-Description: start and stop OurDB
# Description: OurDB is a very fast and reliable
database
# engine used for illustrating init scripts
### END INIT INFO
```

Conclusion

La LSB permet d'éviter une nouvelle guerre des UNIX ! Elle permet de fabriquer une fois et d'installer partout ! Dans ce numéro vous trouverez en complément un article sur AppChecker.



■ Frédéric Lepied
frederic.lepied@splitted-desktop.com

Ancien de la direction technique de Mandrakesoft (2002-2006), il intègre Intel, puis en 2008 Splitted-Desktop Systems en qualité de directeur de l'innovation. **Splitted-Desktop Systems** est spécialisé dans le développement de solutions matérielles et logicielles destinés aux accès et services Internet.





Mon code est-il légal ?

Peut-on intégrer sans souci un code source extérieur ? La réponse est **NON** surtout si la traçabilité de ce code n'est pas certaine ou si la licence sous lequel il est, peut être incompatible avec la licence envisagée du logiciel en développement. La prudence est de mise.

L'open source propose des dizaines de milliers de projets, plus ou moins actifs et autant de codes source que l'on peut utiliser, à condition de faire attention à leur provenance et à la ou les licences utilisées. Les défis sont les suivants : trouver le bon code, gérer les licences et veiller à la sécurité du code. Privilégiez les projets connus. De plus en plus d'applications sont développées avec des briques de diverses origines. On ne compte plus les projets de développement hybride mêlant du code externe, interne, ouvert et propriétaire. Ces projets posent des problèmes au quotidien : car il faut intégrer le code extérieur, s'assurer de son niveau de sécurité et de sa qualité.

Il faut trouver le bon code (bonne version, qualité et maturité de celui-ci, visibilité des licences utilisées, voire des redevances à verser) puis évaluer les risques potentiels que l'on encourt. Cela implique de connaître parfaitement son code source, même quand on fait du copier / coller et surtout de déterminer précisément l'origine de tout code extérieur même si un développeur de l'équipe en utilise mais ne le dit pas ! Le moindre doute doit alerter le développeur, le chef de projet, les responsables puis les responsables du service juridique s'il y en a un.

Deux pistes pour veiller à la conformité de son code

Aujourd'hui, il existe des entités juridiques spécialisées dans la traçabilité du code et dans la gestion des licences. Soit en cabinet indépendant soit dans des SSII, ainsi Linagora possède un pôle juridique reconnu qui peut intervenir pour des clients demandeurs. L'autre solution, les



deux peuvent d'ailleurs fonctionner de concert, consiste à utiliser un outil de vérification automatique du code qui :

- s'appuie sur une large base de données la plus exhaustive possible
- scanne automatiquement du code pour identifier l'origine des codes, la licence et les codes potentiellement litigieux

Un des outils les plus connus est Black Duck. Un autre éditeur propose aussi un outillage assez similaire : Palamida. Ces outils s'intègrent dans le processus de développement. Il est ainsi possible de vérifier que seul le code dûment autorisé est présent

dans le code final. Cela évite bien des problèmes par la suite. Black Duck définit ainsi sa mission : trouver le code nécessaire, sélectionner le code selon ses critères, approuver le code, produire le projet avec le code sélectionné, valider le code global, monitorer l'utilisation du code et de sa diffusion dans les applications. Pour assurer une telle mission, l'éditeur repose sur une base de données particulièrement importante : 170 000 projets open source référencés, 400 millions de fichiers, plusieurs milliards de lignes de code, prise en compte de 1 400 licences, 28 000 failles de sécurité référencées ! Surtout, la base subit une mise à jour constante. Toutes ces solutions coûtent cher, mais pour une grande entreprise, c'est indispensable. Pour un développeur indépendant, une SSII ou une PME, un petit éditeur, difficile de se payer une telle prestation mais si un client demande un rapport sur la propriété intellectuelle du code, vous devrez le fournir.

■ François Tonic

Nuxeo surveille et audite son code

// Nous n'utilisons pas de code de "provenance plus ou moins inconnue". Le choix des librairies est toujours réfléchi, après un audit technique et juridique des différentes options qui se présentent à nous, y compris pour écrire nos propres librairies. Nous avons la contrainte que notre projet doit être sous une licence libre "non-contaminante". Nous excluons donc par exemple l'usage de composants sous licence GPL et nous privilégions les licences Apache et LGPL. Dans le doute sur la compatibilité entre deux licences, nous étudions la question en détail, en allant jusqu'à contacter les auteurs pour avoir leur avis et si le doute persiste, nous nous abstenons de l'utiliser. La décision d'utiliser telle ou telle bibliothèque n'est pas laissée au choix des seuls développeurs, mais implique la direction technique et les architectes seniors du projet. De plus, l'ensemble des développeurs est sensibilisé. Les revues de code continues sont l'occasion de surveiller l'introduction de nouvelles dépendances dans la base de code. Enfin, nous faisons avant chaque version un audit de toutes les librairies qui sont utilisées pour vérifier qu'aucune des règles n'a été enfreinte par inadvertance. Nous avons développé un outil, basé sur Maven, qui facilite ce travail // , nous a confié Stéphane Fermigier (Nuxeo).

Métier : développeur Open Source

Libres, passionnés, innovants, autonomes. Telles sont les qualités le plus souvent citées pour les développeurs open source. Des parcours, des formations, mais aussi des modes de recrutement originaux caractérisent les spécialistes du Libre.

En ces temps de crise, qui touche même les SSLL, l'Open Source semble bien tirer son épingle du jeu. Pour des raisons économiques, les entreprises préfèrent recourir à des technologies qui n'exigent pas un investissement financier en licences logicielles. Mais l'impératif économique n'est pas le seul facteur en jeu. " *Ce n'est pas tant le facteur coût que la liberté, l'autonomie, la souplesse, que recherchent les entreprises* ", constate



Marine Soroko, directrice associée de Core-Techs, éditeur qui se définit comme " Open Source Factory ". Aussi les éditeurs, SSLL et autres sociétés spécialisées en logiciel libre ont-ils pour la plupart des plans d'embauche assez importants. Alter Way prévoit 50 recrutements pour 2009 et une croissance organique de +25%. Après avoir rallié une centaine de nouveaux collaborateurs en 2008, Linagora annonce un plan de recrutement quasi équivalent pour l'année 2009. Ce sont ainsi des centaines de nouveaux postes qui vont se créer dans ce secteur. Les missions sont très variées. Depuis la R&D chez les éditeurs, encore très nombreux, jusqu'à l'intégration chez les SSLL. Même les grands acteurs de l'informatique (IBM, Oracle, Microsoft) y prêtent une oreille attentive.

Un développement vers l'applicatif

L'essor de l'Open Source est aussi lié à l'évolution du marché. " *Les solutions technologiques arrivent à maturité, d'où il résulte que le Libre prend de plus en plus de poids* ", souligne **Philippe Montargès**, PDG d'Alter Way,



Philippe Montargès

intégrateur open source. Et, comme le rappelle Cédric Carbone, directeur technique de Talend, fournisseur de solutions d'intégration de données open source : " L'Open Source a fortement évolué, et donc les profils aussi. Après avoir couvert les champs fonctionnels de l'infrastructure, des réseaux, des systèmes d'exploitation, il s'est étendu aux bases de données et middleware, et investit maintenant les outils applicatifs et les couches métiers (PGI, GRC, etc.). " Ce que confirme **Jean-Noël de Galzain**, PDG de Wallix, spécialiste de la sécurité informatique en open source : " *Le marché a mûri sur les applications web, GED, gestion du contenu, bases de données, PGI, GRC, messageries... Il y a une forte demande autour des couches métiers.* "



Dès lors, aux profils très techniques des premiers candidats, succèdent des compétences plus orientées métiers. " *Nous recherchons des informaticiens open source qui ont acquis une expérience métier pour parler avec des maîtrises d'ouvrage* ", précise Cédric Carbone.

Par ailleurs, alors que les méthodologies dites classiques étaient prodiguées par de grands éditeurs, " *les nouvelles méthodologies, plus liées à des processus (agiles, scrum et*

autres méthodes collaboratives...), donnent une longueur d'avance aux développeurs open source ", indique Jean-Noël de Galzain.

L'ADN du développeur open source

Il n'existe pas de cursus standard pour le développeur open source. C'est un passionné, " autodidacte " du point de vue des compétences techniques open source, qui a développé par lui-même une forte expertise.

L' " ADN du développeur open source " est ainsi défini par Cédric Carbone : " *Très passionné, très motivé, autonome et communiquant en anglais* ". La communication avec la communauté open source sur le web est, en effet, un des points caractérisant ce type de développeur : " *En plus d'un très bon niveau technique, il n'est jamais bloqué car il sait mettre à profit des communautés, utiliser des réseaux sociaux* ", ajoute Cédric Carbone. Car le marché de l'Open Source est mondial.



" *Nous avons des clients partout dans le monde, ce qui nécessite une approche mondialisée* ", confirme

Benjamin Mestrallet,

PDG et fondateur de xExo Platform, éditeur de logiciel pour la collaboration en entreprise.

Dans l'Open Source, le travail est souvent très décentralisé. Aussi de nombreux développeurs télétravaillent, hors du réseau d'entreprise. " *Chez nous, beaucoup de personnes télétravaillent ; ils utilisent des outils de collaboration open source ; ils sont indépendants très rapidement* ", résume Benjamin Mestrallet. Les sociétés spécialisées dans le Libre utilisent souvent cette possibilité, en

Les outils de la Direction Informatique

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



L'INFORMATION SUR MESURE

Actu triée par secteur

Cas clients

Avis d'Experts

*Etudes
&
Statistiques*

Infos des SSII

Vidéos

Actus

Evénements

Newsletter

L'INFORMATION EN CONTINU

www.solutions-logiciels.com





mettant en place des messageries instantanées pour faciliter la collaboration entre développeurs. De ce fait, les échanges se font généralement en anglais, quelle que soit l'origine du développeur, car, il faut le souligner, les sociétés open source n'hésitent pas à faire appel à des développeurs du monde entier.

L'aspect créatif et innovateur est aussi mis en avant par la plupart des acteurs. Dans le domaine du Libre, tout change tout le temps, l'innovation et la veille technologique sont donc incontournables. " Dans le cadre de projets collaboratifs et pour mettre en place de nouveaux produits, nos ingénieurs R&D peuvent rapidement concrétiser une idée, sans contraintes juridiques (pas de licences, pas de droit d'utilisation...) ", indique Jean-Noël de Galzain. " Les développeurs open source sont motivés par la veille technologique permanente ", remarque **David Duquenne**, directeur de division Services & Technologies chez Improve, centre de compétences Objet, Internet et nouvelles technologies. " Ils maîtrisent bien la prospection technologique, consultent régulièrement les forums de discussion, connaissent les projets open source en cours, ils vont cher-

cher l'information directement à la source. " Ce qui compte surtout, pour Philippe Montargès, " c'est l'implication, l'adhésion aux valeurs du logiciel libre, et la créativité. "

Des facilités de recrutement

Les compétences recherchées par les spécialistes de l'Open Source sont multiples : ingénieurs R&D, chefs de projet, consultants, techniciens d'exploitation, intégrateurs, formateurs... Même si les rémunérations peuvent être plus faibles que pour les autres domaines, les acteurs de l'Open Source ne semblent pas avoir de difficultés à recruter, et –pour ce qui concerne les SSL – le turn-over est relativement faible par rapport aux SSII classiques. " Le Libre est un attrait pour ceux qui cherchent à ajouter cette technologie à leur expérience, par exemple dans le cadre d'applications internet/intranet, de projets GED, de sécurité... ", ajoute Jean-Noël de Galzain. De plus, ils peuvent avoir l'occasion de participer à des projets de R&D très ambitieux, comme celui auquel participe Wallix dans le cadre du pôle de compétitivité Systematic. Les candidats sont souvent connus de la société avant de

faire l'objet de recrutement. La communauté autour d'un produit ou d'un éditeur peut comprendre plusieurs milliers de personnes. Ainsi Talend fait état de 2 000 à 3 000 bêta-testeurs, sans compter les développeurs de patches et autres contributeurs, qui connaissent déjà les solutions de cet éditeur et travaillent couramment avec elles. " Nous faisons des offres d'emploi sur les forums de notre communauté ", indique Cédric Carbone. " Les candidats connaissent déjà notre technologie, notre société, notre mode de fonctionnement. " Chez Improve, si l'on recrute des profils expérimentés sur les technologies récentes, les juniors font souvent leur entrée via des stages, où ils ont travaillé en interne sur un projet de R&D en open source. Si Alter Way recrute, comme tout le monde, des profils sortant d'écoles d'ingénieurs, la société peut aussi intégrer des candidats moins diplômés, voire des autodidactes : " Nous les faisons monter en puissance grâce à nos formations internes ", précise Philippe Montargès. " Nous sommes dans une logique d'investissement, en évitant de basculer dans la logique traditionnelle de candidats prêts à l'emploi. "

■ Claire Rémy



Les 25 erreurs de programmation les plus fréquentes

EXCLUSIF



Le rapport **complet**
en français
est disponible
sur
www.programmez.com



Au cœur du Mozilla Labs



Que représente Mozilla Labs ? Avant tout une pépinière d'idées, les projets élaborés aboutiront ou non à une fonction, un logiciel. Il s'agit de tester des concepts, des orientations logicielles et de regarder la réaction de la communauté, des éditeurs, des développeurs. C'est là que se concoctent quelques idées du futur de Mozilla.



Aujourd'hui, le Mozilla Labs, c'est : 6 développeurs, 2 designers, 9 intégrateurs, une cinquantaine de contributeurs réguliers de la communauté et plus d'une centaine de personnes dans la partie " concept series ". Le cerveau de l'ensemble est basé à Mountain View et à Toronto, même si plusieurs bureaux locaux sont éparpillés dans le monde. Actuellement, une douzaine de projets sont actifs et évoluent au gré des sessions techniques, des idées, ou sont laissés tels quels... **Aza Raskin**, leader de l'expérience utilisateur du Labs, nous parle un peu plus de cette boîte à idée de l'Open Source.

Programmez ! : comment se déroule une journée type dans les Labs ?

Aza Raskin : il n'existe pas de journée

type au Labs. Chaque jour est différent, avec une inspiration différente, un nouveau focus. Pourvu que cela rende le web (plus) ouvert et contribue à améliorer le lieu où on travaille, joue, communique.

P. ! : De quelle manière une idée germe, se construit et comment la fait-on aboutir ?

A.R. : Le Labs est un espace virtuel où ensemble, on crée quelque chose, on expérimente, on joue avec les dernières innovations du web, et de la technologie. Des milliers de personnes participent à notre processus Concept Series [<http://labs.mozilla.com/projects/concept-series/>]. Au Mozilla Labs, nous regardons les gens utiliser le web et nous nous posons la question : comment pouvons-nous les aider à moderniser, à améliorer l'ex-

périence ? Trop souvent, nous sommes tous, en tant qu'utilisateurs, tombés dans le piège du " syndrome de Stockholm " et apprenons à aimer le système qui nous piège. En observant la façon dont les utilisateurs mettent en œuvre le web au quotidien, nous pouvons créer des innovations qui peuvent libérer l'expérience, l'usage. Par exemple, Weave nous libère des contraintes de l'utilisation de multiples ordinateurs et terminaux. Avec Weave, les mots de passe, onglets, historiques, marque-pages sont automatiquement repris ainsi que l'expérience de l'utilisateur. Ils le suivent quels que soient les terminaux utilisés ! On se sent toujours comme à la maison. Ubiquity nous libère de la nécessité de jongler entre les sites web et les tâches sont à jour. Avec Google,

Une photo de famille des équipes, des contributeurs...



vous tapez ce que vous voulez trouver. Avec ce projet, vous tapez ce que vous voulez faire. Le navigateur le fait, où que vous soyez !

P. ! : lorsqu'une idée est validée, comment le projet démarre-t-il, qui le dirige et de quelle manière peut-il aboutir ?

A. R. : De nombreux projets se réalisent très rapidement. *Coop* ou encore *New Tab* furent créés en moins d'une semaine ! D'autres, comme *Ubiquity* et *Weave*, possèdent

un impact plus important et se conçoivent sur la durée. La gestion projet est la même que pour le reste de Mozilla et de l'Open Source en général. Chaque projet a un " champion ", nous avons des téléconférences chaque semaine, des wikis, des blogs, des chats ! De plus, nos conversations sont ouvertes et tout le monde peut y participer.

P. ! : qui décide de lancer le développement d'un projet ? le gère au quotidien ? Qui trouve les développeurs ?

A. R. : Il s'agit de l'inspiration, du feeling. Si une idée n'inspire pas les participants ou ne convainc pas un " champion " d'en prendre les rênes, alors nous ne le faisons pas ! Les personnes qui travaillent à Mozilla sont en quelque sorte l'échafaudage de la communauté qui crée, développe, soutient Firefox. Mozilla Labs est également un échafaudage. Le leader d'un projet peut sortir de nos rangs ou venir de l'extérieur. Et il peut vivre n'importe où dans le monde. Le feedback est critique pour les projets du Labs. Un projet ne vit pas, n'évolue pas dans le vide ! Les idées se transforment au fil du temps !

P. ! : n'importe qui peut-il aider le Mozilla Labs ?

A. R. : Oui, tout le monde peut participer au Labs. Il rejoindra le Concept Series. Nous sommes particulièrement intéressés par les designers ainsi bien sûr, que par les développeurs. Mais pour garder un contact au plus près des utilisateurs, de la réalité, régulièrement (en principe chaque mois), Mozilla Labs organise les Labs Nights. Occasion pour discuter des nouvelles évolutions, des fonctions, des tendances, recueillir les sentiments...

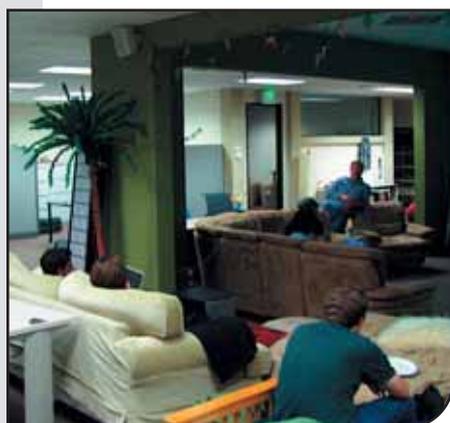
■ François Tonic



De gauche à droite : Aza Raskin, Jonathan DiCarlo, Dan Mills, Rhian Gracey, Myk Melez



Un coin de bureau des Labs



Ambiance détente – travail durant les soirées Labs...

Les projets du Labs

Bespin : un éditeur de code ligne

Le développement sera-t-il en ligne dans quelques années ? C'est en tout cas une des tendances actuelles chez les éditeurs. Mozilla lance le projet Bespin qui est avant tout un framework open source pour créer un éditeur de code. Dans la version actuelle, il supporte la coloration, les fonctions basiques d'édition, import / export de projets, prévisualisation de fichiers, etc.

Weave : une expérience en dehors des terminaux

Comment garder son expérience utilisateur, ses marque pages, quel que soit le navigateur et les terminaux. c'est un véritable défi technique. Weave constitue un projet passionnant et stratégique pour le futur de l'internet et de firefox. Comment se rendre totalement indépendant...

Personas : personnalisé l'interface

Comment simplement et rapidement personnaliser l'interface de Firefox ? Pour Labs, la réponse est Personas. Car le modèle actuel est vu comme trop lourd, trop compliqué. Il faut faire plus simple, plus intuitif...

Prism : le site web sur son bureau

Ce projet avait défrayé la chronique lors de sa présentation. Prism permet depuis son bureau d'accéder à son application web depuis le client Prism installé sur son bureau. Le site web est transformé depuis Firefox que l'on récupère ensuite dans Prism.

Test Pilot : l'autre manière de faire des tests

L'un des soucis de Labs et des logiciels en général consiste à récupérer et traiter efficacement les commentaires, les bugs, bref l'ensemble du feedback. Pour ce faire, Labs a conçu Test Pilot qui se présente sous la forme d'une extension Firefox.

Concept Series : penser l'interface de demain

Labs travaille activement sur l'interface de demain du web et des logiciels web. Dans cette section, on retrouve de véritables "concept cars" qui aboutiront ou non. On y trouve le concept de l'interface mobile. Et tout le monde peut proposer son propre concept. Pourquoi pas vous ?

Labs recèle encore bien d'autres trésors. N'hésitez pas à visiter régulièrement le site officiel pour les découvrir, les tester.

Site : <http://labs.mozilla.com>



DÉVELOPPEZ VOTRE SAVOIR-FAIRE



Langage et code, développement web, carrières et métier :
Programmez !, c'est votre outil de veille technologique.

Pour votre développement personnel et professionnel, abonnez-vous à Programmez !

Choisissez votre formule

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

11 numéros par an : 49 €*

Economisez 16,45 €*

*Tarif France métropolitaine

+ Abonnement INTÉGRAL

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

Cette option est réservée aux abonnés pour 1 an au magazine, quel que soit le type d'abonnement (Standard, Numérique, Etudiant). Le prix de leur abonnement normal est majoré de 10 € (prix identique

pour toutes zones géographiques). Pendant la durée de leur abonnement, ils ont ainsi accès, en supplément, à tous les anciens numéros et articles/dossiers parus.

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

PROGRAMMEZ

- Abonnement 1 an au magazine : 49 €** (au lieu de 65,45 € tarif au numéro) *Tarif France métropolitaine*
- Abonnement Intégral : 1 an au magazine + archives sur Internet et PDF : 59 €** *Tarif France métropolitaine*
- Abonnement Etudiant : 1 an au magazine : 39 €** (joindre copie carte étudiant) *Offre France métropolitaine*

M. Mme Mlle Entreprise : Fonction :

Nom : Prénom :

Adresse :

Code postal : Ville :

Tél : E-mail :

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

A remplir et retourner sous enveloppe affranchie à :
Programmez ! - Service Abonnements - 22 rue René Boulanger - 75472 Paris Cedex 10.

abonnements.programmez@groupe-gli.com

PRO Le magazine du développement **grammez!**

Offre limitée,
valable jusqu'au
30 avril 2009

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

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

Générer vos applications à partir des modèles UML2 avec BLU AGE 2009

Une version d'évaluation de BLU AGE BUILD édition 2009 est disponible sur le CD de ce numéro. En guise de prise en main, vous trouverez un modèle PIM exemple dans le dossier "évaluation guide" du CD. A partir d'un modèle, il est possible de générer l'intégralité d'une application bien au-delà du CRUD. BLU AGE BUILD permet de générer en temps réel les applications, de les exécuter pour les tester et de les déboguer dans le modèle.

Le modèle UML de cette application est composé :

- de 2 classes stéréotypées "entity" : "Contact" et "Salutation" placées dans un package stéréotypé "PK_ENTITY". Chaque attribut est typé et peut être spécialisé à l'aide d'un ou plusieurs stéréotypes. Ces classes représenteront le modèle des entités persistantes. Elles peuvent être liées entre elles par des relations. Dans notre exemple, nous avons une classe "Contact" qui représente l'entité du contact et une classe "Salutation" qui définit la civilité du contact [Fig.1].
- de 2 classes Business Object (BO) dérivées des entités définies ci-dessus et placées dans un package stéréotypé "PK_BUSINESS_OBJECT". Ces classes sont celles qui sont manipulées par l'application. Ceci permet d'enrichir le modèle métier avec des éléments non persistés [Fig.1].
- d'un service "ServiceContactCreate" déclaré à l'aide d'une interface (au

sens UML) et inclus dans un package stéréotypé "PK_SERVICE". Ce package peut contenir autant de services que nécessaire. Ces services définissent les traitements métiers d'une application. Chaque service manipule des "Business Objects" (BO).

Les services

Ces services sont de 4 familles :

- 1 - Services standards : opérations de traitements primitifs (CRUD, recherche, etc.). Ils sont stéréotypés : create, delete, update, find by id, etc.,
 - 2 - Services complexes : séquences de traitements issus d'autres services. Les services complexes peuvent inclure des conditions et des itérations. Ils sont stéréotypés "complex" puis décrits dans des diagrammes de séquences,
 - 3 - Services spécifiques : réutilisation de bibliothèques déjà disponibles (capital existant dans un SI). Ils sont stéréotypés "specific",
 - 4 - Services Web : utilisations de web services. Ils sont stéréotypés "ClientWebService".
- d'un acteur au sens UML. Cet acteur correspond à un rôle disponible pour les utilisateurs de l'application. Dans notre cas, nous avons un rôle nommé "simpleUser".

de uses cases inclus dans des packages UML permettant de définir l'organisation fonctionnelle de l'application. Les rôles définis ci-dessus sont associés à ces packages par des relations de dépendance qui permettent de déterminer les droits d'accès.

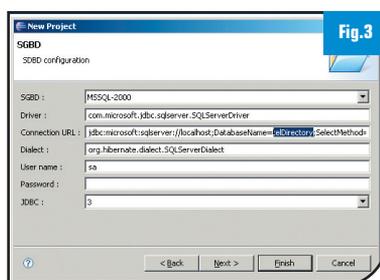
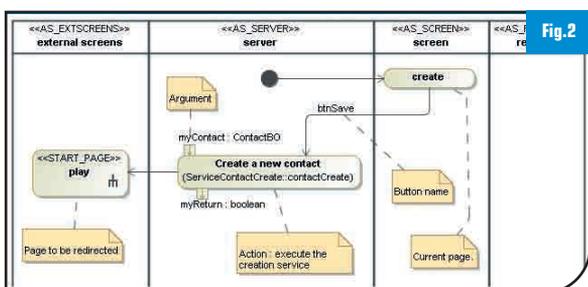
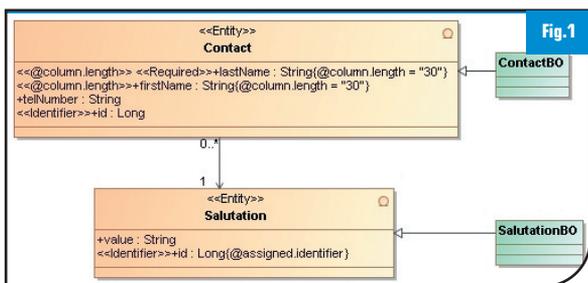
Chaque uses cases contient un ou plusieurs diagrammes d'activité correspondant à une page de l'application. Ces diagrammes décrivent les traitements réalisés suite aux actions exécutées sur la page, par exemple "clic sur sauvegarder". Chaque diagramme d'activité est décomposé en 4 "swimlines" permettant de classifier les traitements.

Le diagramme d'activité présenté ici [Fig.2] décrit l'affichage de la page "create" puis l'exécution du traitement d'insertion en base suite au clic du bouton "btnSave", c'est-à-dire l'exécution du service "ServiceContactCreate" (défini plus haut) avec comme paramètre l'objet "myContact" instan-

Figure 1 : Entités et business objects

Figure 2 : Diagramme d'activité de la page de création

Figure 3 : Paramétrage d'un projet



BLU AGE
AGILE MODEL TRANSFORMATION
2009 EDITION - BUILD

A propos de BLU AGE Edition 2009

Pour outiller vos projets agiles, BLU AGE dans son édition 2009 propose 3 modules : BUILD (sur le CD en version test), permettant d'élaborer de manière assistée des modèles PIM exécutables et de les tester en les transformant en applications JAVA ; DELIVER, permettant de générer en temps-réel vos applications vers vos Frameworks techniques; et REVERSE MODELING, vous offrant la possibilité d'extraire les spécifications d'applications existantes sous forme de modèles UML2.

ce de "ContactBO". Une fois le service exécuté, la page "play" est servie au client. Cette page "play" possède aussi son propre diagramme d'activité. Dans notre exemple le diagramme d'activité est très simple car il n'y a qu'une seule action de type primitif à réaliser : création d'un contact en base. Toutefois, BLU AGE permet de gérer des actions multiples ainsi que d'ajouter des conditions. Après avoir installé et exécuté BLU AGE (voir encadré), créez un projet : "file>new>other" puis sélectionnez "BLU AGE Project>BLU AGE Project". Cliquez ensuite sur "next". Sur la boîte de dialogue définissez le nom de votre projet, par exemple "telDirectory" puis cliquez sur "next". Complétez les informations demandées : [Fig.3]

- SGBD : serveur de base de données,
- Driver : automatique, driver pour la connexion à la base,
- Connexion URL : chaîne de connexion à la base de données, remplacez le nom de la base par le nom que vous utiliserez,
- User name et Password : utilisateur et mot de passe d'un utilisateur de la base de données autorisé,
- JDBC : Niveau de compatibilité du driver JDBC.

Cliquez sur "finish" et le projet sera créé dans l'environnement BLU AGE BUILD. Il contient 4 dossiers :

- mockup : pour stocker l'ensemble des pages XHTML paramétrées du modèle,
- model : pour stocker le modèle UML au format EMF,
- ressources : qui contiendra le PSM généré par BLU AGE,
- specific : pour stocker les biblio-

thèques spécifiques que l'on souhaite associer au projet.

Les éléments du modèle sont disponibles sur le CD. Pour les importer, il suffit de faire un copier/coller des contenus des dossiers dans l'environnement Eclipse.

Paramétrez la maquette XHTML en appliquant des tags BLU AGE afin d'associer les éléments du modèle à la présentation. Cette action est réalisée à l'aide de l'environnement BLU AGE BUILD. Dans l'exemple de la [Fig.4] nous avons paramétré le champ de saisie du formulaire à l'aide du tag BLU AGE <bluage:TextField> en précisant que ce champ permet de renseigner l'attribut "Contact.lastName" de l'instance "myContact".

Génération et test

Une fois la maquette entièrement paramétrée, lancez la génération. Faites un clic droit sur la ressource "bluage.workflow" puis sélectionnez "Run As>BLU AGE Launch Generation" [Fig.5], sur la boîte de dialogue qui s'ouvre, gardez les options par défaut et cliquez sur "finish". La génération démarre. Vous pourrez voir la progression dans une console Eclipse. Après quelques secondes l'application est générée. 3 projets ont été ajoutés au workspace Eclipse : telDirectory-entities, telDirectory-service et telDirectory-web.

L'environnement permet de déployer les applications générées sur un environnement Apache Tomcat 5.5 pour les tester. Faites un clic droit sur le projet "telDirectory-web", sélectionnez "Run As" puis "Run on Server", choisissez Apache Tomcat 5.5 Ser-

Installation de l'environnement

Pré requis (voir document de prérequis sur le CD)

- Windows XP/2003, 2Go mémoire avec 500Mo d'espace.
- JDK 1.6 (<http://java.sun.com/javase/downloads/index.jsp>)
- Tomcat 5.5.25 (<http://tomcat.apache.org/>)
- SGBD (SQL Server ou autre SGBD)
- Modeleur UML permettant l'exportation au format EMF 2.0 (ici nous utilisons MagicDraw 15.5, vous pouvez demander une version d'évaluation par mail : magicdraw@bluage.com ou par téléphone au +33 1 56 05 60 91)

Installation (voir guide d'installation sur le CD)

- Double cliquez sur le fichier bluage_build_327.exe du répertoire du CD et suivez les instructions.

Lancement :

- Menu Démarrer puis "Programmes>NETFECTIVE>BLUAGE> BLU AGE BUILD"

Support BLU AGE :

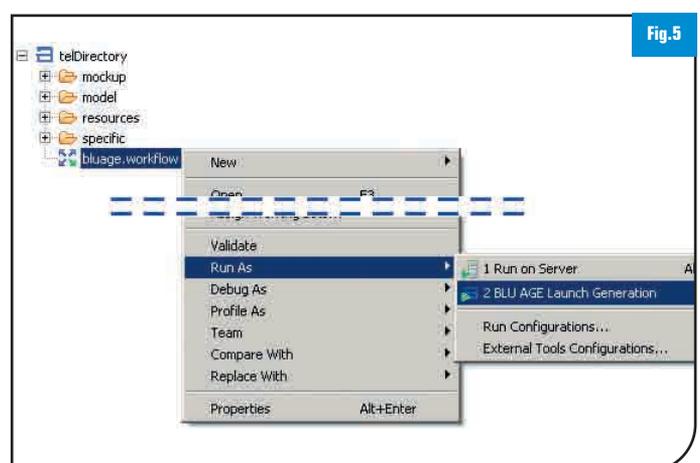
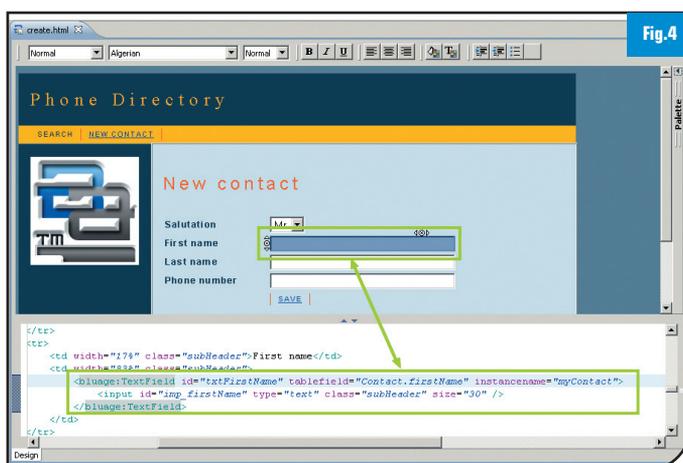
- Par téléphone : +33 1 56 05 60 91
- Par e-mail : Bluage.support@bluage.com

ver, paramétrez le serveur (choisissez le JRE et renseignez le répertoire d'installation de Tomcat). Une fois le paramétrage finalisé, l'application s'exécute. Si vous souhaitez que Hibernate construise automatiquement le schéma de la base de données, ouvrez le fichier "telDirectory-entities>src>hibernate.cfg.xml" et donnez la valeur "update" à la propriété "hibernate.hbm2ddl.auto". Vous trouverez sur la distribution présente sur le CD de nombreux autres exemples. Ils sont accessibles en sélectionnant "Help>Cheat Sheets" puis en sélectionnant "BLU AGE Tutorial". D'autres tutoriels sont disponibles en ligne sur : <http://www.bluage.com/programmez>

■ Jean-Luc Recoussine
BLU AGE Institute Manager

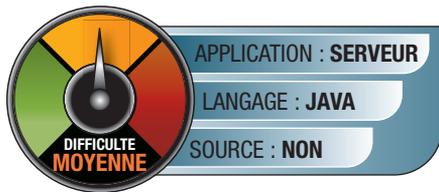
Figure 4 :
Paramétrage
de la maquette

Figure 5 :
configuration
de la génération



Terracotta : une approche différente du clustering

Au cours de la dernière décennie, les systèmes informatiques ont enregistré une montée en charge foudroyante, et de nouvelles problématiques se sont posées. Aujourd'hui, les technologies de type EJB, JMS, Jini montrent leurs limites, tandis que le trafic réseau et les capacités de traitement des bases de données arrivent à saturation. Dans cet article, nous verrons comment Terracotta peut soulager les architectures Java, grâce à une approche radicalement différente.



Au préalable, faisons un tour rapide des principaux problèmes rencontrés lorsqu'une application doit supporter une forte montée en charge. Nous prendrons ici

l'exemple d'une application web. Lorsque le nombre de clients simultanés augmente et qu'un unique serveur web ne suffit plus, il devient nécessaire de mettre en place un cluster, c'est-à-dire un agrégat de serveurs se répartissant le traitement des requêtes de manière transparente pour l'utilisateur. Pour assurer la sécurité des données de session des utilisateurs et améliorer la résilience générale du système, les serveurs disposent de mécanismes de réplication. Mais ceux-ci souffrent de sérieuses limitations :

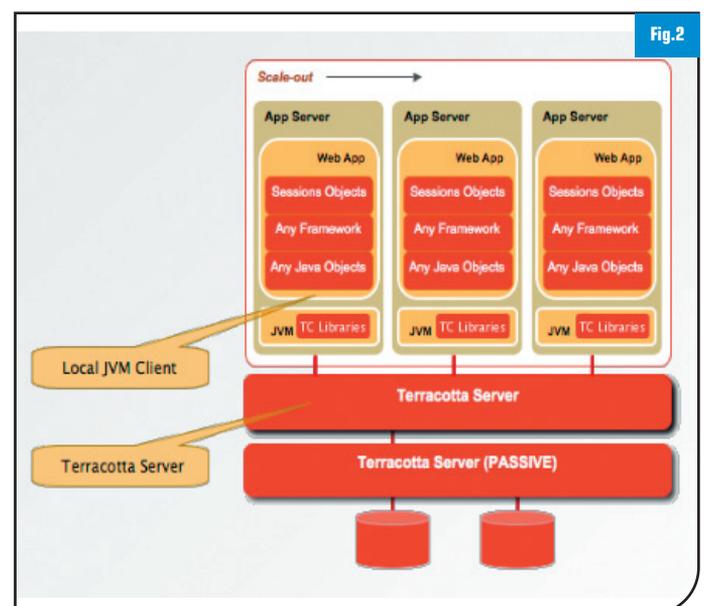
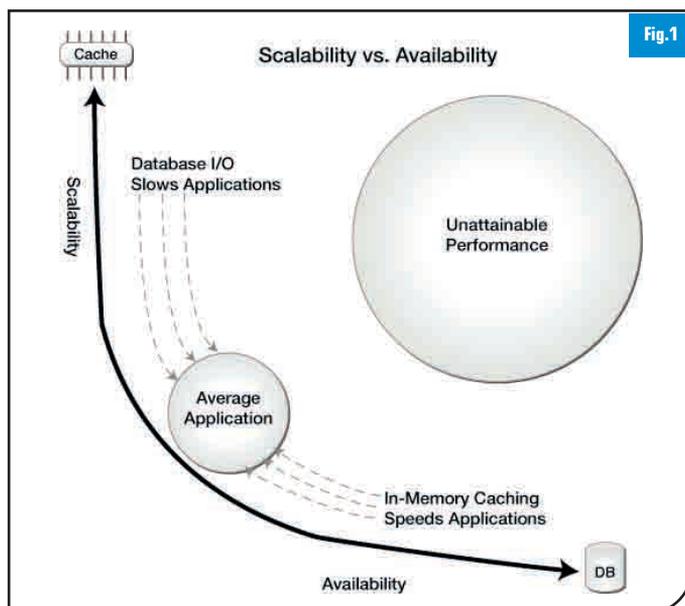
- Ils sont peu efficaces car ils reposent sur la sérialisation standard de Java, qui ne permet d'échanger que des objets complets, et pas de simples différentiels d'état.
 - Ils saturent rapidement le réseau car le nombre de liaisons inter-serveurs augmente exponentiellement avec le nombre de serveurs.
- Conscients de ces limitations et très sensibles à la sécurisation des données, les architectes estiment généralement plus prudent de stocker toutes les informations, mêmes temporaires, dans la base

de données. Les architectures REST vont par exemple dans ce sens, en promouvant des couches de présentation totalement "stateless". Mais cette décision a des impacts importants sur les performances : plus sûre mais infiniment moins réactive que la mémoire vive et supportant mal la mise en cluster, la base de données devient alors le principal goulet d'étranglement de l'application.

Pour éviter les requêtes inutiles et conserver des performances acceptables, il est donc nécessaire de mettre en place des caches distribués, qui posent à leur tour des problèmes de réplication... Le travail de l'architecte SI consiste donc à trouver le juste milieu entre la performance et la sécurité des données, tout en maîtrisant la charge des serveurs et la densité du trafic réseau. [Fig.1]

1 ARCHITECTURE DE TERRACOTTA

Terracotta est un middleware open-source pour Java de type NAM (Network Attached Memory). De la même manière qu'un NAS (Network Attached Storage) permet à plusieurs machines d'accéder aux mêmes fichiers sur un disque partagé, un NAM permet à plusieurs applications (ou plusieurs instances d'une même application) d'accéder aux mêmes graphes d'objets au sein d'une mémoire virtuelle partagée. Afin de limiter le trafic réseau, il s'emploie à maintenir les



données au plus près du code qui les utilise. Ainsi, lorsqu'une application obtient un lock sur un sous-ensemble du graphe d'objet commun, ce sous-ensemble est monté (ou maintenu) au niveau de la JVM de l'application, qui y accède alors localement. Inversement, toute modification apportée aux objets partagés est immédiatement répliquée sur le NAM pour des raisons de sécurité.

Pour toutes les problématiques de mise en cluster d'applications, de répartition de charge (datagrid) ou de distribution de données (caches), cette solution présente de nombreux avantages. Premièrement, le développement et la maintenance des applications en sont fortement simplifiés, puisqu'elles ne dépendent plus d'une API explicite pour s'échanger des données (JMS, RMI, SOAP..) : celles-ci peuvent être consultées et modifiées de manière transparente, comme si elles étaient disponibles localement. Ensuite, il n'y a plus besoin de répliquer préventivement les données : elles sont sécurisées au sein du NAM, et seront transmises, à la demande et de manière optimisée, aux seules applications devant y accéder. Enfin, ce système autorise la manipulation de structures qui seraient habituellement trop volumineuses pour tenir dans la mémoire locale des applications : les informations nécessaires sont fournies en temps réel par le NAM.

Techniquement, Terracotta se compose de deux éléments :

- Un serveur indépendant chargé de la gestion de la mémoire partagée. Afin de garantir une reprise sur incident immédiate, toutes les données sont sécurisées sur disque de manière asynchrone, et le serveur lui-même peut être mis en redondance active ou passive.
- Un " *bootClassLoader* " spécial qui instrumente la JVM des applications clientes et assure la liaison avec le serveur Terracotta. Un fichier de configuration " tc-config.xml " permet alors de définir quels graphes d'objets doivent être déployés sur le NAM (il est également possible d'utiliser des annotations).

La clusterisation est donc fournie comme un simple service d'infrastructure, qui ne nécessite aucune modification du code (de fait, vous ne verrez aucun extrait de code Java dans cet article) et aucun apprentissage particulier de la part des développeurs. [Fig.2]

L'*instrumentation* de la JVM et des classes partagées autorise de nombreuses optimisations. Par exemple, contrairement au mécanisme de sérialisation Java qui transmet des objets complets, Terracotta travaille à un niveau plus fin et ne transmet que des différentiels, de manière compressée et par batches. Les applications instrumentées peuvent également être monitorées par la console d'administration de Terracotta, afin de suivre en temps réel l'état du cluster, le volume des données échangées, repérer les " deadlocks " distribués, etc.

Pour finir, Terracotta propose des modules d'intégration (TIM, Terracotta Integration Module), qui fournissent toute la configuration nécessaire pour s'intégrer aux principaux frameworks d'entreprise : Spring, Hibernate, EHCACHE, Struts...

Actuellement, Terracotta est déployé avec succès chez des grands comptes de la finance et des services (WalMart, Comcast, JPMorgan, Pearson...).

2 UN PEU DE PRATIQUE

Clusteriser ses sessions HTTP

Voyons maintenant l'un des principaux cas d'utilisation de Terracotta, qui est également l'un des plus simples à mettre en œuvre : la

clusterisation de serveurs web et le partage de leurs sessions HTTP. En interne, chaque serveur web gère ses sessions HTTP sous la forme d'une Map, lui permettant de retrouver les données d'un utilisateur particulier à partir de son identifiant de session (jsessionid). La solution proposée par Terracotta consiste à déployer sur le NAM une seule et unique Map, commune à tous les serveurs du cluster et contenant l'ensemble des sessions HTTP. Ainsi, tout ajout, suppression ou modification (ajout ou suppression d'un attribut, expiration...) d'une session leur sera immédiatement visible.

Comment faire en pratique ? Comme nous l'avons vu plus haut, pour clusteriser un graphe d'objets, il suffit d'en déclarer la racine dans le fichier de configuration. La Map des sessions est malheureusement un détail d'implémentation qui varie d'un serveur web à l'autre, mais Terracotta dispose de réglages prédéfinis pour les principaux produits du marché (Tomcat, Websphere, Weblogic, Jboss, Jetty...), et effectue automatiquement les déclarations adéquates. Il ne reste plus au développeur qu'à demander l'instrumentation de ses propres classes, et à indiquer le nom de l'application web à clusteriser. Pour vous convaincre de la simplicité de mise en œuvre de cette technologie, voici la configuration suffisante pour paramétrer une application web :

```
<tc:tc-config xmlns:tc="http://www.terracotta.org/config">
  <servers>
    <server host="%i" name="ZenikaServer">
      <data>data/server-data</data>
      <logs>logs/server-logs</logs>
    </server>
  </servers>
  <clients>
    <logs>logs/client-logs/%(webserver.log.name)</logs>
  </clients>
  <application>
    <dso>
      <instrumented-classes>
        <include>
          <class>com.zenika.gallery.model.SharedAlbum</class>
        </include>
        <include>
          <class>com.zenika.gallery.model.Image</class>
        </include>
      </instrumented-classes>
      <web-applications>
        <web-application>MaGallery</web-application>
      </web-applications>
    </dso>
  </application>
</tc:tc-config>
```

La configuration commence par la définition de l'emplacement des fichiers de la base de données interne de Terracotta, ainsi que des fichiers de log du serveur et des clients.

Viennent ensuite les classes devant être instrumentées : ici, nous avons déclaré "SharedAlbum" et "Image", car l'application met en place des instances en session. Pour finir, la spécification du nom de l'application web permet à Terracotta de déterminer la Map des sessions associée et de la partager.

La mise en œuvre du clustering de sessions HTTP est donc très

simple. Toutefois il faut garder à l'esprit certaines considérations d'architecture. Comme indiqué plus haut, Terracotta tente de maintenir les données au plus proche de l'application. C'est pourquoi il est fortement conseillé d'utiliser un répartiteur de charge gérant l'affinité de session : ainsi, un utilisateur donné est toujours servi par le même serveur, qui dispose déjà localement de ses données de session. Si un autre serveur était sélectionné, il acquerrait à son tour le lock sur ces données et Terracotta devrait les lui transmettre, générant du trafic réseau inutile.

Partager des données entre applications

Nous avons vu qu'il était facile de partager un graphe d'objets entre plusieurs instances d'une même application.

Nous allons maintenant démontrer que c'est également réalisable entre des applications différentes, pourvu qu'elles partagent une structure commune de données.

Nous prendrons l'exemple d'une application permettant de sélectionner et d'afficher des images, déclinée en deux versions : une version web écrite en Wicket et déployée sous Tomcat et Windows, et une version "standalone" proposant une interface Swing sous Mac OS X. Le choix d'un album dans le navigateur provoque l'apparition de l'image correspondante dans l'application Swing, et il est également possible de cocher la case "Slideshow", afin de déclencher l'affichage successif de toutes les images de l'album sur le client lourd. [Fig.3]

Voyons maintenant les fichiers de configuration des deux applications, très similaires à celui de l'exemple précédent. Cette fois, il nous incombe de spécifier les racines des graphes d'objets à partager, ce qui se fait très simplement grâce à la balise <root>. Ce qui est intéressant dans le cas présent, c'est que les propriétés "Displayer.album" du client Swing et "AlbumPage.album" du client web, toutes deux de type "SharedAlbum", sont configurées sous le même identifiant "album". Terracotta permettra alors aux deux applications de manipuler strictement le même objet en mémoire partagée, de sorte que toute modification de son état par l'une sera instantanément visible par l'autre.

Voici le tc-config.xml de l'application Swing :

```
<roots>
  <root>
    <field-name>com.zenika.gallery.Displayer.album</field-name>
    <root-name>album</root-name>
  </root>
</roots>
```



Fig.3

Et celui de l'application Wicket :

```
<roots>
  <root>
    <field-name>org.londonwicket.gallery.AlbumPage.album</field-name>
    <root-name>album</root-name>
  </root>
</roots>
```

La console d'administration de Terracotta, extrêmement complète, permet de visualiser en temps réel la configuration du NAM et de monitorer sa santé. Sur la capture d'écran ci-dessous, on peut ainsi reconnaître le serveur Terracotta, les deux clients connectés (Wicket et Swing) ainsi que les objets partagés, dont notre SharedAlbum déclaré sous l'identifiant "album" ; il est même possible de voir l'état de sa propriété "slideShow", qui indique que la case à cocher n'est actuellement pas sélectionnée. [Fig.4]

Dernier point de configuration : de la même manière qu'une application multi-threadée localement, une application clusterisée doit protéger l'accès à ses données partagées par des verrous ("locks"). Afin que Terracotta puisse les appliquer à tout le cluster, il est nécessaire de lui indiquer les méthodes en faisant usage. C'est le rôle de la section <locks> du fichier de configuration :

```
<autolock auto-synchronized="true">
  <method-expression>
    java.lang.String com.zenika.gallery.SharedAlbum.getDefaultImageUri()
  </method-expression>
  <lock-level>read</lock-level>
</autolock>

<autolock auto-synchronized="true">
  <method-expression>
    void com.zenika.gallery.SharedAlbum.updateSlideShowStatus(java.lang.Boolean)
  </method-expression>
  <lock-level>write</lock-level>
</autolock>
```

Caches distribués

Pour finir, intéressons-nous à la problématique de gestion des caches. Nous prendrons ici l'exemple d'un index permettant de catégoriser et de rechercher rapidement un fichier au sein d'un disque. Pour information, il est tiré d'un cas réel gérant plusieurs



Fig.4

milliards d'entrées avec des temps de réponse de quelques millisecondes. La solution classique consiste à stocker cet index dans une base de données, ce qui pose de multiples problèmes au fur et à mesure que le nombre d'entrées croît; en particulier, les temps d'accès, au départ corrects, vont augmenter rapidement avec la taille de l'index.

De plus, la base ne peut supporter qu'un nombre limité de requêtes et les coûts successifs en temps et matériel nécessaires pour repousser cette limite sont loin d'être négligeables.

Conserver la totalité de cet index en mémoire (sans Terracotta) est facilement réalisable avec une simple Map et quelques gigaoctets de RAM. Les temps d'accès s'en trouveraient grandement réduits mais se poserait alors la question de la fiabilité de l'index.

C'est précisément là que Terracotta intervient : en utilisant une Map conçue pour les accès concurrents, par exemple une `ConcurrentHashMap`, et en la partageant sur un cluster, on peut profiter des avantages offerts par la RAM tout en comblant ses lacunes, en particulier en termes de fiabilité: l'index peut être virtuellement répliqué sur chaque nœud du cluster et une copie en base est automatiquement gérée par le serveur Terracotta.

Un système de **load-balancing** simple permet alors de tirer parti de toute la puissance de traitement des nœuds, et de rediriger automatiquement le trafic sur un nœud valide en cas de défaillance afin d'assurer la continuité du service. De plus, il suffit d'ajouter des nœuds pour supporter plus de requêtes sur l'index ("scale-out").

La solution simple présentée ci-dessus possède tout de même quelques inconvénients: en cas de modification de l'index, le serveur Terracotta acquiert un verrou en écriture sur la Map et aucune autre opération ne peut être effectuée simultanément tant que ce verrou n'a pas été retiré.

Les performances globales de l'index risquent donc de s'effondrer rapidement pour quelques modifications de l'index. Pour pallier ce problème, une solution consiste à remplacer la structure unique de stockage par une structure personnalisée avec un mécanisme de verrouillage partiel et localisé. Celle-ci doit être ordonnée et plus fragmentée de manière à ce que les verrous posés par le serveur n'affectent qu'une faible proportion des données. Il faut aussi veiller à ce que la structure puisse supporter des accès fortement concurrents et ce sans apparition d'inter-verrouillages bloquants ("deadlocks"). Il s'agit donc de créer et synchroniser un arbre de Maps constitué de Maps ordonnées.

Là encore, la transparence de Terracotta s'impose comme un de ses points forts : même si la structure de données ou l'algorithme est complexe, il vous suffit de les développer comme d'habitude, puis de mettre au point la configuration associée, comme vu précédemment :

désignation des classes partagées à instrumenter, affectation des verrous en lecture ou en écriture sur les méthodes concernées, et finalement paramétrage de la racine de l'arbre des Maps comme racine de graphe d'objets déployée sur le cluster.

Au final, on dispose d'un index totalement en mémoire, mais facilement extensible et dont la fiabilité est garantie. Les temps d'accès aux données brutes sont réduits de plusieurs ordres de grandeur par rapport à une base de données, et des accès simultanés en lecture intra- et inter-JVM sont autorisés. Les caches de toutes les instances de l'application sur le cluster sont centralisés et partagés de manière transparente comme un cache unique, et l'ajout de nouvelles machines permet d'augmenter la charge maximale supportée de manière quasi-linéaire. Grâce à Terracotta, la notion de "cache as a service" reprend donc tout son sens.

3 CONCLUSION

En conclusion, Terracotta est un middleware open-source qui permet de considérer la mise en cluster comme un simple service d'infrastructure, transparent pour les applications et les équipes de développement. A travers trois cas d'utilisation typiques, nous avons démontré sa puissance et sa facilité de mise en œuvre.

Maintenant, à vous de jouer ! Terracotta n'étant pas intrusif, il est facile de le tester sans risque. Nous vous invitons donc à le télécharger et à lire sa documentation sur <http://www.terracotta.org>.



■ Olivier Croisier
Architecte chez Zenika



■ Guillaume Tinon
Consultant chez Zenika, société spécialisée dans les architectures Java / JEE. www.zenika.com

glossaire

Bootclassloader ou "**bootstrap classloader**" : élément de la machine virtuelle responsable du chargement des classes Java bas niveau (Object, String...).

Instrumentation : consiste à modifier dynamiquement le code des classes lors de leur chargement par un classloader, généralement à des fins de collection de statistiques, d'ajout de traces, ou d'analyse dynamique.

Load-balancing : mécanisme par lequel la charge de travail est répartie, si possible équitablement, entre les différents serveurs d'un cluster.

Abonnez-vous à **e-PROgrammez!** Le magazine du développement

2,7 € seulement par numéro au format PDF (pour le monde entier)

et pour **0,84 €** de plus par mois : abonnement illimité aux archives (numéros du magazine, et articles, en PDF)

Abonnez-vous sur www.programmez.com

Maîtriser les fondamentaux de PHP!

Nous poursuivons dans cette 2^e partie la découverte du langage PHP. Nous aborderons les opérateurs, les structures de contrôles et les boucles. Bonne programmation avec PHP !



LES OPÉRATEURS

Les opérateurs sont des symboles qui permettent de manipuler des variables. On peut notamment effectuer des opérations, affecter ou comparer des valeurs, etc.

Opérateurs d'affectation

L'opérateur d'affectation est le signe égal. La variable avant le symbole égal (=) prend la valeur spécifiée après.

Opérateurs arithmétiques

Les opérateurs arithmétiques en PHP ne nécessitent pas de présentation particulière, car ce sont les opérateurs mathématiques traditionnels.

Opérateurs arithmétiques

Opérateur	Opération	Exemple	Description	Résultat
+	Addition	echo \$a + \$b;	Calcule la somme	13
-	Soustraction	echo \$a - \$b;	Calcule la différence	5
*	Multiplication	echo \$a * \$b;	Calcule le produit	36
/	Division	echo \$a / \$b;	Calcule la division	2.25
%	Modulo	echo \$a % \$b;	Calcule le modulo	1

Incrémementation

PHP possède quelques raccourcis pour vous éviter d'écrire des instructions pour incrémenter telles que : \$i = \$i+1; Utilisez ++ ou -- respectivement pour incrémenter ou décrémenter une variable d'une seule unité.

```
<?php
    $i = 1;
    $i = $i+1; // Est équivalent à :
    $i++;

    $i = $i-1; // Est équivalent à :
    $i--;
?>
```

Ces opérateurs peuvent figurer devant ou derrière la variable :

++\$a : la variable est incrémentée puis évaluée,

\$a++ : la variable est évaluée puis incrémentée.

La concaténation

L'opérateur de concaténation est le point ".". Cette opération est très importante car elle est souvent utilisée. Cet opérateur vous permet par exemple d'empiler des informations dans une variable.

```
<?php
    $table = 'users';
    $id     = 5;
    $sql    = 'SELECT * FROM ' . $table . ' WHERE id='.$id;
    // $sql est égal à "SELECT * FROM users WHERE id=5"

    $var    = 'hello ';
    $var .= 'world';
    // $var est égal à 'hello world'
?>
```

Opérateurs de comparaison

Les opérateurs de comparaison sont principalement utilisés en combinaison avec les structures conditionnelles (if, for, while, etc.) que nous verrons peu après. Ces opérateurs permettent de comparer deux termes et de renvoyer un booléen vrai ou faux (TRUE ou FALSE) selon la véracité de la comparaison.

Opérateurs de comparaison

Opérateur	Signification	Exemple	Description
==	Égal à	\$a == \$b;	Renvoie TRUE si \$a est égal à \$b.
<	Inférieur à	\$a < \$b;	Renvoie TRUE si \$a est plus petit que \$b.
>	Supérieur à	\$a > \$b;	Renvoie TRUE si \$a est plus grand que \$b.
<=	Inférieur ou égal à	\$a <= \$b;	Renvoie TRUE si \$a est inférieur ou égal à \$b.
>=	Supérieur ou égal à	\$a >= \$b;	Renvoie TRUE si \$a est supérieur ou égal à \$b.
!=	Différent de	\$a != \$b;	Renvoie TRUE si \$a est différent de \$b.
===	Égal à, en type et en valeur	\$a === \$b;	Renvoie TRUE si \$a est égal à \$b et si les deux variables ont le même type.
!==	Différent en valeur ou différent en type	\$a !== \$b;	Renvoie TRUE si \$a a une valeur différente de celle de \$b ou si \$a n'a pas le même type que \$b.

Opérateurs logiques

Les opérateurs logiques servent énormément dans les structures de contrôle. Le tableau suivant présente les plus courants :

Opérateurs logiques

Opérateur	Exemple	Évalué à vrai (TRUE) quand :
!!	\$b;	\$b ne renvoie pas TRUE.
&&	\$a && \$b;	\$a et \$b renvoient TRUE.
	\$a \$b;	Au moins l'un des deux renvoie TRUE.
AND	\$a AND \$b;	\$a et \$b renvoient TRUE.
OR	\$a OR \$b;	Au moins l'un des deux renvoie TRUE.
XOR	\$a XOR \$b;	L'un des deux, et uniquement l'un des deux, renvoie TRUE.

STRUCTURES DE CONTRÔLE

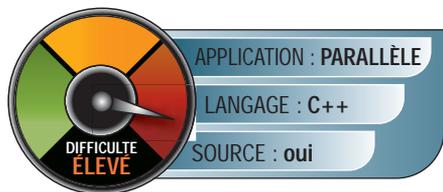
Les structures de contrôle permettent de répéter certaines actions ou de soumettre certaines exécutions à des conditions. En PHP, leur syntaxe est similaire à celle du langage C. Ces structures fonctionnent pour la plupart à partir d'un test. Ce test est une expression qui doit renvoyer une valeur comprise comme un booléen. Le plus souvent, on utilisera les opérateurs logiques et de comparaison, mais il est possible d'avoir une expression complexe comprenant des appels de fonctions et des affectations de variables.

Les conditions

Dans vos scripts, il sera important de pouvoir effectuer une prise de décision, et donc de poser des conditions à l'exécution de telle ou telle action.

Découvrir et mettre en œuvre CUDA

Si votre PC est équipé d'une GPU NVidia, la librairie CUDA vous permet d'en faire un monstre de calcul parallèle. Nous continuons la découverte des bases de la programmation CUDA commencée le mois dernier.



CUDA est une librairie de programmation parallèle. Une de plus ? Pas tout à fait... Tandis que les librairies de programmation parallèles "classiques" vont

répartir le travail sur les CPU présentes sur une machine ou même un cluster de machines, avec une planification logicielle, CUDA se propose d'exploiter les extraordinaires ressources des co-processeurs graphiques, ou GPU, NVidia. En effet, alors qu'un PC dernier cri se verra doté d'un Microprocesseur à 4 cœurs, une carte graphique NVidia de milieu de gamme au moment de la rédaction de cet article, une 8800 GT par exemple, va comporter 14 multiprocesseurs. Un multiprocesseur est un ensemble de 8 processeurs simples. Au total, vous disposez, l'auriez vous soupçonné, d'une unité de calcul à 112 processeurs, ce qui est 28 fois plus puissant que votre CPU dernier cri. 28 fois plus puissant ? En fait bien plus que cela, car si les librairies de calcul parallèle classiques planifient et synchronisent les threads de manière logicielle, ce travail est essentiellement effectué directement par la GPU, au niveau matériel, ce qui induit des performances sans équivalent. Entrons un peu plus dans les détails. Les multiprocesseurs disposent de 8192 registres à partager entre les threads.

Sur une 880GT, les blocs peuvent compter jusqu'à 512 threads. Un multiprocesseur peut supporter 8 blocs de threads et le composant comporte 14 multiprocesseurs, ce qui donne un nombre de 57 344 threads. Bien sûr ceci dans des conditions optimales, c'est-à-dire si le nombre de registres ainsi que la mémoire partagée sont suffisants pour faire tourner les threads. Sinon, le composant réduit le nombre de threads simultanés, ce qui en général laisse quand même une formidable puissance à disposition. Ainsi des projets de calculs très lourds, tels que Folding@Home ont vu leurs temps de calcul réduits par un facteur 100 grâce à CUDA. Bien sûr il y a aussi un lot d'inconvénients. Un code parallélisé s'exécutant dans la GPU, il est impossible d'invoquer depuis celui-ci une API du système hôte. C'est une limitation de taille. D'autre part, CUDA est ainsi fait qu'un maximum d'efficacité sera atteint avec un très grand nombre de threads, ce qui rend la conception et le développement des applications difficiles. CUDA fera merveille pour des calculs facilement parallélisables, une convolution d'images par exemple, et sera moins intéressant dans d'autres situations.

Et il est des cas où CUDA n'aura même aucun intérêt. Le mois dernier, nous avons vu l'utilisation la plus élémentaire en écrivant du code s'exécutant dans un seul thread. Aujourd'hui, nous allons essayer de mieux cerner les mécanismes internes de CUDA pour une bonne manipulation des threads. Nous espérons ainsi fournir le tutoriel qui manque cruellement à la documentation de CUDA. Les exemples ont été compilés sous Windows et avec Visual Studio 2005. Nous rappelons que Visual Studio 2008 ne convient pas pour travailler avec CUDA. En revanche, le compilateur libre MinGW

convient très bien. Sous Linux, GCC fait évidemment l'affaire. Le lecteur voudra bien se reporter à l'article du mois dernier pour la manipulation du frontal nvcc.

1 COMPRENDRE LE PRINCIPE DES BLOCS ET GRILLES DE THREADS

Comme les généralités données plus haut à propos de l'architecture des GPU NVidia laissent entendre, les threads ne sont pas lancés les uns après les autres par une application mais par groupe. Quel que soit le nombre de threads lancés, ceux-ci font partie d'une grille et dans cette grille sont rangés des blocs de threads. Imaginons une feuille de papier avec des carreaux. La feuille est la grille et les carreaux les blocs. Ceci pour avoir une représentation simple de la chose. Mais en réalité les choses sont plus complexes. La grille et les blocs sont des objets à trois dimensions. La troisième dimension d'une grille vaut toujours 1, donc finalement la grille est toujours plane, mais sa troisième dimension existe bel et bien. Le bloc quant à lui peut être effectivement tridimensionnel. Pourquoi est-ce aussi compliqué, du moins en apparence ? Il ne faut pas oublier que les GPU sont avant tout destinées à travailler des images. Une image se traduit en mémoire par une grille de points. Voici la raison d'être de notre grille de threads. Les traitements peuvent être effectués par petites zones, comme avec une mosaïque. Voici la raison d'être des blocs de threads. Enfin notre grille de points peut avoir une profondeur soit pour rendre des couleurs ou pour faire des effets de superposition. C'est ce qui explique qu'un bloc de threads puisse avoir trois dimensions de valeur différente de 1. Nous avons compris, une image constituée de pixels est mappée vers autant de threads pour son traitement au sein de la GPU.

2 PREMIER EXEMPLE MULTITHREAD

Pour notre premier exemple, nous voulons lancer 9 threads. Comprendre: nous allons créer une grille à une dimension contenant un bloc de threads à une dimension valant 9, ce qui nous fera 9 threads. Les deux autres dimensions valant bien évidemment 1. Que devons nous faire des threads ? Nous allons initialiser une zone de mémoire partagée contenant neuf vecteurs à trois dimensions. Chaque thread affectera les coordonnées d'un vecteur par sa position dans le bloc. Voici le code (DemoldThreads sur le site) :

```
#include <stdio.h>

#include <cuda.h>
#include <vector_types.h>

const int nb_threads = 9;

void init() {
    CUresult result = cuInit(0);
```

```

switch(result) {
case CUDA_SUCCESS:
    puts("Librairie CUDA initialisee correctement");
    break;
case CUDA_ERROR_INVALID_VALUE:
case CUDA_ERROR_NO_DEVICE:
default:
    puts("Une erreur s'est produite à l'initialisation de CUDA");
    exit(1);
}
}

__global__ void setVec(int3 *v) {
int i = threadIdx.x;
v[i].x = threadIdx.x;
v[i].y = threadIdx.y;
v[i].z = threadIdx.z;
}

void run(int3 *v_array) {
int3 *gpu_v_array;
int size;

size = sizeof(int3) * nb_threads;
cudaMalloc((void**)&gpu_v_array, size);
cudaMemset((void *)gpu_v_array, -1, size);
setVec<<<1, nb_threads>>>(gpu_v_array);
cudaMemcpy(v_array, gpu_v_array, size, cudaMemcpyDeviceToHost);
cudaFree(gpu_v_array);
}

void print_vector(int3 *v) {
printf("x: %d, y: %d, z: %d\n", v->x, v->y, v->z);
}

int main(int argc, char** argv) {
int3 v_array[nb_threads];

puts("Cude DemoIdThreads -- Programmez!");
init();
run((int3*)&v_array);
for(int i=0; i<nb_threads; i++)
    print_vector(&v_array[i]);
return 0;
}

```

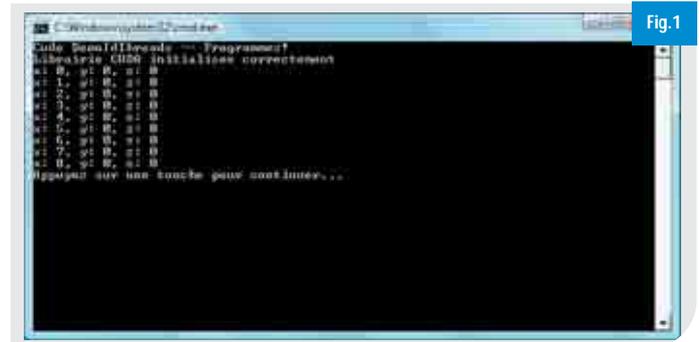


Fig.1

Notre premier exemple au travail. Neuf threads rangés dans un bloc à une dimension.

dimension en x et numérotés 0 dans les dimensions en y et z. Pour lancer le kernel nous avons écrit ceci :

```
setVec<<<1, nb_threads>>>(gpu_v_array);
```

mais cette écriture, que l'on trouve dans le premier exemple de la documentation n'est en fait qu'un sucre syntaxique. Les deux arguments passés entre les chevrons sont de types dim3, un type CUDA ainsi déclaré :

```

struct dim3 {
    unsigned int x, y, z;
#ifdef __cplusplus
    dim3(unsigned int x = 1, unsigned int y = 1, unsigned int z = 1)
: x(x), y(y), z(z) {}
    dim3(uint3 v) : x(v.x), y(v.y), z(v.z) {}
    operator uint3(void) { uint3 t; t.x = x; t.y = y; t.z = z; return t; }
#endif /* __cplusplus */
};

```

Donc en C++ le code complet (cf. exemple sur le site) est :

```

dim3 le_block(9, 1, 1);
dim3 la_grille(1, 1, 1);
setVec<<<la_grille, le_block>>>(gpu_v_array);

```

et en C :

```

#undef __cplusplus
dim3 le_block;
dim3 la_grille;
le_block.x = 9;
le_block.y = 1;
le_block.z = 1;
la_grille.x = la_grille.y = la_grille.z = 1;
setVec<<<la_grille, le_block>>>(gpu_v_array);
#define __cplusplus

```

Ce code est simple et pour l'essentiel semblable au code à un thread du mois dernier. Le coeur de ce programme est le kernel setVec. On appelle kernel une fonction s'exécutant sur la GPU. Le code de ce kernel est simple. Tout thread détient une variable threadIdx. Idx pour Index, et non x pour coordonnée en x. L'appellation est trompeuse. En fait comme la variable threadIdx est censée renseigner sur la position du thread dans son bloc, c'est nécessairement une entité à trois dimensions.

Et effectivement son type CUDA est int3 c'est-à-dire un vecteur d'entiers à trois dimensions. L'illustration ci-dessus [Fig.1] montre notre application au travail. Neuf threads numérotés de 0 à 8 dans la

Il est ainsi on ne peut plus clair que grille et blocs sont des structures de données à trois dimensions systématiquement.

3 TRAVAILLER AVEC UN BLOC MULTIDIMENSIONNEL

Ce travail va mettre en évidence une particularité de CUDA. Nous allons créer un bloc tridimensionnel contenant 27 threads et créer en mémoire partagée de l'espace pour 27 vecteurs tridimensionnels. Chaque thread affectera les coordonnées de "son" vecteur avec les coordonnées de sa position dans le bloc tridimensionnel. Cet exemple s'appelle DemoBlock sur notre site :

Ce code montre que CUDA se comporte à l'encontre de ce qu'on attend instinctivement. Ainsi on pourrait croire que les positions des threads dans le bloc correspondent à ce que l'on aurait par trois boucles imbriquées, comme ceci :

```
int limite = 3;
for(int x=0; x<limite; x++)
  for(int y=0; y<limite; y++)
    for(int z=0; z<limite; z++)
      printf("x: %d, y: %d, z: %d\n", x, y, z);
```

Mais en fait le comportement CUDA trouve son équivalent ainsi :

```
int limite = 3;
for(int z=0; z<limite; z++)
  for(int y=0; y<limite; y++)
    for(int x=0; x<limite; x++)
      printf("x: %d, y: %d, z: %d\n", x, y, z);
```

Comme le montre l'illustration ci-dessous [Fig.2]. Ceci explique la formule du calcul d'index dans le tableau de vecteurs du code d'exemple :

```
int index = i + j*3 + k*3*3;
```

Encore une fois si l'on prend le temps de la réflexion, ce comportement sous CUDA est tout à fait logique. La GPU a pour vocation de traiter des images qui peuvent être formées par des couches de plan de pixels. Boucler d'abord sur la coordonnée en z correspond à travailler sur un plan de pixels, puis un autre, et ainsi de suite, tandis que le travail dans un plan est fait par le bouclage sur les coordonnées en y puis en x. Et non en x et y comme, une nouvelle fois, l'instinct pourrait nous le faire croire. Donc tout cela est parfaitement logique, mais il est préférable de bien le garder à l'esprit lorsqu'on écrit du code, sinon les surprises sont de taille.) Attention, ce que nous venons d'expliquer, et en particulier notre analogie avec des indices de boucle ne doit pas induire en erreur. Nous avons parlé du positionnement des threads dans un bloc, en aucun cas cas d'un ordre d'exécution des threads. Cet ordre est absolument imprévisible a priori, puisque nous sommes dans un environnement totalement concurrent. Et dans une grille, comment cela se passe-t-il pour les indices de blocs ? Le principe est le même que pour les threads. L'exemple DemoGrid sur le site illustre cela. Voici un extrait du code :

```
__global__ void printGrid() {
#ifdef __DEVICE_EMULATION__
  printf("Block: %d, %d, %d -- Thread: %d %d %d\n",
         blockIdx.x,
         blockIdx.y,
```

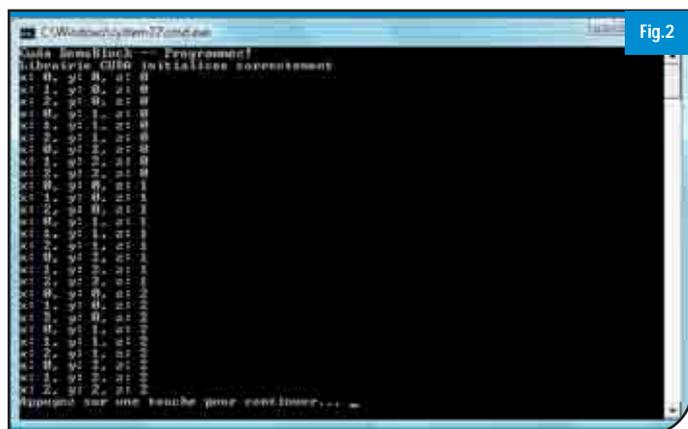


Fig.2

```
blockIdx.z,
threadIdx.x,
threadIdx.y,
threadIdx.z);
#endif
}

void run() {
  dim3 dimBlock(2, 2, 2);
  dim3 dimGrid(2, 2, 1);
  printGrid<<<dimGrid, dimBlock>>>();
}
```

Nous créons des blocs de 8 threads ($2*2*2$) dans une grille $2x2$. Cela fait 32 threads au total. Chaque thread affiche directement sa position dans la grille et le bloc comme on le voit ci-dessous [Fig.3]. Cet exemple montre en outre une particularité du runtime de CUDA. La fonction `printf` directement depuis le kernel. C'est normalement impossible. Pour que ce programme fonctionne il doit être compilé avec l'option.

```
-deviceemu
```

afin que le code soit exécuté non dans la GPU mais dans un émulateur. En outre la macro `__DEVICE_EMULATION__` est définie. Ce procédé ne permet pas de créer de nombreux threads, mais il est bien pratique pour déboguer.

4 SYNCHRONISER DES THREADS

Terminons ce tutoriel avec la question de la synchronisation des threads, car en CUDA comme avec toute bibliothèque concurrente on finit toujours par rencontrer ce problème, surtout dès que l'on aborde les calculs mathématiques, car dans ce cas il est très fréquent qu'un thread ait pour mission de collecter le résultat du travail des autres. Nous simulons cette situation avec notre dernier exemple, dans lequel nous lançons trois threads. Les threads d'indice non nul affectent de cette valeur d'indice, augmentée de un, toutes les coordonnées de respectivement trois vecteurs à trois dimensions. Quand cela est terminé, le thread d'indice nul collecte les résultats selon cette formule

```
coordonnee0 = coordonnee1*2 + coordonnee2*3
```

A la fin du calcul, chaque coordonnée du vecteur 0 doit contenir la valeur 13 ($2*2 + 3*3$); Voici un extrait de l'exemple DemoSynchro qui fait cela :

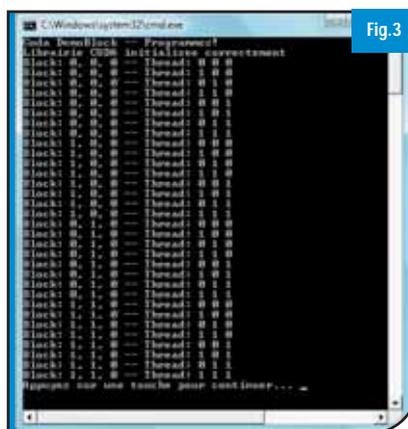


Fig.3

[Fig.2]
Les indices de positions des threads dans un bloc sont logiques, mais contre-intuitifs.

[Fig.3]
Les blocs sont indicés dans une grille de la même manière que les threads dans un bloc.

```

__global__ void compute(int3 *v) {
int id = threadIdx.x;
/* // Mauvais code
if(id == 0) {
    __syncthreads(); // Mauvais code, avec ou sans __syncthreads
    v[id].x = v[1].x*2 + v[2].x*3;
    v[id].y = v[1].y*2 + v[2].y*3;
    v[id].z = v[1].z*2 + v[2].z*3;
}
else {
    for(int i=0; i<50000; i++)
        for(int i=0; i<1000; i++);
    v[id].x = id+1;
    v[id].y = id+1;
    v[id].z = id+1;
}
// fin mauvais code */

// Bon code
v[id].x = id+1;
v[id].y = id+1;
v[id].z = id+1;
if(id!=0) {
    for(int i=0; i<50000; i++)
        for(int i=0; i<1000; i++);
}

// Sans l'appel à syncthreads, le code
// ne se comporte pas correctement
__syncthreads();
if(id == 0) {
    v[id].x = v[1].x*2 + v[2].x*3;
    v[id].y = v[1].y*2 + v[2].y*3;
    v[id].z = v[1].z*2 + v[2].z*3;
}
}

```

En CUDA, hormis les opérations de copie de mémoire, ainsi que nous l'avons mentionné le mois dernier, il existe une seule fonction qui sert de barrière, ou autrement dit, de point de jonction des

threads. Cette fonction se nomme `__syncthreads` et comme toujours avec la programmation concurrente, comporte son lot de pièges. Pour commencer on ne répétera jamais assez que `__syncthreads` ne fonctionne que pour tous les threads d'un même bloc et non pour tous les threads d'une même grille. Enfin, le piège classique dans ce style de programmation est d'invoquer la fonction dans le code dédié à un thread. C'est ce que fait le "mauvais code" placé en commentaires dans l'exemple ci-dessus. L'appel à `__syncthreads` y est placé dans un bloc qui n'est exécuté que pour le thread d'indice (0, 0, 0).

```

if(id == 0) {
    __syncthreads(); // Mauvais
}

```

Ceci ne fonctionne pas, comme en témoigne l'illustration ci-dessous. [Fig.4]. Le principe à respecter est qu'une barrière doit toujours pouvoir être atteinte par le code de n'importe quel thread indifféremment. Le "bon code" ci-dessus respecte le principe. [Fig.5] Tous les threads, y compris (0, 0, 0) atteignent la barrière. Peu importe que (0, 0, 0) soit soumis à cette contrainte. Dans l'exemple nous testons aussi si l'on n'est pas dans (0, 0, 0). Le but de ce test est simplement d'avoir des threads de durée différente pour bien mettre en évidence les résultats. Mais fondamentalement :

```

__global__ void compute(int3 *v) {
int id = threadIdx.x;
v[id].x = id+1;
v[id].y = id+1;
v[id].z = id+1;

if(id == 0) {
    v[id].x = v[1].x*2 + v[2].x*3;
    v[id].y = v[1].y*2 + v[2].y*3;
    v[id].z = v[1].z*2 + v[2].z*3;
}
}

```

est le kernel nécessaire et suffisant pour notre exemple. Ainsi s'achève ce tutoriel CUDA qui nous l'espérons vous permettra d'aborder dans de bonnes conditions les remarquables, mais complexes, exemples de la CUDA SDK.

■ Frédéric Mazué - fmazue@programmez.com

L'appel à `syncthreads` est visible par tous les threads, le code se comporte correctement.

Un appel à `syncthreads` mal placé et rien ne fonctionne comme on s'y attend.

La reconnaissance vocale sous Windows

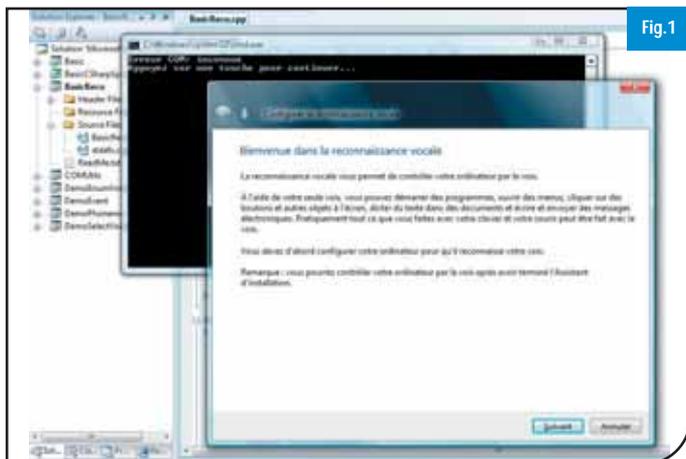
Le mois dernier nous avons découvert la synthèse vocale. A partir des bases acquises, nous nous intéressons aujourd'hui à la reconnaissance vocale, pour que votre machine obéisse à la voix de son maître.



L'API Microsoft Speech est un vaste domaine. Le mois dernier nous l'avons abordée sous l'aspect de la synthèse vocale. Maintenant que notre PC est capable de balbutier, faisons-lui tendre l'oreille en abordant l'autre aspect de Microsoft Speech: la reconnaissance vocale. Les choses sont aujourd'hui un petit peu plus difficiles. Toute l'API est en COM comme pour la reconnaissance vocale, la documentation est moins claire et le code écrit doit forcément être asynchrone. Le présent article s'appuie sur celui du mois précédent, aussi bien en ce qui concerne les rudiments de programmation COM que les managements de base de Speech. Le lecteur est très chaleureusement invité à prendre connaissance de l'article précédent avant d'aborder celui-ci. La bonne nouvelle est que si vous disposez de Windows Vista, vous pourrez lui parler en Français. Si vous avez un Windows XP, les principes de travail sont les mêmes, l'Anglais étant la langue comprise par défaut. Comme le mois précédent nous avons utilisé Visual Studio 2008, mais tout compilateur C++ conviendra pourvu qu'une Windows SDK soit installée sur votre machine. Enfin, comme d'habitude nous utilisons un pointeur intelligent maison pour automatiser le comptage de référence dans les interfaces COM.

1 PAS DE HELLO WORLD!

Cette fois nous sommes obligés de faire une entorse à la tradition :) Nous allons simplement écrire un programme rudimentaire, qui se mettra à notre écoute et rendra la main au premier mot qu'il reconnaîtra. Pour que cela fonctionne, il est impératif que la reconnaissance vocale ait été réglée une fois manuellement sur votre machine. Sans cela les APIs vous retourneront un code d'erreur et en principe vous serez convié à faire ce réglage, comme illustré ci-dessous [Fig.1]. La reconnaissance peut fonctionner selon deux



Si la reconnaissance vocale n'est pas configurée sur votre machine, les APIs COM retourneront une erreur.

modes: le mode intra-processus et le mode partagé. Nous ne travaillerons qu'avec ce dernier, le premier obligeant à manipuler les flux audio, ce qui est plus lourd. Une application qui veut utiliser la reconnaissance vocale ne peut pas savoir quand une phrase sera prononcée par l'utilisateur, puis reconnue par le moteur. Nous sommes donc contraints d'adopter une approche asynchrone. Comme le mois dernier, nous faisons le choix d'attendre le basculement d'un objet de synchronisation Windows. Comme pour la synthèse vocale, il est également possible d'être notifié au sein d'une procédure de fenêtre. Voici notre premier exemple :

```
// BasicReco.cpp
#include "COMUtils.h"
#include <sapi.h>
#include <iostream>

const ULONGLONG grammarID = 1;

void Run()
{
    HRESULT hr;
    SmartCOMPtr<ISpRecognizer> recognizer;
    SmartCOMPtr<ISpRecoContext> reco_context;
    SmartCOMPtr<ISpRecoGrammar> reco_grammar;
    HANDLE event;
    ULONGLONG interest;

    hr = ::CoCreateInstance(
        CLSID_SpSharedRecognizer,
        NULL,
        CLSCTX_ALL,
        IID_ISpRecognizer,
        reinterpret_cast<void*>(&recognizer));
    COMHelper::TestOk(hr);

    hr = recognizer->CreateRecoContext(&reco_context);
    COMHelper::TestOk(hr);

    // ou même directement
    /*
    hr = ::CoCreateInstance(
        CLSID_SpSharedRecoContext,
        NULL,
        CLSCTX_ALL,
        IID_ISpRecoContext,
        reinterpret_cast<void*>(&reco_context));
    COMHelper::TestOk(hr);
    */
}
```

```

hr = reco_context->CreateGrammar(grammarID, &reco_grammar);
COMHelper::TestOk(hr);

hr = reco_grammar->LoadDictation(NULL, SPLO_STATIC);
COMHelper::TestOk(hr);

hr = reco_context->SetNotifyWin32Event();
COMHelper::TestOk(hr);

event = reco_context->GetNotifyEventHandle();
if(event == INVALID_HANDLE_VALUE)
    COMHelper::TestOk(E_FAIL);

std::cout << "Vous devez peut etre dire:"
    " commencer l'ecoute" << std::endl;
// Si ci-dessous non invoquée, le moteur de reconnaissance
démarré
// normalement, mais nous ne sommes jamais notifiés de rien
interest = SPFEI(SPEI_RECOGNITION);
hr = reco_context->SetInterest(interest, interest);
COMHelper::TestOk(hr);
// lancer l'écoute
hr = reco_grammar->SetDictationState(SPRS_ACTIVE);
COMHelper::TestOk(hr);

// Attendre la reconnaissance d'un mot
HANDLE tab_handle[1];
tab_handle[0] = event;
::WaitForMultipleObjects(1, tab_handle, FALSE, INFINITE);
std::cout << "J'ai entendu quelque chose" << std::endl;
hr = reco_grammar->SetDictationState(SPRS_INACTIVE);
std::cout << "Maintenant je me bouche les oreilles" << std::endl;
std::cout << "Bye, bye" << std::endl;
}

int main(int argc, char** argv)
{
    try
    {
        COMInitializer ci;
        Run();
    }
    catch(COMException ce)
    {
        std::cout << ce.raison() << std::endl;
        return EXIT_FAILURE;
    }
    return EXIT_SUCCESS;
}

```

La démarche est simple. Nous commençons par instancier l'interface COM *ISpRecognizer*, ce qui revient à démarrer le moteur de reconnaissance vocale. Pour être en mesure de travailler avec lui, nous devons ensuite créer un contexte de reconnaissance et depuis celui-ci, créer une grammaire. Une grammaire est un ensemble de mots et de règles les assemblant, pour former des phrases. L'interface de gestion de grammaire, *ISpRecoGrammar*, étant instanciée,

nous y chargeons la grammaire par défaut via l'invocation de méthode *LoadDictation*(NULL, SPLO_STATIC). Ceci fait que le moteur comprend théoriquement tout: mots, phrases, etc. Nous verrons que c'est beaucoup trop, surtout si le moteur n'a pas été entraîné à travailler avec une voix particulière. Le reste du code est similaire à celui du mois dernier. Nous obtenons un objet de synchronisation et nous attendons son basculement. Comme nous avons déclaré être intéressés par un événement particulier :

```
interest = SPFEI(SPEI_RECOGNITION);
```

Le basculement se produit dès que le moteur a reconnu quelque chose, quoi que ce soit. Le programme rend alors immédiatement la main.

2 TEXTE RECONNU ET ALTERNATIVE

Ce que reconnaît le moteur, s'il n'est pas (très) entraîné à notre voix, a de quoi surprendre. Intéressons nous à l'exemple *DemoRecoGetText*. C'est tout d'abord le même code que l'exemple précédent, *basicReco.cpp*. Simplement nous avons ajouté une fonction pour afficher le texte reconnu. Cette fonction est appelée à la première reconnaissance, c'est-à-dire dès que l'objet de synchronisation bascule. Voici son code :

```

void GetText(SmartCOMPtr<ISpRecoContext> reco_context)
{
    const ULONG max_events = 10;
    ULONG nb_events;
    // voir la déclaration de SPEVENT dans sapi.h
    // TRES instructif
    SPEVENT les_events[max_events];
    HRESULT hr;
    SmartCOMPtr<ISpRecoResult> reco_result;

    hr = reco_context->GetEvents(max_events, les_events, &nb_events);
    // Attention hr vaut S_FALSE si tout va bien mais
    // que le nombre d'événements obtenus est inférieur
    // au nombre d'événements demandés
    if(!(hr == S_OK || hr == S_FALSE))
        COMHelper::TestOk(hr);
    std::cout << "Nb events: " << nb_events << std::endl;
    *(&reco_result) =
        reinterpret_cast<ISpRecoResult*>(les_events[0].lParam);

    WCHAR* le_texte;
    hr = reco_result->GetText(
        SP_GETWHOLEPHRASE,
        SP_GETWHOLEPHRASE, FALSE, &le_texte, NULL);
    COMHelper::TestOk(hr);
    std::wcout << "J'ai compris: " << le_texte << std::endl;
    ::CoTaskMemFree(le_texte);

    std::cout << "Alternatives:" << std::endl << std::endl;
    const ULONG max_alternates = 30;
    ULONG nb_alternates;
    ISpPhraseAlt* alternatives[max_alternates];
    reco_result->GetAlternates(0, SPPR_ALL_ELEMENTS,
        max_alternates, alternatives, &nb_alternates);
}

```



```

    hStateQuitter, NULL, L"Quitter", L" ", SPWT_LEXICAL, 1, NULL);
    COMHelper::TestOk(hr);

    // Très important,
    // sinon rien n'est fait
    hr = reco_grammar->Commit(0);
    COMHelper::TestOk(hr);
    hr = reco_grammar->SetRuleState(ruleExitName, 0, SPRS_ACTIVE),
    COMHelper::TestOk(hr);

    hr = reco_context->Resume(0);
    COMHelper::TestOk(hr);

    return reco_grammar;
}

```

Ce code ne pose pas de difficultés en soi. On remarquera simplement que la définition de la grammaire est faite entre les appels des méthodes *Pause* et *Resume* du contexte. La MSDN précise qu'à ce moment le code ne doit pas lambiner, car le moteur travaille toujours et le flux audio est alors placé dans un tampon susceptible d'arriver à saturation. On remarque encore l'appel à *Commit* qui valide la règle dans le contexte et surtout l'appel à *SetRuleState* qui place la règle dans un état actif. Dès lors, l'exemple saura entendre quand vous lui direz 'Quittez' et ne se trompera pas.

4 TRAVAIL AVEC PLUSIEURS CONTEXTES

Il est possible de travailler avec des règles plus complexes, et plusieurs contextes. Notre dernier exemple illustre cela. Nous travaillons avec deux contextes. Nous définissons une règle de reconnaissance dans le premier contexte telle que le moteur générera un événement lorsque les mots "un deux trois" seront reconnus en cascade. Le second contexte est identique à celui de l'exemple précédent et reconnaît le mot 'Quitter' seulement. Nous avons fait le choix d'activer un contexte puis l'autre pour obéir à la logique supposée d'une application. Toutefois il est parfaitement possible d'activer plusieurs contextes simultanément. Dans ce cas, l'API Win32 Wait-ForMultipleObject recevrait un tableau d'objets de synchronisation, chacun d'eux étant issu d'un contexte de reconnaissance. Voici un extrait de notre dernier exemple, *DemoRecoCount*.

```

const ULONGLONG grammarExitID = 1;
const WCHAR* ruleExitName = L"RegleQuitter";

const ULONGLONG grammarCountID = 2;
const WCHAR* ruleCountName = L"RegleCompter";

SmartCOMPtr<ISpRecoGrammar>
InstallGrammarCount(SmartCOMPtr<ISpRecoContext> reco_context)
{
    HRESULT hr;
    SPSTATEHANDLE hStateCountOne, hStateCountTwo, hStateCountThree;
    SmartCOMPtr<ISpRecoGrammar> reco_grammar;

```

```

    hr = reco_context->CreateGrammar(grammarCountID, &reco_grammar);
    COMHelper::TestOk(hr);

    hr = reco_grammar->ResetGrammar(GetUserDefaultUILanguage());
    COMHelper::TestOk(hr);

    // Créer une règle
    // On commence par créer les trois états successifs
    hr = reco_grammar->GetRule(ruleCountName,
        0, SPRAF_TopLevel | SPRAF_Active, TRUE, &hStateCountOne);
    COMHelper::TestOk(hr);
    hr = reco_grammar->CreateNewState(hStateCountOne, &hStateCountTwo);
    COMHelper::TestOk(hr);
    hr = reco_grammar->CreateNewState(hStateCountTwo, &hStateCountThree);
    COMHelper::TestOk(hr);

    // Ajouter un
    hr = reco_grammar->AddWordTransition(
        hStateCountOne, hStateCountTwo,
        L"Un", L" ", SPWT_LEXICAL, 1, NULL);
    COMHelper::TestOk(hr);
    // Ajouter deux
    hr = reco_grammar->AddWordTransition(
        hStateCountTwo, hStateCountThree, L"Deux", L" ",
        SPWT_LEXICAL, 1, NULL);
    COMHelper::TestOk(hr);
    // Ajouter deux
    hr = reco_grammar->AddWordTransition(
        hStateCountThree, NULL, L"Trois", L" ", SPWT_LEXICAL, 1, NULL);
    COMHelper::TestOk(hr);

    // Très important,
    // sinon rien n'est fait
    hr = reco_grammar->Commit(0);
    COMHelper::TestOk(hr);

    return reco_grammar;
}

SmartCOMPtr<ISpRecoGrammar>
InstallGrammarExit(SmartCOMPtr<ISpRecoContext> reco_context)
{
    // Identique à l'exemple précédent
}

```

La fonction *InstallGrammarCount* illustre comment définir un enchaînement de mots, qui revient en fait à un enchaînement d'états. Enfin, si ici nous avons tout défini par programmation, les règles complexes et surtout volumineuses peuvent être définies dans des fichiers XML et chargées depuis ces fichiers. Ce point ne pose aucune difficulté et nous renvoyons le lecteur à la documentation MSDN.

■ Frédéric Mazué - fmazue@programmez.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

Développez en Ruby on Rails avec 3rdRail

3rdRail est un environnement de développement édité par CodeGear, qui fournit un certain nombre de fonctionnalités afin de programmer des applications Ruby ou –et c'est son but avoué– des applications web avec Ruby on Rails.



Gestion des serveurs et Project Commander

Nous allons maintenant voir ce que nous avons "créé" jus-

qu'ici. Supprimez le fichier `index.html` situé dans le répertoire "public" du projet. Vous constaterez au passage que certains des répertoires comme celui-ci ou "vendor", "tmp" (...) sont laissés à l'identique de leur organisation physique. Afin de pouvoir utiliser notre scaffold plus facilement, redirigeons la racine du site vers l'index du contrôleur `billets`. Ouvrez pour cela le fichier "routes.rb", qui se trouve sous le noeud "Configuration". A la fin de la classe, en-dessous des trois routes définies par défaut, ajoutez la ligne :

```
map.root :controller => "billets"
```

Gestion des serveurs

Nous allons maintenant pouvoir découvrir une nouvelle vue de l'IDE, qui est l'onglet de gestion des serveurs. Vous l'aurez sans doute déjà repéré, il est situé dans la partie inférieure de la fenêtre, au même niveau que la console.

L'IDE crée un "objet" serveur pour chaque projet Rails, et chacun de ces serveurs est accessible à tout moment dans cet onglet. On voit ainsi que j'ai actuellement trois projets Rails importés dans 3rdRail, ayant chacun un serveur dans cette vue. Ces serveurs sont votre interface avec le script Mongrel ou WEBrick qui vous permettra de tester votre application en période de développement. Ils ont trois états : à l'arrêt, démarré ou démarré en mode de débogage. Nous ne verrons pas le débogage dans cet article, je vous propose donc de sélectionner votre serveur et de le démarrer de manière standard. Accédez maintenant à votre site avec votre navigateur préféré à l'adresse <http://localhost:3000/>. L'adresse du port pouvant changer, vérifiez au début de la console qui s'ouvre en même temps que le lancement du serveur celui utilisé sur votre machine. Pas de surprise : vous arrivez sur le listing des billets disponibles (aucun pour le moment), avec un lien pour créer un nouveau billet.

Maintenant que nous avons vérifié que tout fonctionnait bien du côté du scaffolding, avançons dans notre application : un blog où tout visiteur peut poster, ce n'est pas tout à fait ce que l'on souhaite !

Nous allons donc ajouter les fonctionnalités d'identification de base (inscription, connexion et déconnexion) et utiliser pour cela le plugin `restful_authentication`.

Project Commander

Avec 3rdRail, nous disposons du **Project Commander**, une console un peu particulière, exécutée dans le contexte du projet et qui propose l'autocomplétion sur les commandes comme `rake`, `gem`, `rdoc`, `script/generate`, `script/plugin` etc. Si vous ne voyez pas de Project Commander dans la partie inférieure de l'IDE, vous pouvez l'afficher

grâce au menu "Window > Project Commander > Show Last or New". Installons donc le générateur `restful_authentication` à partir du Project Commander, grâce à la commande `script/plugin install http://svn.techno-weenie.net/projects/plugins/restful_authentication/`. Vous noterez qu'il n'y a pas besoin d'inclure l'appel à l'interpréteur ruby : il est implicite. Attendez la fin du téléchargement, puis à l'identique lancez la commande de génération du plugin : `script/generate authenticated User sessions`

En premier paramètre, `User` désigne le modèle qui sera généré pour gérer les comptes utilisateurs, et `sessions` correspond au contrôleur utilisé pour gérer les connexions et déconnexions des utilisateurs. Avant d'aller plus loin, effectuons tout d'abord une petite retouche nécessaire au bon fonctionnement du plugin : ouvrez le fichier "sessions_controller.rb" en double-cliquant sur "Controllers > Sessions". Vous trouverez cette ligne au début du fichier :

```
include AuthenticatedSystem
```

Un commentaire au-dessus vous indique de déplacer cette ligne dans le contrôleur de l'application. Coupez et collez donc cette ligne dans `application.rb`, et supprimez son homologue dans le fichier `users_controller.rb`. Ainsi, tous les controllers auront accès aux fonctionnalités du plugin, ce qui est bien sûr nécessaire pour la suite.

Navigation grâce aux dépendances

Ouvrez le fichier "billet.rb", dans "Models > Billet", et ajoutez l'association qui liera un billet à son auteur : `belongs_to :auteur, :class_name => "User"`.

Vous pouvez bien sûr utiliser simplement `belongs_to :user`, attention alors à faire les corrections nécessaires dans la suite de l'article. Il faut maintenant spécifier, à l'inverse, qu'un utilisateur peut posséder plusieurs billets. Cela se fait dans le fichier "user.rb" dans "Models > User" mais pour y accéder, nous allons utiliser un autre outil de 3rdRail : la vue des **dépendances**, dans la droite de l'IDE.

Cette vue des dépendances montre d'une part, dans la liste "outbound", **vers quoi** l'élément actif a des liens et d'autre part, dans la liste "inbound", **qui** a des liens vers l'élément actif. Ledit élément étant mis à jour en fonction de votre position dans l'éditeur.

Nous pouvons donc voir, maintenant que l'association `belongs_to` est ajoutée, qu'elle apparaît dans la liste "outbound", et que par conséquent le modèle "User" apparaît sous "Rails Model" dans cette même liste. Il suffit de double-cliquer sur cet item pour se rendre dans le fichier "user.rb".

Vous pouvez donc y ajouter l'association `has_many :billets`, et vous verrez cette association apparaître dans les dépendances.

Cette vue est très utile pour se rendre compte d'un coup d'oeil de la portée des modifications que l'on peut apporter à un fichier. Quels seront les autres modèles ou contrôleurs qui seront impactés par tel ou tel changement dans le code ? Qui référence ce fichier, cette classe ? Et bien d'autres questions...

Création d'une migration

Nos associations étant créées, il nous faut ajouter un champ `auteur_id` (ou `user_id` suivant le choix fait plus haut) à notre table `billets`. Nous allons donc créer une nouvelle migration : ouvrez le noeud "Database" et effectuez un clic droit sur l'item "Migrations" pour choisir "New Migration...". Cela vous ouvre un assistant demandant le nom de la migration.

Entrez par exemple "update billets", et validez. Votre fichier de migration et son squelette sont alors générés, et nous pouvons renseigner les fonctions `self.up` et `self.down` :

```
def self.up
  add_column :billets, :auteur_id, :integer
end

def self.down
  remove_column :billets, :auteur_id
end
```

Sauvegardez et lancez la migration par un clic droit sur la migration dans le Rails Explorer, et en choisissant l'item "Rails > Run Database Migration (...)". Pour terminer cet article, nous allons donc modifier quelques petites choses pour utiliser au mieux le plugin `restful_authentication` et finaliser notre première version de ce mini-blog.

Ajout de routes

En premier lieu, nous allons ajouter quelques routes pour faciliter l'accès au formulaire d'inscription, de connexion, et à la déconnexion. Ouvrez le fichier "routes.rb" sous le noeud "Configuration" et ajoutez les lignes suivantes :

```
map.inscription "/inscription", :controller => "users", :action => "new"
map.connexion "/connexion", :controller => "sessions", :action => "new"
map.deconnexion "/deconnexion", :controller => "sessions", :action => "destroy"
```

Vous pouvez d'ores et déjà accéder à <http://localhost:3000/inscription> et créer un nouveau compte utilisateur, c'est-à-dire un nouvel objet User dans votre base de données.

L'accès à <http://localhost:3000/connexion> et <http://localhost:3000/deconnexion> vous permettra respectivement de créer et détruire une session, et donc de vous connecter et déconnecter.

Côté contrôleur...

Côté contrôleur, il nous reste à associer correctement un billet à l'utilisateur qui le crée, et à protéger par mot de passe les accès aux pages de création, d'édition et de suppression de messages.

La fonction `current_user` du plugin renvoie l'instance de l'utilisateur courant, et il nous suffit donc de modifier l'action `create` du contrôleur `billets` pour lier un billet à son auteur (ajouts en gras) :

```
def create
  @billet = Billet.new(params[:billet])

  @billet.auteur = current_user
  ...
```

Dorénavant, tout billet créé sera associé à l'utilisateur actuellement connecté, mais encore faut-il forcer la connexion. Nous allons pour cela utiliser un filtre avant l'exécution des actions de création, modification et suppression de billet afin de s'assurer qu'une session est ouverte. Toujours dans le fichier "billets_controller.rb", ajoutez au début de la classe :

```
before_filter :login_required, :only => [:new, :update, :destroy, :edit]
```

La fonction `login_required` fournie par le plugin `restful_authentication` permet de s'assurer qu'une session est ouverte et, le cas échéant, de forcer le passage par la page de connexion.

Ajout d'une méthode au modèle User

Enfin, avant de passer aux dernières mises au point dans les vues, nous allons ajouter une méthode toute simple au modèle User, qui nous permettra de savoir si un utilisateur possède ou non un billet donné. Nous allons appeler cette méthode `owns?`, et l'ajouter au fichier "user.rb" dans les modèles :

```
def owns?(billet)
  billet.auteur_id == self.id
end
```

Modification du layout

Sous le noeud "Layouts", vous verrez que la génération de notre ressource Billet a créé un layout pour le contrôleur billets.

Changez son nom de "billet.html.erb" vers "application.html.erb", il sera ainsi appliqué à la totalité de l'application, y compris aux fenêtres d'inscription et de connexion. Modifions maintenant ce layout pour ajouter les informations et les liens de connexion, déconnexion et inscription. Le principe est simple : si l'utilisateur est connecté, nous afficherons "Connecté en tant que login" suivi d'un lien de déconnexion. Dans le cas contraire, nous afficherons "Déconnecté" suivi d'un lien de connexion et d'inscription. La fonction `logged_in?` du plugin `restful_authentication` nous permettra de savoir si une session est ouverte. Utilisons-la donc à bon escient :

```
<% if logged_in? %>
  Connecté en tant que <%= current_user.login %> | <%= link_to
  "Déconnexion", deconnexion_path %>
<%else%>
  Déconnecté | <%= link_to "Connexion", connexion_path %> - <
  = link_to
  "Inscription", inscription_path %>
<%end%>
```

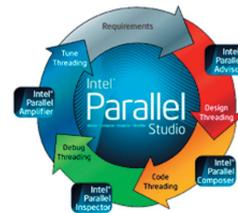
Ajoutez ce code juste à l'ouverture du tag `<body>` du layout, et vous aurez maintenant une indication de l'état de votre connexion au site. Nous avons donc maintenant un système minimaliste de blog gérant l'inscription et la connexion d'utilisateur afin de protéger les accès qui nécessitent de l'être. A vous de coder maintenant !

Pour en savoir plus : <http://www.codegearfrance.com>

■ Olivier Lance

Intel Parallel Studio

Plus de cœurs, plus rapide, plus scalable : du multi-core au "many-core"



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

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

QUATRE DÉFIS

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

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

Le troisième problème vient de l'équilibre de charge entre threads. Pour que votre tâche s'exécute le plus vite possible vos threads doivent avoir la même charge de travail. Si vous en donnez plus à l'un qu'aux autres il finira plus tard et les autres attendront. Si votre charge est facile à prédire vous pou-

vez la scinder en parties égales dès le début et la répartir entre vos threads, si elle ne l'est pas, scidez la en petites parties et faites une queue où les threads iront piocher au cours du travail.

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

INTEL PARALLEL AMPLIFIER

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

patibilité que vous travailliez au niveau du thread, avec OpenMP ou Intel Threading Building Blocks, par exemple.

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

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

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

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

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

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

EXEMPLE

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

POINTS CHAUDS

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

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

CONCURRENCE

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

Je regarde le détail par fonction et vois réapparaître la fonction d'avancement. Elle n'acapare plus le processeur mais empêche les threads de s'exécuter en parallèle. Le détail du source est sans appel : la section critique

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

ATTENTES ET BLOCAGES

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

L'instruction *parallel for* accepte des arguments. Pour commencer testons le cas simple : découpe de la charge en deux parties identiques avec " parallel for schedule (static,2) ". Résultat : 4.41s, bien, mais si je tente de perturber un peu la machine pendant l'exécution le temps augmente rapidement (un des threads travaille alors plus vite que l'autre). Avec (static,8) je suis plus robuste face aux perturbations et atteint un respectable 4.12s. Allons jusqu'à (static,200000) : 4.42s, les deux threads finissent, quoiqu'il arrive, en même temps

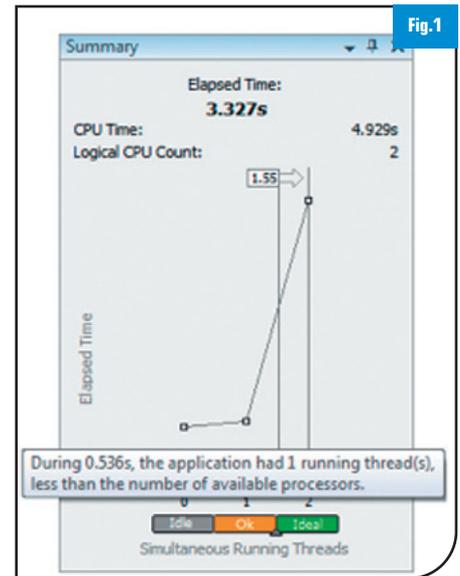


Fig.1

mais nous voyons le coût de la gestion de queue qui augmente. L'idéal se trouve probablement avec (guided) où la queue donne de gros morceaux au début et de petits à la fin, pour optimiser l'équilibre balance/communication, résultat : 4.09s. Le graphique nous indique que le logiciel s'exécute à 99% avec deux threads, le rêve ! Vous pouvez maintenant escompter une excellente utilisation d'un grand nombre de cœurs.

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

CONCLUSION

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

■ Paul Guernonprez - Ingénieur logiciel Intel

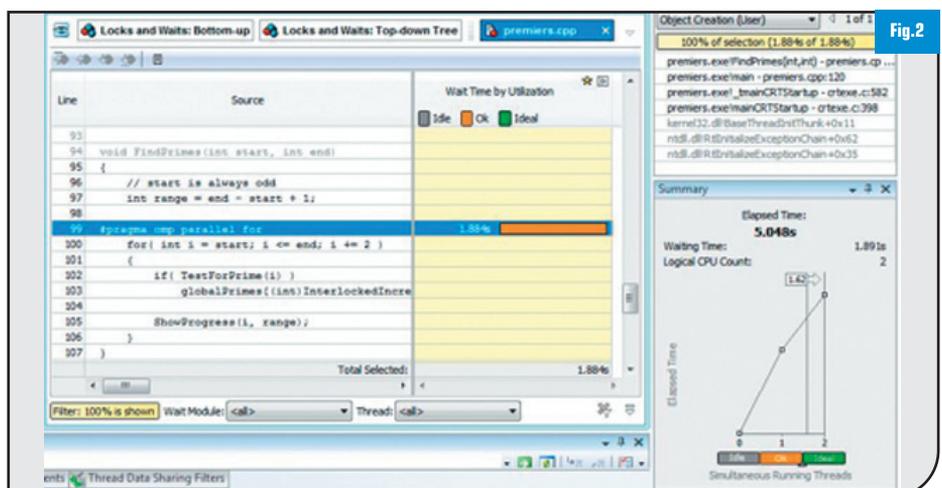
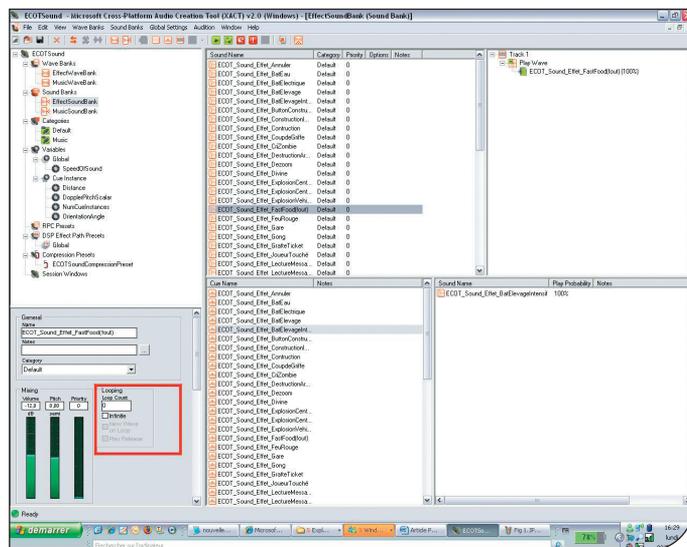


Fig.2

Créer un jeu vidéo de A à Z avec XNA

3^e partie

La création d'un jeu vidéo est une savante alchimie. Il faut composer donc avec les graphismes, le gameplay, l'intelligence artificielle ou encore l'ergonomie avec une vision d'ensemble. Si un point est de faible qualité, il va niveler vers le bas la qualité de tout le travail lui-même. La gestion sonore est d'ailleurs un élément encore un peu plus particulier...



En effet, il contribue à rendre " vivante " l'application (si je construis un bâtiment, cela déclenche tel son, etc.) mais surtout à " immerger " le joueur et enrichir son expérience quel que soit le type de jeu. Par exemple, dans un jeu d'infiltration ou survival horror, la pression va nettement augmenter si on entend le souffle de notre avatar ou chaque bruit de pas. On a encore tous à l'esprit les musiques de la génération GameBoy, notamment Tetris. D'après les dernières études marketing, le son joue pour plus de 50% dans le succès d'un jeu !

Le son avec XNA

Avec la sortie de XNA Framework 3.0, il est possible désormais d'implémenter la gestion sonore de deux manières différentes :

- Microsoft Cross-Platform Audio Creation Tool (XACT) : il s'agit d'un outil tiers qui permet de compiler les sons et d'y appliquer quelques réglages très simplement (réglage du volume, boucle d'un son, sa fréquence, ajout d'effets. C'est également un outil apprécié par les *sound designer* car il leur offre plusieurs fonctionnalités avancées permettant d'obtenir ces résultats avec un minimum de code. L'inconvénient est par contre que XACT ne supporte que le Wav.
- API sonore et intégrée à XNA. Le Zune ne supportant pas XACT, il est désormais possible de ne plus passer par cet outil également sur PC et Xbox 360. On peut alors rajouter les sons en les glissant-déposant simplement dans le Content Manager. Les formats sont plus larges (support du MP3, Wav, Wma, etc.) mais il n'est pas possible de reproduire toutes les possibilités qu'offre XACT.

La magie de XACT

Ouvrons tout d'abord XACT (Menu Démarrer, même répertoire que XNA puis Tools). Création d'un nouveau

projet. Le format est .xap. Chaque projet contient au moins une Wave Bank et une Sound Bank. Grosso modo, une banque de fichiers sonores et une banque de son de votre projet. A vous d'en créer une en cliquant droit sur l'espace correspondant et éponyme sur le bord gauche de l'écran.

Vous allez maintenant ajouter plusieurs sons à votre Wave Bank. Allez sur la fenêtre et wave bank et clic droit -> Insertion Nouveau Fichier. Ils sont normalement apparus dans la fenêtre, rien de compliqué pour l'instant. Sélectionnez les sons de la Wave Bank et glissez-déposez-les dans les deux sections de fenêtre de votre Sound Bank : Sound & Cue. On interagit en XNA pour créer des effets sonores avec les Cues et lorsque l'on les définit dans XACT, on peut leur attacher un ou plusieurs sons, provenant d'une seule ou plusieurs **wave bank** différentes. Le panel gauche permet de définir quelques options sonores : volume, fréquence ou encore le nombre de fois que le son va boucler. Après avoir terminé ces configurations, il ne reste plus qu'à compiler et quitter.

Musique Maestro !

Chaque soundbank génère un .xsb et chaque wavebank un .xwb Glissez-déposez le fichier .xap dans le content Manager, en créant par exemple un répertoire Audio. Il n'est pas nécessaire d'y rajouter les autres fichiers, ils seront ainsi mis à jour automatiquement. Contrairement à la plupart d'autres ressources, les sons ne sont pas chargés en mémoire par la méthode Content.Load. Le Content Pipeline est toutefois appelé à la compilation pour parser les fichiers créés par XACT et instancier des classes plus " traditionnelles ".

Il existe deux modes de "lecture" d'une wave bank : le *in-memory* et le *streaming*. Dans le premier cas, tous les *wav* sont entièrement mis en mémoire depuis le support



de stockage. La lecture est donc très rapide et n'accuse aucun temps de latence. C'est évidemment le mode de lecture le plus conseillé. Par contre, il va sans dire que la mémoire de nos PC et surtout de la XBOX 360 (512 Mo de Ram) peut très vite être atteinte de cette manière lorsque le nombre / poids des sons est très important. On recommande principalement d'utiliser le streaming pour les musiques, surtout de fond, les dialogues ou encore le son de cinématiques. Le streaming présente toutefois des possibilités de latence. Pour l'éviter au maximum, le sound designer peut déjà réduire le poids des fichiers et/ou les compresser (XACT permet de configurer un " Compression Pisset " avec un % de compression). Enfin, XACT alloue un buffer pour chaque son en streaming (beaucoup plus petit que la taille réelle du fichier) afin de le " pré-charger " et supprimer ce problème.

Prenons un cas très simple : l'utilisation d'une seule wavebank et une seule soundbank. On va d'abord instancier un " moteur " de son qui permettra ensuite d'instancier notre couple wavebank / soundbank.

```
AudioEngine audioEngine;
WaveBank waveBank;
SoundBank soundBank;

public void InitSound()
{
    audioEngine = new AudioEngine("FirstXactProject.
xgs");
    waveBank = new WaveBank(audioEngine, "Wave Bank.
xwb");
    soundBank = new SoundBank(audioEngine, "Sound Bank.
xsb");
}
```

Un son peut-être joué sans passer par la création d'un Cue mais cette dernière méthode est plus flexible car elle offre la possibilité d'accéder à plusieurs méthodes, comme vérifier si le son est en train de se jouer, le mettre en pause, l'arrêter, etc. Voici quelques méthodes utilitaires :

```
public static void PlayMusic(string name)
{
    if (this.cue != null)
        this.cue.Dispose();
    this.cue = waveBank.GetCue(name);
    this.cue.Play();
}

public void StopMusic()
{
    if (IsPlayingMusic())
        this.cue.Stop(AudioStopOptions.AsAuthored);
}
```

```
}

public bool IsPlayingMusic()
{
    if (this.cue == null)
        return false;
    return this.cue.IsPlaying;
}

public static float GetVolume()
{
    return audioEngine.GetGlobalVariable("Engine
Thrust");
}
```

Toutes les routes mènent à Rome

Comme nous vous l'avons présenté au début de cette chronique, il existe une autre manière de gérer les sons sous XNA. On doit d'abord glisser-déposer les fichiers dans le ContentManager. Il existe deux classes pour cela : *SoundEffect* pour justement des effets sonores, plutôt court, et *Song*, davantage pour les musiques.

```
SoundEffect mySound = Content.Load<SoundEffect>(@"Au
dio\MySoundEffect");
SoundEffectInstance sef = mySound.Play(0.1f, 0.0f,
0.0f, true);
```

La méthode Play peut être également appelée sans paramètre. Cet appel permet respectivement de définir le volume sonore (compris entre 0 et 1), la fréquence (entre -1 et 1), de définir la direction (-1 pour le côté gauche, 0 pour la valeur par défaut et 1 pour la droite) et le booléen permet de renseigner si le son doit boucler ou non.

```
Song mySong = Content.Load("Audio\MyMusic");
MediaPlayer.Play(mySong);
```

Le process est un peu différent pour les musiques. La classe Song permet de récupérer les informations sur une musique (Album, Genre, etc.) et la classe MediaPlayer remplit son rôle de " lecteur multimédia " via ses méthodes statistiques, on peut ainsi mettre à la suite une autre musique, changer le volume, etc.

Les possibilités sont donc illimitées et il ne tient qu'à vous maintenant de les exploiter : son 3D, localisation d'un effet sonore, gestion des volumes, etc. Le mois prochain, nous concluons cette chronique sur XNA en nous concentrant sur le portage d'un jeu vers la XBOX 360.

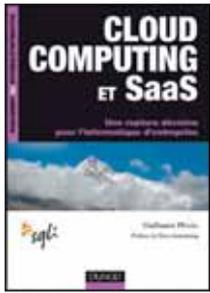
Suite et fin le mois prochain.

■ Frédéric PEDRO & Anthony CHEN KUANG PIAO

Lauréat de l'Imagine CUP dans la catégorie jeu vidéo et soutenue par Microsoft, ECOThink développe des jeux vidéo innovants grâce à la technologie XNA. contact@ecological-tycoon.com

ARCHITECTURE

Cloud Computing et SaaS



Difficulté : ** / ***

Editeur : Dunod

Auteur : Guillaume Plouin

Prix : 27 €

Voilà le premier véritable livre " tout public IT et technique " en français sur le nuage informatique et le monde du SaaS

en général. L'auteur rappelle rapidement les origines du cloud et du SaaS, notamment via les fameux services ASP, puis les RIA, les applications hébergées. On entame ensuite les fondements du Cloud Computing et du SaaS avec les aspects services, plateformes et infrastructures. L'auteur rappelle avec justesse que le SaaS est un nouveau modèle logiciel (de distribution et économique). La 2e partie se veut plus orientée Cloud et examine comment s'y mettre selon le profil utilisateur (utilisateur, décideur et technicien). Le livre fait une large place aux offres des éditeurs. Sur le cloud, l'ouvrage parle des architectures et de plate forme via le PaaS. Ce livre est une bonne introduction à ces évolutions.

WEB

Javascript

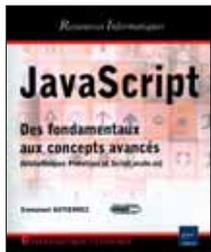
Difficulté : ***

Editeur : Eni

Auteur :

Emmanuel Gutierrez

Prix : 27,14 €



Javascript continue à être mal vu des développeurs et à juste titre. Malgré les améliorations dans l'outillage et les frameworks, Javascript demeure complexe et difficile à maîtriser. Nous allons donc revenir aux fondamentaux du langage et aborder quelques notions avancées, notamment via les frameworks comme Prototype. Un bon ouvrage pour débiter en douceur.

BASE DE DONNÉES

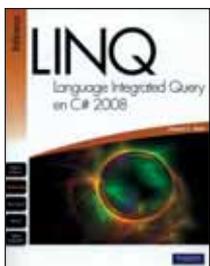
LINQ

Difficulté : ***

Editeur : Pearson

Auteur : Joseph C. Ratz

Prix : 55 €



Linq est une des technologies les plus prometteuses de Microsoft pour communiquer, échanger avec des données, une base de

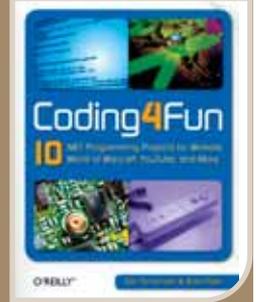
livre du mois

Coding4Fun

Difficulté : *** / **** - Editeur : O'Reilly

Auteur : collectif - Prix : 39,99 \$ US

Qui a dit que l'on ne pouvait pas coder fun ? Certainement pas Programmez ! Et cet ouvrage est là pour rappeler à tous les codeurs, à la maison, au bureau, en vacances, que la programmation peut être drôle et pas " prise de tête " ! Ce livre le prouve avec maestria. Bien que 100 % .Net, en 10 exemples, les auteurs nous montrent des projets, des technologies loin du site web. Notre coup de cœur concerne la programmation d'une wii remote (console wii). Mais on y parle aussi jeux avec la Xbox et le Zune, de vidéo avec YouTube, de Twitter. C'est certes ludique mais finalement, on apprend des paradigmes de développement que l'on ne rencontre pas forcément tous les jours... En anglais.



données, sans passer par des requêtes SQL. Mais maîtriser une telle technologie nécessite une bonne pratique et une connaissance approfondie de ce langage. L'ouvrage se destine aux développeurs C# voulant manipuler Linq. Et l'ensemble des déclinaisons Linq est abordée : Objects, XML, Dataset, SQL. A chaque fois, l'auteur introduit le type de Linq, sa structure, son fonctionnement, les classes, les opérateurs... Précis et complet, ce livre séduira le développeur et même le DBA qui voudrait en savoir plus. Seul petit manque : l'absence d'un chapitre sur pLinq.

MATÉRIEL

Hacez votre Eee PC

Difficulté : **

Editeur : Eyrolles

Auteur : Christophe Guelff

Prix : 19,90 €



Les netbooks sont un paradis pour le bidouilleur et le développeur en tout genre. Ce livre nous montre les entrailles du Eee PC dans toutes leurs beautés ! Que ce soit en distribution Linux ou en Windows, le netbook peut être boosté à l'extrême ou encore accéder à des fonctions plus ou moins cachées. On découvrira tout d'abord l'univers du netbook, puis le mode console, toujours utile à connaître. L'auteur aborde ensuite l'optimisation du poste, les logiciels avec Xandros ou Windows et toutes les astuces pour booster le démarrage et le matériel. Le titre est sans doute un peu trompeur quand on referme le livre mais finalement, il peut vous apprendre quelques astuces bienvenues.

WEB

Expression Web

Difficulté : ***

Editeur :

Microsoft Press

Auteur : Chris Leeds

Prix : 29,90 €



Qu'est-ce qu'Expression Web ? Comment le maîtriser ? Voici le guide officiel Microsoft sur son outil phare de développement web. On débute en douceur par le rappel de ce qu'est une page web moderne, les standards. Puis on entame les choses sérieuses avec la création, le déploiement, la gestion du site. Enfin, on termine par les aspects propres à ASP.Net.

CYCLE DE VIE

Team Foundation Server

Difficulté : ***

Editeur : Eni

Auteur : Florent Santin

Prix : 39 €



Perçu comme complexe à maîtriser et à mettre en œuvre, Team Foundation Server (TFS) n'en est pas moins un environnement stratégique pour les équipes de développement. On revient tout d'abord sur l'approche cycle de Microsoft et les différentes éditions de Visual Studio. Le but est de nous prendre la main, de l'installation à la mise en œuvre concrète. Et là toutes les notions avancées de TFS sont abordées : le contrôle de source, la compilation automatique (avec Team Build), la documentation, l'environnement de travail, etc. Un must !

Retrouvez
dans ce magazine
une version d'évaluation

“ Avec l'atelier BLU AGE de transformation de modèles UML2, produisez instantanément des applications métiers 100% conformes aux besoins de vos utilisateurs ! ”

Modèle = Application

Application = Modèle

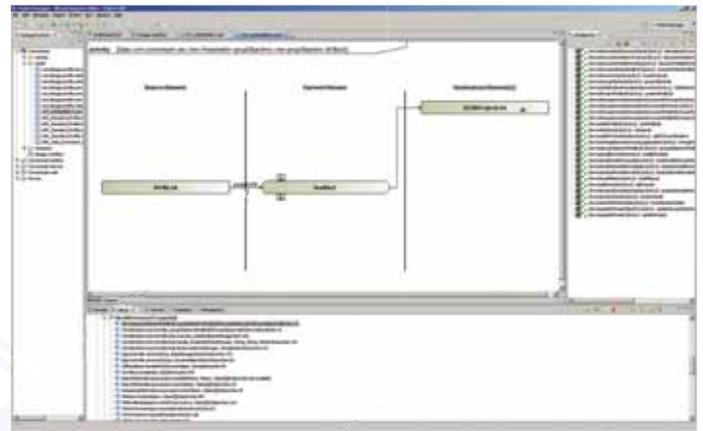
Avec BLU AGE®, valorisez enfin vos connaissances en UML !
Rejoignez la révolution des développements agiles !



Informations et évaluation gratuite :
01 56 05 60 91
www.bluage.com

Eclipsez vos problèmes, Rationalisez vos développements !

- Génération de 100% du code source de vos applications,
- Génération vers la majorité des Framework JAVA et .Net existants,
- Transformation et débogage de vos modèles UML2 en temps réel,
- Intégration native avec les IDE Eclipse et IBM Rational,
- Extraction du modèle PIM des applications existantes pour modernisation,
- Intégration avec les modeleurs UML 2 du marché,
- Génération de BPEL,
- Nouveau ! Génère vos applications riches Microsoft .NET (RIA/XAML).



Construit sur Eclipse, BLU AGE® Edition 2009 transforme instantanément vos modèles UML en applications JAVA EE et .Net.
Conforme au MDA, BLU AGE® outille vos développements agiles et vos projets de modernisation d'applications.



 BLU AGE est cofinancé par l'Union Européenne. L'Europe s'engage en Aquitaine avec le Fonds européen de développement régional.

BLU AGE® is a registered trademark of NETFFECTIVE TECHNOLOGY - Trademarks are property of their respective owners
MDA, UML and MDD are either registered trademarks or trademarks of Object Management Group, Inc. in the United States and/or other countries
Eclipse, Built on Eclipse and Eclipse Ready, BIRT, Higgins are trademarks of Eclipse Foundation, Inc.
The Ready For IBM Rational software mark and the trademarks contained therein are trademarks of IBM Corp. IBM is not the licensor of this Business Partner's product and does not make any warranties regarding this Business Partner's product.
Microsoft is a registered trademark of Microsoft Corp. in the U.S. and other countries.

Voyez toute la forêt.



Présentation de la gamme d'outils de visualisation la plus puissante du marché :

ILOG, une société du groupe IBM, fournit la gamme d'outils la plus complète du marché, pour la création d'interfaces utilisateurs graphiques et interactives avancées. Les outils de visualisation d'ILOG facilitent la prise de décision en rendant de gros volumes de données complexes plus facilement compréhensibles et manipulables par les utilisateurs.

Diagrammes · Tableaux de bord · Cartes · Graphiques · Interface personnalisées · Java · .NET · C++ · Eclipse · Flex/AIR · Ajax

La suite d'outils ILOG visualisation – la plus fiable et la plus performante du marché – est conçue pour résoudre les applications les plus exigeantes et offre une large gamme de fonctionnalités pour un large éventail d'industries :

Aérospatial et défense
Banques/Valeurs/Assurance
Santé/Drogues/Biotechnologie

Fabrication et matériaux
Logiciel et services
Matériel et équipement de technologie

Services de télécommunications
Transport et logistique
Ressources Naturelles/Energie

ILOG VISUALISATION



Découvrez de nouvelles possibilités.

